

CipherTrust Cloud Key Manager for SAP Applications in Google Cloud Platform Providing Bring Your Own Key 'BYOK' and Cloud-native Key Management



The Business Challenge

As enterprise data and workloads continue to migrate to the Google Cloud Platform (GCP), the need to keep sensitive data secure continues to grow. In a recent IDC Data Threat Report study, 50% of all corporate data is stored in the cloud of which 48% of that data is considered sensitive. With a significant footprint across the large enterprise community, SAP figures meaningfully in GCP migration discussions – both for the volume of customers that trust SAP applications and for the type of sensitive data they use.

While GCP offers encryption and other data protection schemes to support migration efforts, best practices, regulations, and government policies are increasingly requiring a level of sovereignty in data protection. Ultimately, organizations must keep control of their keys separate from the data they secure. Google and SAP address these requirements through various Bring Your Own Keys (BYOK) design architectures that enable the companies owning the workloads to control their encryption keys and their workloads. The separation of data from encryption keys provides the sovereignty they need to ensure that their data can remain securely within the confines of a specific geography and that their sensitive data in GCP is useless even if it is accessed by unauthorized users.

In addition to this, the flexibility and benefits of a multi-cloud deployment also increase the complexity of managing encryption keys across the enterprise's workloads. Managing the lifecycle of both BYOK keys kept within the enterprise, and cloud-native keys kept in GCP is complicated and time-consuming. Silos of cloud encryption keys makes IT administration more difficult and complicates the reporting needed to establish regulatory compliance.

When taken all together, these challenges can present serious obstacles to customers moving their SAP investments into GCP. Fortunately, SAP and Thales are working together to address these challenges for our customers.

The Solution

Thales and SAP now offer external, multi-cloud key lifecycle management for SAP applications. With the integration between SAP's Data Custodian and Thales' CipherTrust Cloud Key Manager, organizations can seamlessly manage the encryption key life cycle of SAP applications on GCP from the same pane of glass as their other cloud encryption implementations. The SAP Data Custodian offers integrated visibility and security functionality in addition to key management to secure GCP-based SAP data. By adding CipherTrust Cloud Key Manager, highly-regulated customers can externally root their encryption keys in a purpose-built hardware appliance, or generate their own encryption keys to be used by their SAP applications.

Features and Benefits

- Generate encryption keys with HSM based entropy and bring them to your SAP applications on GCP
- Make encryption administration easier with centralized key lifecycle management across multiple cloud environments
- Comply with the most stringent data protection and sovereignty mandates with encryption and key management
- Simplified compliance reporting with detailed audit logs and prepackaged reports
- Root keys in up to FIPS 140-2 Level 3 security by leveraging CipherTrust Manager or Thales Luna HSMs

In Summary

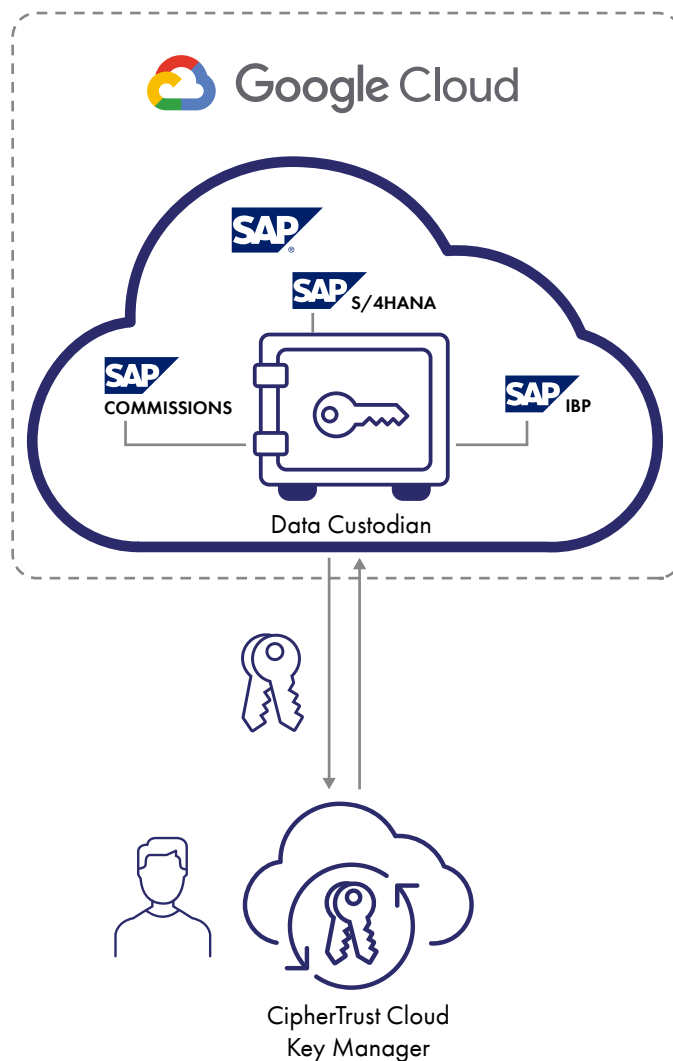
Whether they choose to take advantage of SAP's native security functionality or bring their own keys to the encryption conversation, CipherTrust Cloud Key Manager simplifies ongoing administration and eases the burden of demonstrating regulatory compliance.

Together, SAP and Thales make it easy to demonstrate data sovereignty for SAP workloads on GCP through encryption key control and key lifecycle management via an intuitive 'single pane of glass.'

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.



About SAP

Headquartered in Walldorf, Germany, SAP is the world's largest inter-enterprise software company, and the world's third-largest independent software supplier overall. SAP employs over 28,909 people in more than 50 countries. Their professionals are dedicated to providing high-level customer support and services. For more information, visit www.sap.com.

Product Brief

CipherTrust Cloud Key Management

thalestct.com

THALES
Building a future we can all trust

Industry best practices as defined by the Cloud Security Alliance (CSA) require that keys be stored and managed outside of the cloud service provider and the associated encryption operations¹. Cloud Service Providers (CSPs) can comply with best practices by offering Bring Your Own Key (BYOK) or Hold Your Own Key (HYOK) services to enable customer control of the keys used to encrypt their data. Customer control of the keys allows for the separation, creation, ownership and control, including revocation, of encryption keys or tenant secrets used to create the keys.

CipherTrust Cloud Key Management (CCKM) increases efficiency by reducing the operational burden—even when all of the cloud keys are native keys. Giving customers lifecycle control, centralized management within and among clouds, and visibility of cloud encryption keys reduces key management complexity and operational costs. Customers report that they stepped away from managing keys across a heterogeneous environment and invested in CCKM to enable them to move securely to the cloud—and their cloud use is growing exponentially, reducing management overhead and the potential for security holes.

Control Cloud Encryption Keys

- Leverage the value of BYOK and HYOK services with full-lifecycle cloud encryption key management
- Gain higher efficiency with centralized key management across hybrid, single- and multi-cloud environments, including key discovery, management of native cloud keys and automated key rotation
- Comply with the most stringent data protection mandates with secure key origination
- Amplify the benefits of native keys by using a robust multi-cloud platform with outstanding UI

Best Practices

Industry and internal data protection mandates require increasing protection for sensitive cloud data. Meanwhile, CSA and industry analysts state that cloud encryption keys should be managed by customers. Key management systems (KMSs) can grow to hundreds of master keys per KMS—requiring knowledge of how, when, and by whom encryption keys are used, in addition to backing up, monitoring, rotating, expiring, archiving, suspending, restoring, and destroying keys as needed. CCKM provides comprehensive key lifecycle management to provide visibility on a single pane of glass, and automate and fulfill requirements for safe, comprehensive key management across multiple clouds.

Improved Efficiency

CCKM offers multiple capabilities in support of increasing efficiency:

- Secure centralized key management for Native, BYOK and HYOK cloud keys from a single browser window, across multiple clouds, regions, accounts, subscriptions, projects, applications, org ids, etc.
- Automated synchronization ensuring that cloud console-specific key operations are visible in centralized key management—even if you have already created thousands of native cloud keys at your cloud provider
- Automated key rotation with support for expiring keys ensuring compliance while saving valuable time
- Metadata is collected and laid out in the same order for every cloud provider, removing the need to look for data in disparate places

Single Pane of Glass

Access to each cloud provider from a single console, across multiple accounts, regions, subscriptions and projects makes it easier for organizations to understand how their workloads across different clouds are protected. We continually increase key visibility to make it easier for administrators to manage and control access to the keys in minutes instead of days, and stop threats faster.

Strong Encryption Key Security

Customer key control requires secure key generation and storage. CCKM leverages the security of the CipherTrust Manager, Luna Network HSM, or the Vormetric Data Security Manager (DSM) to create keys with up to FIPS 140-2 Level 3 security.

Hold Your Own Key

CCKM services respond to encryption key requests from cloud providers, supporting many of the emerging HYOK offerings: AWS External Key Store (XKS); Google Cloud External Key Management (EKM), Google EKM Ubiquitous Data Encryption (UDE), and Google Workspace Client-side Encryption for Drive, Gmail, Google Calendar and Google Meet; Microsoft DKE, OCI EKMS, and Salesforce Cached Keys.

¹ See CSA CCM EKM-04.

The Compliance Tools You Need

Visibility reports demonstrate compliance to the regulator for mission-critical workloads. Logs may also be directed to a syslog server or Security Information and Event Management (SIEM) tool.

Automation Tools Support Your Initiatives

CCKM capabilities are available programmatically using RESTful APIs, enabling the power of centralized cloud encryption management to work with your automation and self-service initiatives. Interactively explore the APIs with our built-in API playground.

Flexible Deployment Options

Deployment environments include: public cloud, private cloud, hybrid cloud, physical appliances and an as-a cloud-based subscription service.

Regardless of how and where CCKM is deployed, CCKM can manage keys and access to the keys, in any supported cloud.

To meet different organization’s needs, CCKM is available in virtual and physical form factors: Virtual CCKM is an all-software offering easily deployed and can be run in the cloud or on premises and may be found in several cloud provider marketplaces for fast instantiation. Physical appliances are available for customers who prefer an on-prem solution. CCKM service is available on-demand through the Thales Data Protection on Demand (DPoD) marketplace and in CSP marketplaces.

Multi-cloud Data Security Solutions

CCKM simplifies the need to control and manage encryption keys for cloud services, a critical solution for fulfilling industry and organizational data protection mandates. CCKM and many additional Thales multi-cloud security products, including various Bring Your Own Advanced Encryption offerings, all with centralized key management, offer you choices on how best to protect all your cloud data. It’s your data in their cloud.

Supported clouds and key management ownership models:

Increasing Customer Control

Amazon Web Services (AWS) KMS	Native	BYOK	
AWS CloudHSM	Native		
AWS XKS			HYOK
AWS GovCloud	Native	BYOK	HYOK
Google Cloud Platform CMEK	Native	BYOK	
Google Cloud Platform EKM			HYOK
Google Cloud Platform EKM UDE			HYOK -CC ¹
Google Workspace CSE			HYOK
Microsoft Azure Cloud	Native	BYOK	
Microsoft Azure GovCloud	Native	BYOK	
Microsoft Azure Managed HSMs	Native	BYOK	
Microsoft Office 365		BYOK	HYOK ²
Oracle Cloud Infrastructure	Native	BYOK	HYOK
Oracle Cloud for Government	Native	BYOK	HYOK
Oracle US Defense Cloud	Native	BYOK	HYOK
Oracle National Security Regions	Native	BYOK	
Salesforce.com	Native	BYOK	HYOK ³
Salesforce GovCloud Plus	Native	BYOK	HYOK ³
Salesforce Sandbox	Native	BYOK	HYOK ³
SAP Data Custodian	Native	BYOK	HYOK ⁴

¹ HYOK-CC is HYOK for Confidential Computing

² Microsoft Double Key Encryption (DKE)

³ Cache-only Key Service

⁴ Coming Soon

About Thales Trusted Cyber Technologies

Thales Trusted Cyber Technologies, a business area of Thales Defense & Security, Inc., is a trusted, U.S. provider of cybersecurity solutions dedicated to U.S. Government. We protect the government’s most vital data from the core to the cloud to the edge with a unified approach to data protection. Our solutions reduce the risks associated with the most critical attack vectors and address the government’s most stringent encryption, key management, and access control requirements.

For more information, visit www.thalestct.com



CipherTrust Manager

Version: v2.22 (latest)

4 February 2026

Contents

CipherTrust Cloud Key Manager (CCKM)	5
Overview	6
AWS Resources	20
Managing AWS Accounts	30
Managing AWS Keys	39
Scheduling Operations	122
Managing AWS Reports	134
Managing Policies	138
Managing AWS Templates	142
Migrating to IAM Roles Anywhere Connections	144
Downloading AWS Metadata	144
AWS CloudHSM Key Store Resources	145
Managing an AWS CloudHSM Key Store from CCKM	146
AWS External Key Store Resources	150
Managing External Key Store Resources	159
AWS XKS Performance Summary	176
Microsoft Azure Resources	199
Prerequisites	200
Prerequisites for Azure Cloud	201
Prerequisites for Azure Stack Cloud with Azure AD	205
Prerequisites for Azure Stack Cloud with Azure ADFS	210
Managing Azure Vaults	213
Managing Azure Keys	220
Managing Azure Certificates	271
Managing Azure Secrets	282
Scheduling Operations	289
Managing Azure Reports	299
Viewing Azure Subscriptions	303
Performance Summary	303
Allowing AD Users to Manage Azure Vaults	310
Configuring Connection to Azure Using Private Link	311
Downloading Azure Metadata	312
Microsoft Double Key Encryption (DKE) Resources	313
Managing DKE Endpoints	326
Managing DKE Authorized Tenants	335
Microsoft DKE service performance summary	341
External CipherTrust Resources	365
Managing External CipherTrust Domains	365
Managing External CipherTrust Keys	369
DSM Resources	376
Managing DSM Domains	378
Managing DSM Keys	383

Scheduling Operations	387
Google Cloud Resources	391
Google Cloud Projects	396
Google Cloud Key Rings	401
Google Cloud Keys	406
Google Cloud Reports	453
Scheduling Operations	456
Downloading Google Metadata	464
Performance Summary	465
Google Cloud External Key Manager Resources	475
Managing Google EKM Endpoints	479
Managing Google EKM Endpoint Policies	493
Google EKM Performance Summary	496
Managing Google EKM Cryptospaces	508
Managing Google EKM Cryptospace Endpoints	522
Google Workspace CSE Resources	525
Access Policies	526
High Level Architecture	526
Licensing	529
Prerequisites	529
Communication with KACLS	537
CSE Guest Access Support for Google Meet and Google Drive	538
Google CSE support for Google's end-to-end encrypted email	540
Clustering for Google Workspace	542
Encrypting Data Encryption Keys	542
Decrypting Data Encryption Keys	549
Endpoints	553
Identity Providers	559
Viewing Google Workspace CSE Records	560
Performance Summary	561
Google Calendar	563
Google Drive	579
Google Meet	594
Gmail	612
Key Material Export and Upload	618
Luna HSM Resources	620
Managing Luna HSM Partitions	622
Managing Luna HSM Keys	628
Scheduling Operations	633
Oracle Cloud Resources	637
Managing Oracle Vaults	643
Managing Oracle Keys	649
Managing Oracle Tenancies	675
Scheduling Operations	677
Managing Oracle Reports	684

Managing Oracle Compartments	687
Downloading Oracle Metadata	688
OCI External Resources	689
Managing Identity Providers	694
Managing External Vaults	698
Managing External Keys	705
OCI EKMS Performance Summary	711
SAP Resources	743
Managing SAP Groups	749
Managing SAP Keys	755
Scheduling Operations	792
Managing SAP Reports	798
Viewing Linked SAP Applications	801
Performance Summary	801
Downloading SAP Metadata	809
SAP HYOK Resources	810
Licensing	813
SAP HYOK endpoints	813
SAP HYOK keystores	820
Scheduling Operations	828
Salesforce Resources	832
Managing Salesforce Organizations	844
Managing Salesforce Tenant Secrets	850
Scheduling Operations	868
Managing Salesforce Reports	874
Managing Salesforce Cache-Only Key Endpoints	876
Managing Certificates from Salesforce Organizations	881
Migrate from CCKM Appliance	883
Backup and Restore	942

CipherTrust Cloud Key Manager (CCKM)

This document explains how to:

- Add AWS KMS accounts, Azure vaults, Luna HSM partitions, DSM domains, Google Cloud key rings, Salesforce organizations, SAP Data Custodian groups, Oracle Cloud Infrastructure (OCI) vaults, OCI External Key Management (EKM) (or external) vaults, CipherTrust (external) domains, SAP EKM to CCKM
- Manage AWS, Azure, Luna HSM, DSM, Google CMEK, Google Cloud EKM keys, Salesforce tenant secrets, SAP, OCI keys, OCI EKM keys, CipherTrust (external) keys, Microsoft Double Key Encryption (DKE) endpoints, SAP EKM keys on CCKM
- Manage AWS custom key stores on CCKM
- Schedule key rotation for AWS, Azure, Google Cloud, Salesforce, SAP, SAP EKM, Microsoft DKE, and OCI keys
- Schedule key refresh for Luna HSM, DSM, AWS, Azure, Google Cloud, Salesforce, SAP, and OCI keys

It is assumed, for this document, that you have already configured the CipherTrust Manager appliance. Refer to the [CipherTrust Manager Deployment Guide](#) for instructions.

The next step is to activate and install the CCKM license. Refer to [Licensing](#) for details.

Workflow

This section outlines the general process for managing keys with CCKM. Note that the specific steps for HYOK resources may vary; consult their respective documentation for details.

1. **Add a connection** between the CipherTrust Manager and the desired cloud or key source. Refer to [Connection Manager](#) for details.

This is needed to grant CCKM the access to any of the cloud service or key source users with valid user credentials.

2. **Test the connection.** The connection must be in the ready state.
3. **Add the container** that contains the keys to be managed. A container can be:
 - An AWS account
 - An Azure vault
 - An external CipherTrust Manager domain
 - A Luna HSM partition
 - A DSM domain

-
- A Google Cloud key ring
 - A Salesforce organization
 - A SAP group
 - An OCI vault

When adding a container, you need to select the corresponding connection.

4. **Refresh the container** to download its keys to CCKM. Refreshing a container might take significant amount of time depending on the number of keys stored in it.

CCKM provides options to refresh keys of an individual or all containers of a cloud service or key source.

After a container is refreshed successfully, the downloaded keys can be managed on CCKM itself.

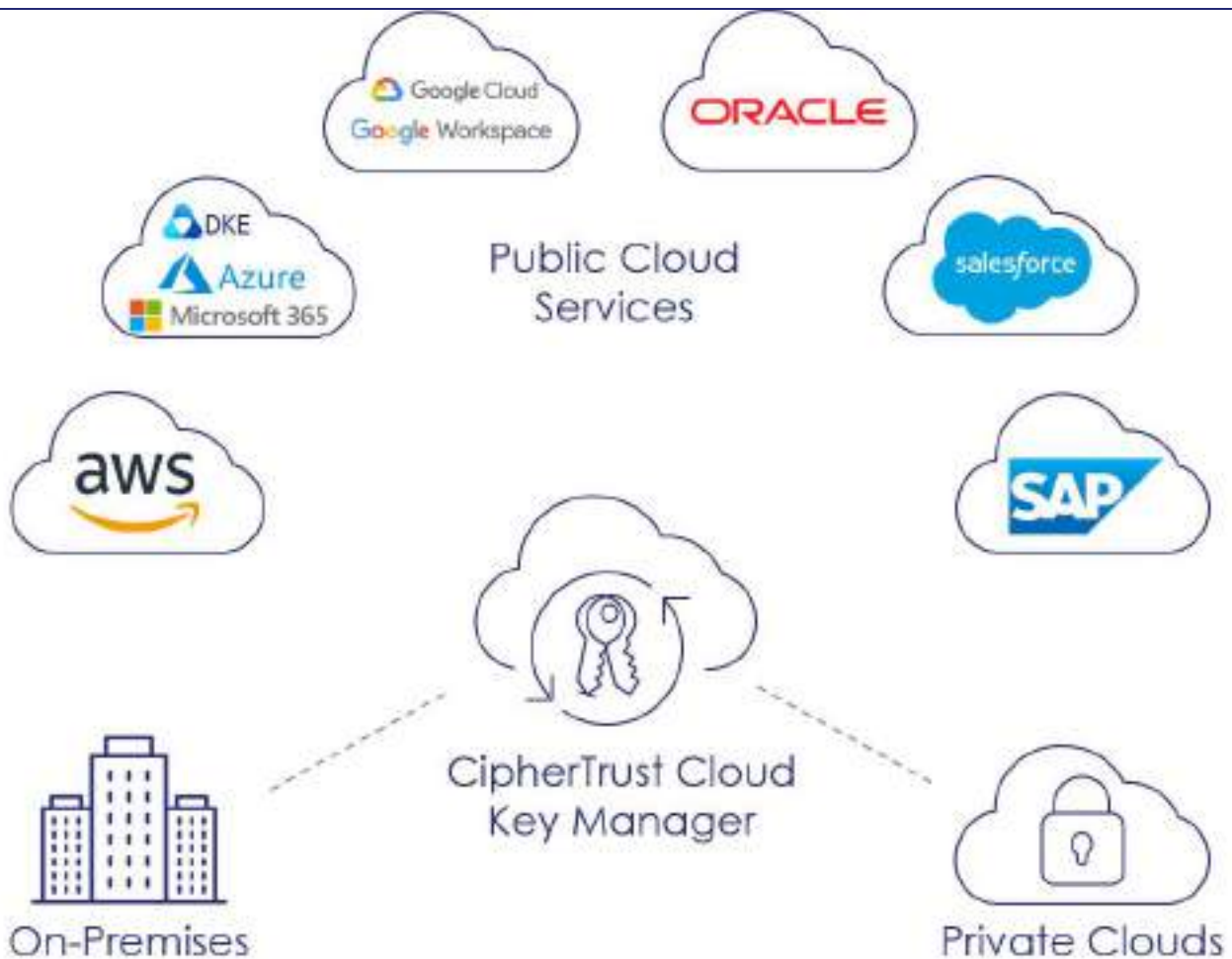
5. **Manage keys.** With CCKM, you can perform supported key operations such as adding, editing, and rotating keys.

CCKM also provides options to schedule key operations and generate reports for the supported clouds. Refer to relevant sections in the [CCKM Administration](#) and [CCKM API](#) documentation for more details about the steps listed above and other CCKM features.

Overview

CipherTrust Cloud Key Manager (CCKM, also referred to as CCKM Embedded) centralizes the management of key life cycle for various cloud services providers. The CCKM complies with data security mandates in cloud storage environments while retaining the custodianship of the encryption keys. Enterprises can back up keys on-premise, destroy keys when no longer needed, and manage the entire life cycle of the cloud keys.

The following diagram shows the high level CCKM overview:



CCKM Components

The CCKM solution comprises the following components:

- CCKM GUI on the CipherTrust Manager for administrators and users
- At least one of the supported clouds
- A supported trusted key source
- A supported Internet browser

The product is delivered as a licensed component of the CipherTrust Manager appliance that can be installed on any one of the supported deployment methods.

Supported Clouds

- Amazon Web Services (AWS)
 - AWS China

-
- AWS GovCloud
 - Azure Cloud
 - Azure China Cloud
 - Azure US Government
 - Google Cloud
 - Oracle Cloud
 - Salesforce
 - Salesforce Sandbox
 - SAP Data Custodian

Note

- AWS China cloud does not support uploading 256-bit keys. It supports 128-bit keys only. CCKM automatically uploads 128-bit keys to the AWS China cloud through the GUI. Other AWS clouds support upload of 256-bit keys.
- AWS China cloud does not support creation of native asymmetric keys.
- Azure China cloud requires that the CipherTrust Manager and automated systems are deployed in the same VPC and subnet.

Supported Cloud Services

- AWS Customer Managed CMKs
- AWS Custom Key Stores
 - AWS CloudHSM Key Stores
 - External Key Stores (XKS)
- Azure Cloud BYOK
- Azure Key Vault Managed HSM
- Azure Stack (Azure Active Directory, Azure AD)
- Azure Stack (Active Directory Federation Services, AD FS)

-
- Google Workspace Client Side Encryption (CSE)
 - Google Cloud CustomerManaged Encryption Keys (CMEK)
 - Google Cloud External Key Manager (EKM)
 - Google Ubiquitous Data Encryption (EKM+UDE)
 - Microsoft Double Key Encryption (DKE)
 - Oracle Cloud Infrastructure (OCI) (BYOK)
 - OCI External Key Management Service (EKMS)
 - Salesforce Bring Your Own Key (BYOK)
 - Salesforce Cache-Only Key Service
 - SAP Data Custodian Key Management Service (BYOK)

Supported Key Sources

CCKM uses the following as the trusted key sources for the encryption keys employed within the supported clouds:

- **CipherTrust Manager**
- **External CipherTrust Manager**
- **Azure Dedicated HSM**
- **Luna Network HSM** (referred to as Luna HSM in this document): This release supports:
 - Luna Network HSM Software and Firmware versions v7.3.x and higher.
 - PED and Password-based HSM configurations.
 - Import of RSA-4096 keys from Luna HSM Software and Firmware versions 7.4.x and higher.
 - Symmetric and asymmetric keys with Luna HSM.

Note

- CCKM doesn't support FM-enabled Luna HSM as a key source.

- **TCT Luna T-Series Network HSM**: The release supports:
 - The TCT Luna T-Series Network HSM uses software version 7.13.2 and firmware version 7.11.2.

Note

- In the CCKM documentation, the HSM referred to as a key source includes both the Luna Network HSM and the TCT Luna T-Series Network HSM.

- **Data Security Manager (DSM):** This release supports DSM v6.4 Cumulative Patch 3 and higher.

Note

Data Security Manager (DSM) is EOL as of June 2024 and its support as a key source is being deprecated.

The CipherTrust Manager supports all clouds that CCKM supports. The CipherTrust Manager stores its own keys and the backup keys from the supported clouds.

Note

On the CipherTrust Manager, CCKM cannot manage source keys created on the CCKM Appliance v1.x.

Supported Deployment Methods

The CCKM is delivered as part of the CipherTrust Manager appliance. So, CCKM can be automatically deployed with the deployment of the CipherTrust Manager in the following supported environments:

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure
- Oracle Cloud
- Private Clouds - Oracle VMware vSphere, Microsoft Hyper-V, and OpenStack
- Physical Appliances

Refer to the [CipherTrust Manager Deployment Guide](#) for the complete list of supported environments and deployment instructions.

Supported Internet Browsers

The CCKM supports the following Internet browsers:

-
- Chrome 51.0.2704 (64-bit) or later
 - Firefox 45.0 or later
 - Microsoft Edge 91.0.864.37 or later

CCKM Functionality

The CCKM provides following functionalities for the supported cloud services:

- Life cycle management of keys, key versions, and attributes:
 - View Keys
 - Update Keys
 - Upload Keys
 - Rotate Keys
 - Delete Keys
- Disaster recovery of keys:
 - Backup Keys
 - Restore Keys
- Hybrid key management:
 - On-premise keys storage
 - Management of both keys originating from trusted key sources and cloud-provider-sourced keys
 - Key synchronization
- Compliance management:
 - On-premise key storage with up to FIPS 140-2 Level 3 certification (CipherTrust Manager K570 with K7 card, Luna Network HSM, DSM)
 - Key storage in public or private clouds, inaccessible to cloud services with FIPS 140-2 Level 1 (Luna Network HSM, DSM)
- Key visibility reporting:
 - Key Activity Report: Inspect individual key histories by operations, for example, when they were refreshed, rotated, edited, or deleted. Also, use this report to compare key activities between CCKM and a cloud service.

- **Key Aging Report:** Track keys by their expiration dates. Audit a range of dates, from past material deletions to future scheduled deletions, within a cloud service.
- **Service/Usage Report:** Monitor key usage by tracking services and applications consuming the keys. View when and where a service requests the use of each key.

Note

Reporting is not supported for the Azure Stack cloud. All clouds do not support all types of reports. Refer to related sections of the CCKM documentation for cloud-specific reports.

User Roles

CCKM has the following users with different responsibilities in administering and using the resources of supported clouds and key sources.

CCKM Admins

There is a System Defined Group named "CCKM Admins". Users within the "CCKM Admins" group are CCKM Administrators. Additionally, the CCKM administrators need the Key Users, Connection Admins, and User Admins permissions to perform key operations on the supported clouds.

A CCKM Administrator is responsible for creating and managing the following resources:

- AWS KMS Accounts, AWS Keys, AWS Custom Key Stores
- Azure Key Vaults, Azure Subscriptions, and Azure Keys
- Luna HSM Partitions, Luna Keys
- DSM Domains, DSM Keys
- Google Cloud Projects, Key Rings, and Keys
- Google EKM endpoints
- Microsoft DKE endpoints
- Salesforce Organizations, Tenant Secrets
- SAP Groups, SAP Keys
- CCKM Schedules
- CCKM Reports

CCKM Users

There is a System Defined Group named "CCKM Users". CCKM users registered with the CipherTrust Manager are part of this group. Additionally, the CCKM users need the Key Users permissions to perform key operations on the supported clouds. As well, they need custom key store permissions to manage AWS custom key stores.

Additional ACLs/Permissions

Additional ACLs/permissions can be granted to different CCKM roles on individual cloud resources such as [AWS Accounts](#), [Azure Key Vaults](#), [DSM Domains](#), [Google Key Rings](#), [Luna HSM Partitions](#), [Oracle Vaults](#), [Salesforce Organizations](#), and [SAP Groups](#).

Refer to the [CCKM documentation](#) of the respective clouds.

Support for Changing the Default Port of Web Interface Setting

The web interface port within CipherTrust Manager can be changed from the default port of **443** to another port. The web interface is used to serve both the GUI and the REST API in CCKM. Changing the web interface port does not affect the BYOK use cases in CCKM. However, it does affect the following cloud services:

- AWS XKS
- Google EKM
- Google Workspace CSE
- Microsoft DKE
- OCI EKMS (HYOK)
- SFDC Cache-only Key

If you plan to change the default port for the CipherTrust Manager web interface, ensure to change it **before** configuring these cloud services on CCKM.

Note

Changing the default port **after** configuring these cloud services is not supported.

To change the web interface port when configuring a CSE, DKE, Google EKM, or SFDC Cache-only endpoint, specify the new port number along with the hostname when adding the hostname. For example, `example.com:8443`.

In the case of AWS XKS and Google Cryptospace, AWS KMS and Google only accept and send requests on the default port of 443. If you wish to use a different port for the web interface other than the default, add a network component, such as a load balancer or a firewall, in front of CipherTrust Manager. Then configure port mapping on this component by mapping port 443 to the non-default port on the CipherTrust Manager you plan to use.

For more information about the CipherTrust Manager web interface, refer to [Interfaces](#).

Proxy Configuration

If you plan to run the CipherTrust Manager appliance behind a proxy, you must configure the proxy, as described in [Proxy Configuration](#). For example, you might need to add the proxy CA certificate to the trusted certificates list for the CCKM traffic that passes through a proxy and terminates TLS connection.

Note

- For CCKM, configure an HTTPS proxy.
- Upgrading the CipherTrust Manager to 2.22 will not disrupt any proxy connections added via root access in the earlier releases. However, to update the CA, use the new UI and APIs introduced in CipherTrust Manager 2.22. See [Trusted CAs in CipherTrust Manager](#) for more information.

For the list of URLs to be whitelisted, refer to [URLs to Whitelist for Running CipherTrust Manager Behind Proxy](#).

Now, you should be able to connect CCKM with the cloud through the proxy.

URLs to Whitelist for Running CipherTrust Manager Behind Proxy

The following URLs must be whitelisted before you run the CipherTrust Manager behind a proxy.

Amazon Web Services

Service	URL	Purpose
CloudWatchLogs	logs.<region-name>.amazonaws.com	Fetch AWS reports.
EC2	ec2.amazonaws.com	Required to fetch AWS regions. After all the regions are fetched, the service is no longer needed.
EC2	ec2.us-east-1.amazonaws.com	Required to fetch AWS regions. After all the regions are fetched, the service is no longer needed.
EC2	ec2.<region-name>.amazonaws.com	Required to fetch AWS regions. After all the regions are fetched, the service is no longer needed.
IAM	iam.amazonaws.com	Manage access to AWS services and resources securely.
KMS	kms.<region-name>.amazonaws.com	Communicate with AWS KMS for key management.
STS	sts.amazonaws.com	Authenticate AWS users.
STS	sts.us-east-1.amazonaws.com	Authenticate AWS users.
STS	sts.<region-name>.amazonaws.com	Authenticate AWS users.

Service	URL	Purpose
AWS	aws.amazon.com	Redirects to service URLs.

AWS GovCloud

Service	URL	Purpose
CloudWatchLogs	logs.<region-name>.amazonaws.com	Fetch AWS reports.
EC2	ec2.amazonaws.com	Required to fetch AWS regions. After all the regions are fetched, the service is no longer needed.
EC2	ec2.us-gov-east-1.amazonaws.com	Required to fetch AWS regions. After all the regions are fetched, the service is no longer needed.
EC2	ec2.<region-name>.amazonaws.com	Required to fetch AWS regions. After all the regions are fetched, the service is no longer needed.
IAM	iam.amazonaws.com	Manage access to AWS services and resources securely.
KMS	kms.<region-name>.amazonaws.com	Communicate with AWS KMS for key management.
STS	sts.us-gov-east-1.amazonaws.com	Authenticate AWS users.
STS	sts.<region-name>.amazonaws.com	Authenticate AWS users.

AWS China Cloud

Service	URL	Purpose
CloudWatchLogs	logs.<region-name>.amazonaws.com.cn	Fetch AWS reports.
EC2	ec2.amazonaws.com.cn	Required to fetch AWS regions. After all the regions are fetched, the service is no longer needed.
EC2	ec2.cn-north-1.amazonaws.com.cn	Required to fetch AWS regions. After all the regions are fetched, the service is no longer needed.
EC2	ec2.<region-name>.amazonaws.com.cn	Required to fetch AWS regions. After all the regions are fetched, the service is no longer needed.
IAM	iam.amazonaws.com	Manage access to AWS services and resources securely.
KMS	kms.<region-name>.amazonaws.com.cn	Communicate with AWS KMS for key management.

Service	URL	Purpose
STS	sts.cn-north-1.amazonaws.com.cn	Authenticate AWS users.
STS	sts.<region-name>.amazonaws.com.cn	Authenticate AWS users.

Azure Public Cloud

URL	Purpose
management.azure.com	Fetch subscriptions and key vaults
login.microsoftonline.com	Authenticate Azure credentials
graph.windows.net	For Azure Stack
vault.azure.net	Manage keys
*.vault.azure.net	Manage keys
api.loganalytics.io	Fetch Azure reports
managedhsm.azure.net	Manage keys
*.managedhsm.azure.net	Manage keys

Azure Public Cloud (for Microsoft DKE)

URL	Purpose
login.microsoftonline.com	Authenticate Azure credentials
graph.microsoft.com	Fetch users and user roles

Azure US Gov Cloud

URL	Purpose
management.core.usgovcloudapi.net	Fetch subscriptions and key vaults
management.usgovcloudapi.net	Fetch subscriptions and key vaults
login.microsoftonline.us	Authenticate Azure credentials
vault.usgovcloudapi.net	Manage keys
*.vault.usgovcloudapi.net	Manage keys
graph.windows.net	For Azure Stack
api.loganalytics.us	Fetch Azure reports

Azure China Cloud

URL	Purpose
management.core.chinacloudapi.cn	Fetch subscriptions and key vaults
management.chinacloudapi.cn	Fetch subscriptions and key vaults
login.chinacloudapi.cn	Authenticate Azure credentials
vault.azure.cn	Manage keys
*.vault.azure.cn	Manage keys
graph.chinacloudapi.cn	For Azure Stack

Google Cloud CMEK

URL	Purpose
https://oauth2.googleapis.com	Authenticate Google cloud service accounts
https://cloudresourcemanager.googleapis.com	Fetch Google cloud projects
https://cloudkms.googleapis.com	Manage Google cloud keys, and fetch Google cloud locations and key rings
https://logging.googleapis.com	Fetch Google cloud reports

Google Cloud EKM

URL	Purpose
https://www.googleapis.com/	Verify and validate authentication and authorization JWTs
https://compute.googleapis.com	Creates and runs virtual machines on Google Cloud platforms. Required for EKM Ubiquitous Data Encryption (UDE) upstream verification.

Google Workspace CSE

URL	Purpose
https://www.googleapis.com/	Verify and validate authorization JWTs
https://admin.google.com	• Configure endpoint URLs and Identity Provider (IdP) settings on the Google Admin Console • Cross-Origin Resource Sharing (CORS) URL to allow requests from the Origin URL
https://krahsc.google.com	CORS URL to allow requests from the Origin URL
https://docs.google.com	Encrypt and decrypt Google Docs
https://drive.google.com	Encrypt and decrypt Google Drive

URL	Purpose
https://meet.google.com	Encrypt and decrypt Google Meet calls
https://calendar.google.com	Encrypt and decrypt Google Calendar events
https://mail.google.com	Encrypt and decrypt Gmail messages
https://client-side-encryption.google.com	CORS URL to allow requests from the Origin URL
Identity Provider URL	Verify and validate the authentication JWTs

The following redirect callback URIs need to be configured on the *IdP application* (for example, `auth0`) for authentication.

- https://client-side-encryption.google.com/callback
- https://client-side-encryption.google.com/oidc/cse/callback
- https://client-side-encryption.google.com/oidc/drive/callback
- https://client-side-encryption.google.com/oidc/gmail/callback
- https://client-side-encryption.google.com/oidc/meet/callback
- https://client-side-encryption.google.com/oidc/calendar/callback
- https://client-side-encryption.google.com/oidc/docs/callback
- https://client-side-encryption.google.com/oidc/sheets/callback
- https://client-side-encryption.google.com/oidc/slides/callback

Oracle Cloud

Service	URL	Purpose
Audit	audit.<region>.oraclecloud.com	Fetch OCI audit logs that will be used for OCI report generation.
Identity	identity.<region>.oci.oraclecloud.com	Required to test connection and fetch Oracle compartments and other related resources.
Identity Domain	<identity domain URL>	Required to verify and validate the authorization JWT.
Vault (KMS)	kms.<region>.oraclecloud.com	Communicate with OCI vaults for key management.
Object Storage	objectstorage.<region>.oraclecloud.com	Communicate with OCI buckets for key backup/restore.

Salesforce Cloud

URL	Purpose
login.salesforce.com	Authenticate Salesforce user credentials
test.salesforce.com	Verify SFDC Sandbox users (used to test connection with the Salesforce Sandbox)
<YourInstance>.salesforce.com	Manage keys
<salesforce hostname URL>	Authentication and managing keys

SAP Data Custodian

Service	URL	Purpose
KMS	<instance-subdomain>.datacustodian.cloud.sap	Communicate with SAP KMS for authorization, audit, cryptography, keys, groups, and backups.

Connecting CCKM in Amazon Virtual Private Cloud (VPC)

When AWS KMS is in Amazon VPC:

- IAM is not supported, as it is a global AWS service. Therefore, CCKM users cannot fetch IAM roles and users.
- CCKM cannot connect with the EC2, KMS, Logs, and STS services using public IP addresses. If their endpoints are not added on AWS, you cannot add AWS connections and AWS keys on the CipherTrust Manager. For successful connection, add the IP addresses of VPC endpoints for these services for your AWS regions to the DNS hosts on the CipherTrust Manager.

For example, you need to add the IP addresses of VPC endpoints of `sts.amazonaws.com`, `ec2.us-east-1.amazonaws.com`, `logs.us-east-1.amazonaws.com`, and `kms.us-east-1.amazonaws.com` service URLs to the DNS hosts list.

To add the IP address of a VPC endpoint:

On AWS Console

1. Create the VPC endpoint for a service.
2. Copy the DNS name of the service.
3. Ping the DNS name to find out the associated IP address.
4. Copy the IP address.

On CipherTrust Manager

1. Log on to the **CipherTrust Manager** GUI.

2. In the left pane, click **Admin Settings > DNS Hosts**.
3. Click **Add DNS Host**.
4. In the **Hostname** field, enter the service URL, for example, `sts.amazonaws.com`.
5. In the **IP Address** field, paste the copied IP address.
6. Click **Save**.

Similarly, add the DNS host entries for all three services, for all your AWS regions.

AWS Resources

This section describes prerequisites to manage AWS resources on the CCKM.

Prerequisites

- Before you can add an AWS account to the CCKM, an AWS connection must already exist on the CipherTrust Manager. A CipherTrust Manager administrator manages connections to external resources on the **Access Management > Connections Management** page of the CipherTrust Manager GUI. Refer to [Connection Manager](#) for details.
- Appropriate permissions to manage the AWS KMS must be added on the **AWS console**.
 - a. **Permissions to list regions:** Add the IAM permission `ec2:DescribeRegions` to list the AWS regions. For example:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "ec2:DescribeRegions",
      "Resource": "*"
    }
  ]
}
```

- b. **Permissions to manage AWS resources:** Add the following IAM permissions to manage AWS resources:
 - `kms:ListAliases`
 - `kms:ListKeyPolicies`
 - `kms:ListKeys`

-
- kms:ListResourceTags
 - kms:DescribeKey
 - kms:GetKeyPolicy
 - kms:GetKeyRotationStatus
 - kms:GetParametersForImport
 - kms:GetPublicKey
 - kms:TagResource
 - kms:UntagResource
 - kms:CancelKeyDeletion
 - kms:CreateAlias
 - kms:CreateKey
 - kms>DeleteAlias
 - kms:DeleteImportedKeyMaterial
 - kms:DisableKey
 - kms:DisableKeyRotation
 - kms:DescribeCustomKeyStores
 - kms:EnableKey
 - kms:EnableKeyRotation
 - kms:ImportKeyMaterial
 - kms:ListKeyRotations
 - kms:RotateKeyOnDemand
 - kms:ScheduleKeyDeletion
 - kms:UpdateAlias
 - kms:UpdateKeyDescription
 - kms:PutKeyPolicy
 - iam:ListGroups
-

- iam:ListRoles
- iam:ListUsers
- logs:DescribeLogGroups
- logs:FilterLogEvents

For example:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DisableKey",
        "kms:ListAliases",
        "kms:ListKeyPolicies",
        "kms:ListKeys",
        "kms:ListResourceTags",
        "kms:DescribeKey",
        "kms:GetKeyPolicy",
        "kms:GetKeyRotationStatus",
        "kms:GetParametersForImport",
        "kms:GetPublicKey",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:CancelKeyDeletion",
        "kms:CreateAlias",
        "kms:CreateKey",
        "kms>DeleteAlias",
        "kms>DeleteImportedKeyMaterial",
        "kms:DescribeCustomKeyStores",
        "kms:DisableKeyRotation",
        "kms:EnableKey",
        "kms:EnableKeyRotation",
        "kms:ImportKeyMaterial",
        "kms:ScheduleKeyDeletion",
        "kms:UpdateAlias",
        "kms:UpdateKeyDescription",
        "kms:PutKeyPolicy",
        "iam:ListGroups",
        "iam:ListRoles",
        "iam:ListUsers",
```



```

        "logs:DescribeLogGroups",
        "logs:FilterLogEvents"
    ],
    "Resource": "*"
}
]
}

```

Note

- To manage a multi-region key, an additional IAM permission `iam:CreateServiceLinkedRole` is required.
- To manage the External Custom Key Stores or CloudHSM Key Stores, additional IAM permissions required to use AWS resources are:
 - `cloudhsm:DescribeClusters`
 - `kms:CreateCustomKeyStore`
 - `kms:ConnectCustomKeyStore`
 - `kms>DeleteCustomKeyStore`
 - `kms:DisconnectCustomKeyStore`
 - `kms:UpdateCustomKeyStore`
 - `iam:CreateServiceLinkedRole`
- Permissions might take some time to be effective on AWS. Until then, a permission error might occur. Wait for some time and retry.

Now, [AWS accounts](#) and [AWS keys](#) can be managed on the CipherTrust Manager.

Note

- To use AWS IAM Roles Anywhere with CCKM, [additional configuration](#) is required.
- Grant permissions to the IAM user management account to use the Automatic Cloud Key Discovery schedule. Refer to the below section for details.
- AWS GovCloud does not support AWS IAM Roles Anywhere.

Additional Configuration for AWS IAM Roles Anywhere

The AWS IAM Roles Anywhere service allows non-federated identities outside AWS to assume IAM roles and use their permissions to access resources. The service provides a secure way for the workloads that run outside of AWS such as servers, containers, and applications to use X.509 digital certificates to obtain temporary AWS credential. This eliminates the need to manage long-term credentials for external workloads.

To use IAM Roles Anywhere, external workloads must use X.509 certificates issued by a Certificate Authority (CA). The CA needs to be registered as a trust anchor with the IAM Roles Anywhere service to establish trust between them. Alternatively, AWS Private Certificate Authority (AWS Private CA) can be used to create a CA and then use that to establish trust with IAM Roles Anywhere.

CA and Client Certificate Requirements

- Client certificates must satisfy the following requirements for authentication:
 - The certificates must be X.509v3
 - Basic constraints must include `CA: false`
 - The key usage must include Digital Signature
 - The signing algorithm must include SHA256 or a stronger algorithm (MD5 and SHA1 signing algorithms are rejected)
- Certificates used as trust anchors must satisfy the following requirements for signature algorithm:
 - The certificates must be X.509v3
 - Basic constraints **MUST** include `CA: true`
 - The key usage must include Certificate Sign, and may include CRL Sign
 - The signing algorithm must include SHA256 or a stronger algorithm (MD5 and SHA1 signing algorithms are rejected)
 - Certificate Revocation Lists (CRLs) are an optional feature of IAM Roles Anywhere

Configuration Steps

To use IAM Roles Anywhere for authentication to AWS from external workloads:

1. **Create a trust anchor.** This anchor is essentially a reference to a CA that IAM Roles Anywhere service will use to validate the authentication requests. Both the root and intermediate CAs can be used as trust anchors.
2. **Create an IAM role** that trusts the IAM Roles Anywhere service principal.
3. **Create a profile** that lists the roles IAM Roles Anywhere assumes. In the profile, you can limit the permissions for a created session with IAM managed policies.

By adding one or more roles to a profile and enabling IAM Roles Anywhere to assume these roles, a non-

AWS workload can use the client certificate issued by the trusted CA to make secure requests to AWS and get temporary credentials to access the AWS environment.

Note

- When the CipherTrust Manager is in AWS VPC, the trust anchor, and profile must be created in the same region from where the CipherTrust Manager instance is launched.
- Refer to the [AWS IAM Roles Anywhere](#) documentation for detailed instructions.

4. **Add an AWS connection** on the CipherTrust Manager with *IAM Roles Anywhere* enabled. While adding the connection for IAM Roles Anywhere, you need to specify:

- The AWS Resource Names (ARNs) of the trust anchor, IAM role, and profile created in the above steps
- Client's private key and certificate

Refer to [Connection Manager](#) for details.

Note

To migrate an AWS connection from using access keys and secrets to one that uses IAM roles, refer to [Migrating to IAM Roles Anywhere Connections](#).

Configurations to Discover all AWS accounts

To allow CCKM to discover all AWS accounts within an AWS Organization.

1. Create an IAM user in the management account.
2. Grant the IAM user the following permissions to manage KMS keys in member accounts through CCKM, as defined in the sample policy below:
 - List all organizational units (OUs) and accounts within the organization.
 - Assume a role in each member account.

Sample policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AutoDiscoveryOrgPermissions",
```

```

        "Effect": "Allow",
        "Action": [
            "organizations:DescribeOrganization",
            "organizations:ListAccounts",
            "organizations:ListChildren",
            "organizations:ListParents",
            "organizations:ListAccountsForParent",
            "organizations:ListOrganizationalUnitsForParent"
        ],
        "Resource": "*"
    },
    {
        "Sid": "AutoDiscoveryAssumeRolePermissions",
        "Effect": "Allow",
        "Action": [
            "sts:AssumeRole"
        ],
        "Resource": "arn:aws:iam::*:role/<Member Accounts role name e.g. AutoDiscoveryRole>"
    }
]
}

```

3. Create a trust relationship between the role in the member account and the user in the management account. Then, assign the necessary permissions as defined in the sample policy below.

Sample policy:

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<Management Account ID>:user/<Management Account IAM user name>"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}

```

4. Create an **AWS CloudFormation StackSet** (or simply **StackSet**) to automatically add a role to the member

accounts. Refer to [Create StackSet \(self-managed permissions\)](#) for details.

When creating a StackSet, select **Service-managed permissions**, update the management account ID, username, and member accounts' role name in the sample CloudFormation template provided below, and then upload the updated template.

Activate automatic deployment to allow StackSets to automatically create or delete CloudFormation stacks when a new AWS account joins or leaves the organization.

To use StackSets in AWS Organizations, the master account must enable data sharing between CloudFormation and Organizations. This can be done from the StackSets console. Once enabled, you can deploy stacks to all accounts in the organization or to specific organizational units (OUs). For more details, refer to [Activate Trusted Access with AWS Organizations](#).

Sample CloudFormation template:

```
{
  "Description": "Create IAM roles and policies to grant access to KMS",
  "Parameters": {
    "AwsAccountId": {
      "Type": "Number",
      "Default": "<Management Account ID>"
    },
    "AwsUserName": {
      "Type": "String",
      "Default": "<Management Account user name>"
    }
  },
  "Resources": {
    "AutoDiscoveryRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "RoleName": "AutoDiscoveryRole",
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "AWS": {
                  "Fn::Sub": "arn:aws:iam::${AwsAccountId}:u
ser/${AwsUserName}"
                }
              }
            }
          ]
        },
        "Action": "sts:AssumeRole"
      }
    }
  }
}
```

```

    }
  ]
}
},
"AutoDiscoveryKMSPolicy": {
  "Type": "AWS::IAM::Policy",
  "Properties": {
    "PolicyName": "AutoDiscoveryKMSPolicy",
    "PolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "ec2:DescribeRegions",
            "kms:DisableKey",
            "kms:ListAliases",
            "kms:ListKeyPolicies",
            "kms:ListKeys",
            "kms:ListResourceTags",
            "kms:DescribeKey",
            "kms:GetKeyPolicy",
            "kms:GetKeyRotationStatus",
            "kms:GetParametersForImport",
            "kms:GetPublicKey",
            "kms:TagResource",
            "kms:UntagResource",
            "kms:CancelKeyDeletion",
            "kms:CreateAlias",
            "kms:CreateKey",
            "kms>DeleteAlias",
            "kms>DeleteImportedKeyMaterial",
            "kms:DescribeCustomKeyStores",
            "kms:DisableKeyRotation",
            "kms:EnableKey",
            "kms:EnableKeyRotation",
            "kms:ImportKeyMaterial",
            "kms:ScheduleKeyDeletion",
            "kms:UpdateAlias",
            "kms:UpdateKeyDescription",
            "kms:PutKeyPolicy",
            "iam:ListGroups",
            "iam:ListRoles",

```

```

        "iam:ListUsers",
        "logs:DescribeLogGroups",
        "logs:FilterLogEvents"
    ],
    "Resource": "*"
}

],
},
"Roles": [
    {
        "Ref": "AutoDiscoveryRole"
    }
]
}
},
"Outputs": {
    "RoleId": {
        "Description": "The ID of the IAM role",
        "Value": {
            "Ref": "AutoDiscoveryRole"
        }
    },
    "RoleArn": {
        "Description": "The ARN of the IAM role",
        "Value": {
            "Fn::GetAtt": [
                "AutoDiscoveryRole",
                "Arn"
            ]
        }
    },
    "KmsPolicyId": {
        "Description": "The ID of the IAM policy for AWS KMS",
        "Value": {
            "Ref": "AutoDiscoveryKMSPolicy"
        }
    }
}
}
}

```

After all of the above mentioned configurations are done, you need to add an AWS connection in CCKM using the management account IAM user credentials.

Refer to [Connection Manager](#) for details.

Managing AWS Accounts

This section describes how to manage AWS accounts on the CCKM.

Before proceeding, make sure to fulfill [prerequisites](#).

Adding AWS Accounts

CCKM allows adding same AWS account in one CipherTrust Manager domain with different names, with each entry having a unique set of regions. By default, only the KMSs linked with the AWS account that is used for connection with the CipherTrust Manager can be added. However, if you want, you can assume a role within the same or a different account, while adding KMS over the same connection.

To add an AWS account to the CCKM:

1. Log on to the **CipherTrust Manager** GUI as administrator.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
4. Click **Add Account**. The Add AWS Account screen is displayed.
5. Select/enter the following details:
 - a. Specify a unique **Name**.
 - b. From the **AWS Connection** drop-down list, select the desired connection.

The **AWS Account ID** and **Available regions** of the selected AWS connection are displayed.

- c. In the **Available regions** section, select the desired regions.

By default, all the regions are selected. You can also use the **Search** box to filter the regions.

If you select a subset of available regions, then the remaining regions can be added under a different AWS KMS account name but under the same AWS Account ID.

- d. Click the right arrow button (). The selected regions move to the **Selected regions** list.

6. Click **Save**. The AWS account is added to the CCKM.

Adding AWS KMS Account and Regions by AssumeRole

CCKM allows adding same AWS account in one CipherTrust Manager domain with different names, with each entry having a unique set of regions. By default, only the KMSs linked with the AWS account that is used for connection with the CipherTrust Manager can be added. However, if you want, you can assume a role within the same or a different account, while adding KMS over the same connection. To do this, you need to specify the ARN and/or external ID of the role to be assumed.

Note

An AssumeRole provides a set of temporary security credentials that you can use to access AWS resources that you might not normally have access to. Refer to [AWS documentation](#) for details on AWS AssumeRole.

Tip

To configure AWS accounts for AssumeRole, refer to [IAM tutorial: Delegate access across AWS accounts using IAM roles](#).

To add an AWS account and regions by AssumeRole:

1. Log on to the **CipherTrust Manager** GUI as administrator.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
4. Click **Add Account**. The Add AWS Account screen is displayed.
5. Select/enter the following details:
 - a. Specify a unique **Name**.
 - b. From the **AWS Connection** drop-down list, select the desired connection.

The **AWS Account ID** and **Available regions** of the selected AWS connection are displayed.

- c. Specify the AssumeRole details:
 - a. Select **Add Assumed Role**. The following fields are displayed:
 - **Assume Role**: Specify the Amazon Resource Name (ARN) of the role to be assumed. This is a mandatory field.
 - **Assume Role External ID**: Specify the external ID for the role to be assumed.
 - b. Click **Test Connection**. When the connection is successful, the **AWS Account ID** associated with the assumed role is displayed. The **Available regions** section is populated with the available regions.
 - d. In the **Available regions** section, select the desired regions.

By default, all the regions are selected. You can also use the **Search** box to filter the regions.

If you select a subset of available regions, then the remaining regions can be added under a different AWS

KMS account name but under the same AWS Account ID.

e. Click the right arrow button (). The selected regions move to the **Selected regions** list.

6. Click **Save**. The AWS account is added to the CCKM.

Refreshing AWS Accounts

Note

With support for on-demand rotation of both native and imported BYOK keys, the refresh operation now takes longer than in previous releases. This is because detailed rotation information must be synchronized from AWS KMS to CCKM.

Refreshing is the process to download keys and key stores created on the AWS KMS to the CCKM. You can refresh keys and key stores from individual or all KMS accounts.

- [Refreshing Specific AWS Accounts](#)
- [Refreshing All AWS Accounts](#)

Refreshing Specific AWS Accounts

To refresh an AWS account:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed. This page displays the list of AWS accounts.
3. Click the overflow icon () corresponding to the desired AWS account and click **Refresh Now**.
4. On the Refresh Now screen, select the desired account regions to be refreshed.
5. Click **Refresh**.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > AWS > AWS Keys** page. Refer to [Viewing AWS Keys](#) for details.

Refreshed [CloudHSM key stores](#) and [external custom key stores](#) appear in the account details.

Refreshing All AWS Accounts

To refresh all AWS accounts:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed. This page displays the list of AWS accounts.
3. Click **Refresh All**.

Note

Refresh all KMS Accounts is a time intensive operation that could take several hours or days to complete. It will continue running in the background. Do you want to continue?

4. Click **Refresh All**.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > AWS > AWS Keys** page. Refer to [Viewing AWS Keys](#) for details.

Refreshed [CloudHSM key stores](#) and [external custom key stores](#) appear in the account details of individual accounts.

Viewing/Editing Details of AWS Accounts

The AWS KMS Accounts page shows the list of existing AWS KMS accounts. Search for the KMS accounts by KMS Name or KMS Account ID.

Viewing AWS Account Details

To view the details of an AWS account:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page displays the following details:

Column	Description
Name	Name of the AWS account.
Last Refreshed	When the AWS account was last refreshed. Never is displayed for accounts that are never refreshed.
Account ID	ID of the AWS account. If the role of another AWS user is assumed, then the associated account ID is displayed.
Connection	Name of the connection.
Cloud	Cloud name.

Column	Description
Regions	Regions in which the account is added. If the role of another AWS user is assumed, then regions associated with the account ID are displayed.
Date Added to CCKM	Date when the AWS KMS account was added to the CCKM.

The **Assume Role ARN** column is hidden by default. This column specifies Amazon Resource Name (ARN) of the assumed role. To show/hide a column, click the custom view icon () , select/clear the desired column, and click **OK**.

Editing AWS Accounts Details

To edit the details of an AWS account:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed. This page displays the list of added AWS accounts.
3. Click the overflow icon () corresponding to the desired AWS account and click **View/Edit Details**. The mini detail view shows the details of the AWS KMS account, such as account ID, connection, Assume Role ARN with its external ID, and associated regions etc.

On this page, you can perform the following:

- Manage user permissions on the AWS account: Refer to [Managing User Permissions on AWS Accounts](#) for details.
- Modify connection and regions: Refer to [Modifying Connection and Regions](#) for details.
- Add or remove assume role: Refer to [Adding or Removing Assume Role](#) for details.

Archiving AWS Accounts

You can archive an AWS account to switch it to read-only mode and to prevent it from using an active license.

You can recover and delete the archived AMS account using the **Recover** and **Delete** options, respectively. For more information, see [Recovering AWS Accounts](#) and [Deleting AWS Accounts](#).

To archive an AWS account:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the overflow icon () corresponding to the desired AWS account, then select **Archive**. The **Archive AWS Account** message appears.

4. Select **I wish to archive this account**.
5. Click **Archive**.

After this step, the **Status** of the archived AWS account changes to **ARCHIVED**, and automatic operations such as synchronization, key rotation, XKS credential rotation no longer function for this account.

Recovering AWS Accounts

You can recover an archived AWS account to make it editable and to activate its licensed version.

To recover an archived AWS account:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the overflow icon () corresponding to the desired AWS account, then select **Recover**. The **Recover AWS Account** message is displayed.
4. Select **I wish to recover this account**.
5. Click **Recover**.

After this step, the **Status** of the recovered AWS account changes to **ACTIVE**.

Managing User Permissions on AWS Accounts

To work with AWS, users/groups must have the minimum set of permissions that allow them to use the AWS resources such as keys, AWS KMS, and reports. Initially, the user only has permission to view the keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform actions on the AWS account. Refer to [User Roles](#) for details.

To add permission for user/group:

1. In the **ACCESS CONTROL** section, click **Assign User/Group**.
2. On the **Assign User/Group** screen, select the user or group to be assigned permissions from the **User/Group** drop-down list.
3. Click **Save**.

The newly added user/group is displayed under **Name** in the **Access Control** section.

Note

Use the All Keys, BYOK Keys, Native Keys, HYOK Keys, Custom Key Store, and Reports tabs to specify permissions for the different key types, custom key store, and reports. These tabs display the operations allowed on the AWS accounts. The default tab All Keys shows permissions for all keys.

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. Under the **ACCESS CONTROL** section, click the desired tab.
2. Select the check box under the desired operation corresponding to the desired users or groups.

Note

On the BYOK Keys, HYOK keys, and Native Keys tab, the **View Key** permission is required for granting the **Add Key** permission. If you grant the **Add Key** permission, the **View Key** permission is automatically added, even if it not granted already.

Similarly on the Custom Key Store tab, the **View** permission is required for granting the **Add** permission. If you grant the **Add** permission, the **View** permission is automatically added, even if it is not granted already.

3. Click **Update**.

A success message is displayed on the screen.

Removing a Permission

To remove a permission assigned to a user or group:

1. Under the **ACCESS CONTROL** section, click the desired tab.
2. Clear the check box under the desired operation corresponding to the desired users or groups.
3. Click **Update**.

A success message is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. Under the **ACCESS CONTROL** section, click the desired tab.
2. Under **Remove**, click the **X** button corresponding to the desired user/group.
3. On the **Remove User / Remove Group** screen, click **Remove**.

Note

Removing this user/group will remove all permissions currently assigned to the user/group. Are you sure you want to continue?

4. Click **Remove**.

This step removes the explicitly added permissions and restores the default permission for the user.

Modifying Connection and Regions

You can update AWS connection and AssumeRole credentials as long as they point to the same AWS account.

To change the connection linked with the AWS account:

1. Under **GENERAL INFO**, from the **AWS Connection** drop-down list, select the desired connection.
2. Click **Update**.

To add regions to the AWS account:

1. Under **GENERAL INFO > Select Regions**, in the **All Available Regions** section, select the desired regions.
2. Click the right arrow button (). The selected regions move to the **All Selected Regions** list.
3. Click **Update**.

To remove regions from the AWS account:

1. Under **GENERAL INFO > Select Regions**, in the **All Selected Regions** section, select the regions to be removed.
2. Click the left arrow button (). The selected regions move to the **All Available Regions** list.
3. Click **Update**.

Adding or Removing Assume Role

You can update AWS connection and AssumeRole credentials as long as they point to the same AWS account.

To add an Assume Role:

1. Under **GENERAL INFO**, select **Add Assume Role**.
2. Select **Add Assumed Role**. The following fields are displayed:
 - **Assume Role**: Specify the Amazon Resource Name (ARN) of the role to be assumed. This is a mandatory field.
 - **Assume Role External ID**: Specify the external ID for the role to be assumed.
3. Click **Test Connection**. When the connection is successful, the **All Available Regions** section is populated with available regions.
4. Add or remove regions as appropriate.
5. Click **Update**.

To change an Assume Role:

1. Under **GENERAL INFO**, change the following details:
 - **Assume Role**: Specify the Amazon Resource Name (ARN) of the role to be assumed. This is a mandatory field.
 - **Assume Role External ID**: Specify the external ID for the role to be assumed.
2. Click **Test Connection**.
3. Add or remove regions as appropriate.
4. Click **Update**.

Deleting AWS Accounts

To delete an AWS KMS account:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed. This page displays the list of added AWS KMS accounts.
3. Click the overflow icon () corresponding to the desired AWS account and click **Delete**. The **Delete AWS Account** message is displayed.
4. Select **I wish to delete this account**.
5. Click **Delete Account**.

Note

After an AWS account is deleted from the CipherTrust Manager, the keys existing in the AWS KMS account (native and BYOK) are not affected. However, you can no longer manage those keys from CCKM. The AWS services using the AWS KMS keys continue to function without any issues.

Managing AWS Keys

This section describes how to manage AWS keys on CipherTrust Cloud Key Manager (CCKM). Before proceeding, you must have an AWS account added to the CCKM. Refer to [Managing AWS Accounts](#) for details.

Note

AWS-managed keys can't be managed by CCKM. Any key management operations on such keys from CCKM will not succeed.

Source Types

For adding AWS keys, CCKM supports the following key material sources:

- **Native:** Create AWS key material directly with native AWS application. Refer to [Creating Native Key Material](#) for details.
- **External (BYOK):** External (Bring Your Own Key). Add key material by creating or uploading new source key from an external source. Refer to [Adding Key Material Using External \(BYOK\) Source](#) for details. You can select CipherTrust Manager (External), CipherTrust Manager (Local), or Vormetric Data Security Manager (DSM) as an external key source, or Decide Later.
- **CloudHSM Key Store:** Create a CloudHSM key from CCKM using key material from AWS CloudHSM, which is the key source.
- **External Custom Key Store (HYOK):** External Custom Key Store (Hold Your Own Key). Add an HYOK key tied to key material stored in a Luna HSM or a CipherTrust Manager depending on which key source you are using. AWS Key Management Services (KMS) communicates to CCKM, which uses the AWS HYOK key as an intermediary to the source key stored in a Luna HSM or CipherTrust Manager. Depending on which key source you are using, Luna or CipherTrust Manager executes the cryptographic operations. You can rotate the HYOK key, which associates a new key in the Luna HSM or associates a new version to the CipherTrust Manager key.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

There are different operations available for HYOK keys than for other key types. You can perform the following HYOK key operations:

- Create an HYOK key
- Linking HYOK keys
- View HYOK key details
- Apply a Key Rotation Schedule
- Adding/Editing Policies
- Enable and disable
- Manually delete an Unlinked HYOK key
- Scheduling Key Deletion
- Manually rotate an HYOK Key
- Block or unblock an HYOK key

Regionality

When creating an AWS Native key or External (BYOK) key, you can specify whether the key is a single-region key or a multi-region key.

Note

This functionality is applicable to Native and BYOK keys.

- **Single-Region Key:** A single-region key cannot be replicated in other AWS regions.
- **Multi-Region Key:** A multi-region key can be replicated in multiple AWS regions. By default, the base key will be referred to as the multi-region primary key. However, any replica of the multi-region key can be set as the primary key later.

Note

After a key is created, its regionality cannot be changed.

Creating Native Key Material

To create AWS key material directly with native AWS application:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**.
3. Click **Add Key**. The Key Material Origin screen of the Add AWS Key wizard is displayed.

Key Material Origin

1. Select the desired **AWS Account** from the drop-down list.
2. Select the desired **Region** from the drop-down list. The list shows the regions in which the selected AWS account is available.
3. Select **Native** as the **Origin Type**.
4. Select **Regionality**. The options are:
 - **Single-Region Key**: A single-region key cannot be replicated in other AWS regions.
 - **Multi-Region Key**: A multi-region key can be replicated in multiple AWS regions. By default, the base key will be referred to as the multi-region primary key. However, any replica of the multi-region key can be set as the primary key later.

Note

After a key is created, its regionality cannot be changed.

5. Click **Next**. The Destination (AWS) Key screen is displayed.

Destination (AWS) Key

1. Under **Select Key Type**, select the desired key type. The options are:
 - **Symmetric**: One key does encryption and decryption.
 - **Asymmetric**: A key pair does encryption and decryption.

2. Under **Select Key Usage**, select the key usage:

- **Encrypt and Decrypt:**
 - For symmetric keys, the key is used only to encrypt and decrypt the data.
 - For asymmetric keys, the public key is used to encrypt while the private key is used to decrypt.
- (Symmetric keys only) **Generate and Verify MAC:** The key is used only to generate and verify hash-based message authentication codes (HMACs).
- (Asymmetric keys only) **Sign and Verify:** In this key pair, the public key is used to sign while the private key is used to verify.

3. Enter a user-friendly **Alias** for the key. This helps uniquely identify a key.

4. (Optional) Provide a brief **Description** of the key.

5. Select the desired **Key Algorithm** from the drop-down list.

Note

The **Algorithm** field is not displayed for symmetric keys when their Key Usage is selected as **Encrypt and Decrypt**.

6. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.

CCKM allows the following characters in tag values:

- Alphanumeric characters
- Special characters **** _ . / = + - @ ****

- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

7. Click **Next**. The Key Policy screen is displayed.

On the **Key Policy** screen, you can specify a policy for the key. If you skip the Key Policy screen by clicking **Next**, a default policy will be applied to the key.

Key Policy

Specify a policy for the key. You can create a new policy, select a saved policy, and build from template.

Click the desired tab to view the instructions.

Create New Policy

1. Select the **Create New Policy** option.
2. Select a policy view. You can select either the Basic view or Raw view. The default view is Basic.

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

In the Basic view, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.
- d. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

3. Select **Save this Key Policy** and **Enter Template Name**.
4. Click **Next**.

Select Saved Policy

1. Select the **Select Saved Policy** option.
2. Select a policy from the **Saved Policies** drop-down list.

You can also update the selected policy, and create a new policy using it.

- a. Select the policy view, you can select **Raw View** or **Basic View**. By default, the policy will open in the Raw

View.

- b. Update the policy.

Raw View

Make the necessary changes in the policy.

Basic View

In the Basic View, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.

Note

If you make changes in one of the views and switch to the other view without updating the policy, you will see a **Warning** message. Click **Yes** to continue.

- c. Click **Update**. The **Update Policy** message is displayed.

To save the changes as a new policy, select the **DO NOT Push Changes** check box, and enter the **New Policy** name.

Note

If the selected policy is Unverified, the Update Policy message will not be displayed, a new policy will not be created.

- d. Click **Apply**.
- e. Click **Next**.

Build From Template

1. Select the **Build From Template** option.
2. **Select Template**.
3. (Optional) Make the necessary changes in the template.

4. Click **Next**.

Based on the selection of the key type and key usage, the Add to Schedule screen or AWS Automatic Key Rotation screen is displayed. Click the desired tab to view the instructions.

Add to Schedule

Note

If Asymmetric is selected as the Key Type when specifying the Destination Key details, the key cannot be scheduled.

1. From the **Rotation** drop-down list, select a schedule to apply.

Note

If you're creating an imported AES symmetric key and the selected schedule has "Rotate Material (imported AES keys)" enabled, the key material will be rotated automatically when the schedule runs. For all other cases (including schedules without this option enabled), a new key will be created on rotation, and the alias will be reassigned to the new key.

2. (Optional) For BYOK or Native keys, you can grant or deny the encrypt permission to the key policy in the AWS account where the key is created in OR to the key policy in *all accounts* the key is shared in:

- Select **Disable Encrypt Permissions on Current Key** to deny the encrypt permission to the key policy in the AWS account where the key is created in. Clear the check box to grant the encrypt permission. By default, **Disable Encrypt Permissions on Current Key** is selected, that is, the encrypt permission is denied to the current key.
- Select **Disable Encrypt Permissions on All Accounts** to deny the encrypt permission to the key policy in all the AWS accounts the key is shared in. Clear the check box to grant the encrypt permission.

3. Select the **Key Origin** from the available options. The key origin can be:

- **CipherTrust**: CipherTrust Manager.
- **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
- **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.

4. Click **Next**.

The Review and Add Key screen is displayed.

AWS Automatic Key Rotation

Note

Automatic key rotation only applies to the AWS native keys that have the algorithm as SYMMETRIC_DEFAULT.

Automatically rotate AWS native keys after a specified time period. To enable automatic key rotation:

1. Enable **automatic key rotation**
2. Enter the **Rotation period (in days)**.

Based on the entered rotation period, the next rotation date will be displayed. The default rotation period is 365 days.

3. Click **Next**.

The Review and Add Key screen is displayed.

Review And Add Key

This screen shows the key details that you have provided. These details are divided into KEY MATERIAL ORIGIN, KEY POLICY, NATIVE KEY, and KEY SCHEDULES (or AWS AUTOMATIC KEY ROTATION) sections. For a multi-region key, the NATIVE KEY section shows Multi-Region Key as the Regionality.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the MATERIAL ORIGIN, KEY POLICY, NATIVE KEY, and KEY SCHEDULES (or AWS AUTOMATIC KEY ROTATION) sections, and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the NATIVE KEY section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **OK**. The Add AWS Key wizard is closed.

The newly created key is displayed in the list of AWS keys.

Adding Key Material Using External (BYOK) Source

To add key material using external BYOK as a key source:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**.
3. Click **Add Key**. The Key Material Origin screen of the Add AWS Key wizard is displayed.

Key Material Origin

1. Select the desired **AWS Account** from the drop-down list.
2. Select the desired **Region** from the drop-down list. The list shows the regions in which the selected AWS account is available.
3. Select **External (BYOK)** as **Origin Type**.
4. Select the **Source**. The options are:
 - **CipherTrust (External)**: Add key by creating or uploading an external CipherTrust key as the source key. Refer to [Adding Key Using External CipherTrust as External \(BYOK\) Source](#) for details.
 - **CipherTrust (Local)**: Add key by creating or uploading a CipherTrust key as the source key. Refer to [Adding Key Using Local CipherTrust as External \(BYOK\) Source](#) for details.
 - **Vormetric DSM**: Add key by creating or uploading a Vormetric DSM key as the source key. Refer to [Adding Key Using Vormetric DSM as External \(BYOK\) Source](#) for details.
 - **Luna HSM**: Add key by creating or uploading a Luna HSM key as the source key. Refer to [Adding Key Using Luna HSM as External \(BYOK\) Source](#) for details.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

- **Decide Later**: Add key now, but decide the key source later. Refer to [Deciding Key Material Later](#) for details.

Adding Key Using External CipherTrust as External (BYOK) Source

To add a key by creating or uploading an external CipherTrust key as the source key:

Key Material Origin

1. Select **CipherTrust (External)** as the **Source**.
2. Select **Regionality**. The options are:

- **Single-Region Key:** A single-region key cannot be replicated in other AWS regions.
- **Multi-Region Key:** A multi-region key can be replicated in multiple AWS regions. By default, the base key will be referred to as the multi-region primary key. However, any replica of the multi-region key can be set as the primary key later.

Note

After a key is created, its regionality cannot be changed.

3. Click **Next**. The Source Key screen is displayed.

Configure CipherTrust (External) Key

1. Select the **Source Key Material**. You can select Create New Key or Copy Existing Key.

Click the desired tab to view the instructions.

Create New Key

- Click **Create New Key**.
- Select Key Type.** The options are **Symmetric** and **Asymmetric**.
- Select a **Domain**.
- Enter the **Key Name**.
- Select **Algorithm**:
 - For **Symmetric** keys, the options are **AES** or **HMAC**.
 - For **Asymmetric** keys, the options are **RSA** or **EC**.
- Select the **Key Size / Key Curve** based on the algorithm:
 - For the **HMAC** algorithm, the options are 256, 384, and 512.
 - For the **RSA** algorithm, the options are 2048, 3072, and 4096.
 - For the **EC** algorithm, the options are secp384r1, secp521r1, and secp256k1.

Note

Not applicable to the **AES** algorithm.

Copy Existing Key

- a. Click **Copy Existing Key**.
- b. Select a **Domain**.
- c. Select **Algorithm**. The options are AES, HMAC, RSA, and EC.
- d. Select the **Key Size / Key Curve** based on the algorithm:
 - For the **HMAC** algorithm, the options are 256, 384, and 512.
 - For the **RSA** algorithm, the options are 2048, 3072, and 4096.
 - For the **EC** algorithm, the options are secp384r1, secp521r1, and secp256k1.

Note

Not applicable to the **AES** algorithm.

- e. (Optional) Enable **Fetch Latest Version Only**.
 - f. Select a **CipherTrust (External) Key** from the list.
2. Click **Next**. The Destination (AWS) Key screen is displayed.

Destination (AWS) Key

1. For RSA Keys, **Select Key Usage**. You can select **Encrypt and Decrypt** or **Sign and Verify**.
2. Enter a user-friendly **Alias** for the key. This helps uniquely identify a key.
3. (Optional) Provide a brief **Description** of the key.
4. (Optional) Set the key expiration date.
 - a. Select the **Set Expiration Date** check box.
 - b. Either enter the expiration time manually or select using the on-screen calendar. The time format is **MM/DD/YYYY HH:MM**, for example, September 16, 2020 1:50 PM.

To select a specific time, click **Time**, and select hours and minutes, from the GUI.

- c. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.

CCKM allows the following characters in tag values:

- Alphanumeric characters
- Special characters `** _ . / = + - @ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

5. Click **Next**. The Key Policy screen is displayed.

On the **Key Policy** screen, you can specify a policy for the key. If you skip the Key Policy screen by clicking **Next**, a default policy will be applied to the key.

Key Policy

Specify a policy for the key. You can create a new policy or select an existing policy.

Click the desired tab to view the instructions.

Create New Policy

1. Select the **Create New Policy** option.
2. Select a policy view. You can select either the Basic view or Raw view. The default view is Basic.

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

In the Basic view, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.
- d. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

3. Select **Save this Key Policy** and **Enter Template Name**.
4. Click **Next**.

Select Saved Policy

1. Select the **Select Saved Policy** option.
2. Select a policy from the **Saved Policies** drop-down list.

You can also update the selected policy, and create a new policy using it.

- a. Select the policy view, you can select **Raw View** or **Basic View**. By default, the policy will open in the Raw View.
- b. Update the policy.

Raw View

Make the necessary changes in the policy.

Basic View

In the Basic View, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.

Note

If you make changes in one of the views and switch to the other view without updating the policy, you will see a **Warning** message. Click **Yes** to continue.

- c. Click **Update**. The **Update Policy** message is displayed.

To save the changes as a new policy, select the **DO NOT Push Changes** check box, and enter the **New Policy** name.

Note

If the selected policy is Unverified, the Update Policy message will not be displayed, a new policy will not be created.

- d. Click **Apply**.
- e. Click **Next**.

Build From Template

1. Select the **Build From Template** option.
2. **Select Template**.
3. (Optional) Make the necessary changes in the template.
4. Click **Next**.

The Add to Schedule screen is displayed.

Add to Schedule

1. From the **Rotation** drop-down list, select a schedule to apply.

Note

If you're creating an imported AES symmetric key and the selected schedule has "Rotate Material (imported AES keys)" enabled, the key material will be rotated automatically when the schedule runs. For all other cases (including schedules without this option enabled), a new key will be created on rotation, and the alias will be reassigned to the new key.

2. Specify whether to **Disable Encrypt Permissions on Current Key**. By default, the encrypt permissions are disabled on the current key. Clear the check box to enable permissions.
3. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.
 - **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.

- **DSM:** Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.

4. Click **Next**.

The Review and Add Key screen is displayed.

Review And Add Key

This screen shows the key details that you have provided. These details are divided into KEY MATERIAL ORIGIN, KEY POLICY, SOURCE KEY, DESTINATION KEY and KEY SCHEDULES sections. For a multi-region key, the DESTINATION KEY section shows Multi-Region Key as the Regionality.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the KEY MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, KEY POLICY and KEY SCHEDULES sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the KEY POLICY, SOURCE KEY, DESTINATION KEY and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **OK**. The Add AWS Key wizard is closed.

The newly created key is displayed in the list of AWS keys.

Adding Key Using Local CipherTrust as External (BYOK) Source

To add a key by creating or uploading a CipherTrust key as the source key:

Key Material Origin

1. Select **CipherTrust (Local)** as the **Source**.

2. Select **Regionality**. The options are:

- **Single-Region Key:** A single-region key cannot be replicated in other AWS regions.
- **Multi-Region Key:** A multi-region key can be replicated in multiple AWS regions. By default, the base key will be referred to as the multi-region primary key. However, any replica of the multi-region key can be set as the primary key later.

Note

After a key is created, its regionality cannot be changed.

3. Click **Next**. The Configure CipherTrust (Local) Key screen is displayed.

Configure CipherTrust (Local) Key

1. Select the **Source Key Material**. You can select Create New Key or Copy Existing Key.

Click the desired tab to view the instructions.

Create New Key

- a. Click **Create New Key**.
- b. **Select Key Type**. The options are **Symmetric** and **Asymmetric**.
- c. Enter the **Key Name**.
- d. Select **Algorithm**:
 - For **Symmetric** keys, the options are **AES** or **HMAC**.
 - For **Asymmetric** keys, the options are **RSA** or **EC**.
- e. Select the **Key Size / Key Curve** based on the algorithm:
 - For the **HMAC** algorithm, the options are 256, 384, and 512.
 - For the **RSA** algorithm, the options are 2048, 3072, and 4096.
 - For the **EC** algorithm, the options are secp384r1, secp521r1, and secp256k1.

Note

Not applicable to the **AES** algorithm.

Copy Existing Key

- a. Click **Copy Existing Key**.
- b. Select **Algorithm**. The options are AES, HMAC, RSA, and EC.
- c. Select the **Key Size / Key Curve** based on the algorithm:

- For the **HMAC** algorithm, the options are 256, 384, and 512.
- For the **RSA** algorithm, the options are 2048, 3072, and 4096.
- For the **EC** algorithm, the options are secp384r1, secp521r1, and secp256k1.

Note

Not applicable to the **AES** algorithm.

- d. (Optional) Enable **Fetch Latest Version Only**.
 - e. Select a **CipherTrust (Local) Key** from the list.
2. Click **Next**. The Destination (AWS) Key screen is displayed.

Destination (AWS) Key

1. For RSA Keys, **Select Key Usage**. You can select **Encrypt and Decrypt** or **Sign and Verify**.
2. Enter a user-friendly **Alias** for the key. This helps uniquely identify a key.
3. (Optional) Provide a brief **Description** of the key.
4. (Optional) Set the key expiration date.
 - a. Select the **Set Expiration Date** check box.
 - b. Either enter the expiration time manually or select using the on-screen calendar. The time format is **MM/DD/YYYY HH:MM**, for example, September 16, 2020 1:50 PM.

To select a specific time, click **Time**, and select hours and minutes, from the GUI.
- c. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:
 - a. Specify a tag name.
 - b. Specify the tag value.

CCKM allows the following characters in tag values:

- Alphanumeric characters
- Special characters **** _ . / = + - @ ****

- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

5. Click **Next**. The Key Policy screen is displayed.

On the **Key Policy** screen, you can specify a policy for the key. If you skip the Key Policy screen by clicking **Next**, a default policy will be applied to the key.

Key Policy

Specify a policy for the key. You can create a new policy or select an existing policy.

Click the desired tab to view the instructions.

Create New Policy

1. Select the **Create New Policy** option.
2. Select a policy view. You can select either the Basic view or Raw view. The default view is Basic.

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

In the Basic view, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.
- d. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

3. Select **Save this Key Policy** and **Enter Template Name**.
4. Click **Next**.

Select Saved Policy

1. Select the **Select Saved Policy** option.
-

-
2. Select a policy from the **Saved Policies** drop-down list.

You can also update the selected policy, and create a new policy using it.

- a. Select the policy view, you can select **Raw View** or **Basic View**. By default, the policy will open in the Raw View.
- b. Update the policy.

Raw View

Make the necessary changes in the policy.

Basic View

In the Basic View, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.

Note

If you make changes in one of the views and switch to the other view without updating the policy, you will see a **Warning** message. Click **Yes** to continue.

- c. Click **Update**. The **Update Policy** message is displayed.

To save the changes as a new policy, select the **DO NOT Push Changes** check box, and enter the **New Policy** name.

Note

If the selected policy is Unverified, the Update Policy message will not be displayed, a new policy will not be created.

- d. Click **Apply**.
- e. Click **Next**.

Build From Template

1. Select the **Build From Template** option.
2. **Select Template.**
3. (Optional) Make the necessary changes in the template.
4. Click **Next**.

The Add to Schedule screen is displayed.

Add to Schedule

1. From the **Rotation** drop-down list, select a schedule to apply.

Note

If you're creating an imported AES symmetric key and the selected schedule has "Rotate Material (imported AES keys)" enabled, the key material will be rotated automatically when the schedule runs. For all other cases (including schedules without this option enabled), a new key will be created on rotation, and the alias will be reassigned to the new key.

2. Specify whether to **Disable Encrypt Permissions on Current Key**. By default, the encrypt permissions are disabled on the current key. Clear the check box to enable permissions.
3. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust:** CipherTrust Manager.
 - **Luna:** HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM:** Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review And Add Key

This screen shows the key details that you have provided. These details are divided into KEY MATERIAL ORIGIN, KEY POLICY, SOURCE KEY, DESTINATION KEY and KEY SCHEDULES sections. For a multi-region key, the DESTINATION KEY section shows Multi-Region Key as the Regionality.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the KEY MATERIAL ORIGIN,

SOURCE KEY, DESTINATION KEY, KEY POLICY and KEY SCHEDULES sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the KEY POLICY, SOURCE KEY, DESTINATION KEY and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **OK**. The Add AWS Key wizard is closed.

The newly created key is displayed in the list of AWS keys.

Adding Key Using Vormetric DSM as External (BYOK) Source

To add a key by creating or uploading a Vormetric DSM key as the source key:

Key Material Origin

1. Select **Vormetric DSM** as the **Source**.
2. Select **Regionality**. The options are:
 - **Single-Region Key**: A single-region key cannot be replicated in other AWS regions.
 - **Multi-Region Key**: A multi-region key can be replicated in multiple AWS regions. By default, the base key will be referred to as the multi-region primary key. However, any replica of the multi-region key can be set as the primary key later.

Note

After a key is created, its regionality cannot be changed.

3. Click **Next**. The Source Key screen is displayed.

Source Key

1. Select the **Source Key Material**. You can select Create New Key or Copy Existing Key.

Click the desired tab to view the instructions.

Create New Key

- a. Click **Create New Key**.
 - b. Enter the **DSM Key Name**.
-

-
- c. Select a **DSM Domain**.

Note

Not applicable to the **AES** algorithm.

Copy Existing Key

- a. Click **Copy Existing Key**.

Note

The by default selected Algorithm and Key Size options are AES and 256, respectively.

- b. Select a **DSM Key** from the list.

- 2. Click **Next**. The Destination (AWS) Key screen is displayed.

Destination (AWS) Key

1. For RSA Keys, **Select Key Usage**. You can select **Encrypt and Decrypt** or **Sign and Verify**.
2. Enter a user-friendly **Alias** for the key. This helps uniquely identify a key.
3. (Optional) Provide a brief **Description** of the key.
4. (Optional) Set the key expiration date.
 - a. Select the **Set Expiration Date** check box.
 - b. Either enter the expiration time manually or select using the on-screen calendar. The time format is **MM/DD/YYYY HH:MM**, for example, September 16, 2020 1:50 PM.

To select a specific time, click **Time**, and select hours and minutes, from the GUI.
- c. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:
 - a. Specify a tag name.
 - b. Specify the tag value.

CCKM allows the following characters in tag values:

- Alphanumeric characters
- Special characters `** _ . / = + - @ **`

c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

5. Click **Next**. The Key Policy screen is displayed.

On the **Key Policy** screen, you can specify a policy for the key. If you skip the Key Policy screen by clicking **Next**, a default policy will be applied to the key.

Key Policy

Specify a policy for the key. You can create a new policy or select an existing policy.

Click the desired tab to view the instructions.

Create New Policy

1. Select the **Create New Policy** option.
2. Select a policy view. You can select either the Basic view or Raw view. The default view is Basic.

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

In the Basic view, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.
- d. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

3. Select **Save this Key Policy** and **Enter Template Name**.

-
- Click **Next**.

Select Saved Policy

- Select the **Select Saved Policy** option.
- Select a policy from the **Saved Policies** drop-down list.

You can also update the selected policy, and create a new policy using it.

- Select the policy view, you can select **Raw View** or **Basic View**. By default, the policy will open in the Raw View.
- Update the policy.

Raw View

Make the necessary changes in the policy.

Basic View

In the Basic View, Key Admins is the default tab.

- On the **Key Admins** tab, select the desired key admins.
- Click the **Key Users** tab.
- Select the desired key users.

Note

If you make changes in one of the views and switch to the other view without updating the policy, you will see a **Warning** message. Click **Yes** to continue.

- Click **Update**. The **Update Policy** message is displayed.

To save the changes as a new policy, select the **DO NOT Push Changes** check box, and enter the **New Policy** name.

Note

If the selected policy is Unverified, the Update Policy message will not be displayed, a new policy will not be created.

d. Click **Apply**.

e. Click **Next**.

Build From Template

1. Select the **Build From Template** option.
2. **Select Template**.
3. (Optional) Make the necessary changes in the template.
4. Click **Next**.

The Add to Schedule screen is displayed.

Add to Schedule

1. From the **Rotation** drop-down list, select a schedule to apply.

Note

If you're creating an imported AES symmetric key and the selected schedule has "Rotate Material (imported AES keys)" enabled, the key material will be rotated automatically when the schedule runs. For all other cases (including schedules without this option enabled), a new key will be created on rotation, and the alias will be reassigned to the new key.

2. Specify whether to **Disable Encrypt Permissions on Current Key**. By default, the encrypt permissions are disabled on the current key. Clear the check box to enable permissions.
3. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.
 - **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review And Add Key

This screen shows the key details that you have provided. These details are divided into KEY MATERIAL ORIGIN, KEY POLICY, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections. For a multi-region key, the DESTINATION KEY section shows Multi-Region Key as the Regionality.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the KEY MATERIAL ORIGIN, KEY POLICY, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the KEY POLICY, SOURCE KEY, DESTINATION KEY and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **OK**. The Add AWS Key wizard is closed.

The newly created key is displayed in the list of AWS keys.

Adding Key Using Luna HSM as External (BYOK) Source

To add a key by creating or uploading a Luna HSM key as the source key:

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Key Material Origin

1. Select **Luna HSM** as the **Source**.
2. Select **Regionality**. The options are:
 - **Single-Region Key**: A single-region key cannot be replicated in other AWS regions.
 - **Multi-Region Key**: A multi-region key can be replicated in multiple AWS regions. By default, the base key will be referred to as the multi-region primary key. However, any replica of the multi-region key can be set as the primary key later.

Note

After a key is created, its regionality cannot be changed.

3. Click **Next**. The Source Key screen is displayed.

Source Key

1. Select the **Source Key Material**. This specifies how to create the key. The options are:

Create New Key

- a. Click **Create New Key**.
- b. **Select Key Type**. The options are **Symmetric** and **Asymmetric**.
- c. Enter the **Key Name**.
- d. Select a **Partition ID**.
- e. Select **Algorithm**:
 - For **Symmetric** keys, the option is **AES**.
 - For **Asymmetric** keys, the options are **RSA** or **EC**.
- f. (Applicable to RSA algorithm only) Select a Mechanism. The options are CKM_RSA_FIPS_186_3_AUX_PRIME_KEY_PAIR_GEN, CKM_RSA_X9_31_KEY_PAIR_GEN, CKM_RSA_FIPS_186_3_PRIME_KEY_PAIR_GEN, and CKM_RSA_PKCS_KEY_PAIR_GEN.

Note

The mechanisms for the AES algorithm and EC algorithm are CKM_AES_KEY_GEN and CKM_EC_KEY_PAIR_GEN, respectively.

- g. Select the **Key Size / Key Curve** based on the algorithm:
 - For the **RSA** algorithm, the options are 2048, 3072, and 4096.
 - For the **EC** algorithm, the options are secp384r1, secp521r1, and secp256k1.

Note

Not applicable to the **AES** algorithm.

Copy Existing Key

- a. Click **Copy Existing Key**.
- b. Select **Algorithm**. The options are AES, RSA, and EC.

-
- c. Select the **Key Size / Key Curve** based on the algorithm:
 - For the **AES** algorithm, the option is 256.
 - For the **RSA** algorithm, the options are 2048, 3072, and 4096.
 - For the **EC** algorithm, the options are secp384r1, secp521r1, and secp256k1.
 - d. Select a **HSM Key** from the list.

2. Click **Next**. The Destination (AWS) Key screen is displayed.

Destination (AWS) Key

1. For RSA keys, **Select Key Usage**. You can select **Encrypt and Decrypt** or **Sign and Verify**.
2. Enter a user-friendly **Alias** for the key. This helps uniquely identify a key.
3. (Optional) Provide a brief **Description** of the key.
4. (Optional) Set the key expiration date.
 - a. Select the **Set Expiration Date** check box.
 - b. Either enter the expiration time manually or select using the on-screen calendar. The time format is **MM/DD/YYYY HH:MM**, for example, September 16, 2020 1:50 PM.

To select a specific time, click **Time**, and select hours and minutes, from the GUI.
- c. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:
 - a. Specify a tag name.
 - b. Specify the tag value.

CCKM allows the following characters in tag values:

- Alphanumeric characters
- Special characters **** _ . / = + - @ ****

- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

5. Click **Next**. The Key Policy screen is displayed.

On the **Key Policy** screen, you can specify a policy for the key. If you skip the Key Policy screen by clicking **Next**, a default policy will be applied to the key.

Key Policy

Specify a policy for the key. You can create a new policy or select an existing policy.

Click the desired tab to view the instructions.

Create New Policy

1. Select the **Create New Policy** option.
2. Select a policy view. You can select either the Basic view or Raw view. The default view is Basic.

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

In the Basic view, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.
- d. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

3. Select **Save this Key Policy** and **Enter Template Name**.
4. Click **Next**.

Select Saved Policy

1. Select the **Select Saved Policy** option.
2. Select a policy from the **Saved Policies** drop-down list.

You can also update the selected policy, and create a new policy using it.

- a. Select the policy view, you can select **Raw View** or **Basic View**. By default, the policy will open in the Raw

View.

- b. Update the policy.

Raw View

Make the necessary changes in the policy.

Basic View

In the Basic View, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.

Note

If you make changes in one of the views and switch to the other view without updating the policy, you will see a **Warning** message. Click **Yes** to continue.

- c. Click **Update**. The **Update Policy** message is displayed.

To save the changes as a new policy, select the **DO NOT Push Changes** check box, and enter the **New Policy** name.

Note

If the selected policy is Unverified, the Update Policy message will not be displayed, a new policy will not be created.

- d. Click **Apply**.
- e. Click **Next**.

Build From Template

1. Select the **Build From Template** option.
2. **Select Template.**
3. (Optional) Make the necessary changes in the template.

4. Click **Next**.

The Add to Schedule screen is displayed.

Add to Schedule

1. From the **Rotation** drop-down list, select a schedule to apply.

Note

If you're creating an imported AES symmetric key and the selected schedule has "Rotate Material (imported AES keys)" enabled, the key material will be rotated automatically when the schedule runs. For all other cases (including schedules without this option enabled), a new key will be created on rotation, and the alias will be reassigned to the new key.

2. Specify whether to **Disable Encrypt Permissions on Current Key**. By default, the encrypt permissions are disabled on the current key. Clear the check box to enable permissions.
3. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.
 - **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review And Add Key

This screen shows the key details that you have provided. These details are divided into KEY MATERIAL ORIGIN, KEY POLICY, SOURCE KEY, DESTINATION KEY and KEY SCHEDULES sections. For a multi-region key, the DESTINATION KEY section shows Multi-Region Key as the Regionality.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the KEY MATERIAL ORIGIN, KEY POLICY, SOURCE KEY, DESTINATION KEY and KEY SCHEDULES sections and update details.

Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the KEY POLICY, SOURCE KEY, DESTINATION KEY and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **OK**. The Add AWS Key wizard is closed.

The newly created key is displayed in the list of AWS keys.

Deciding Key Material Later

Add key now, but decide the key source later.

To add a new key without deciding the key material source:

Key Material Origin

1. Select **Decide Later** as the **Source**.
2. Select **Regionality**. The options are:
 - **Single-Region Key**: A single-region key cannot be replicated in other AWS regions.
 - **Multi-Region Key**: A multi-region key can be replicated in multiple AWS regions. By default, the base key will be referred to as the multi-region primary key. However, any replica of the multi-region key can be set as the primary key later.

Note

After a key is created, its regionality cannot be changed.

3. Click **Next**. The Destination (AWS) Key screen is displayed.

Destination (AWS) Key

1. **Select Key Type**. The options are **Symmetric** and **Asymmetric**.
2. Under **Select Key Usage**, select the key usage:
 - **Encrypt and Decrypt**:
 - For symmetric keys, the key is used only to encrypt and decrypt the data.
 - For asymmetric keys, the public key is used to encrypt while the private key is used to decrypt.
 - (Symmetric keys only) **Generate and Verify MAC**: The key is used only to generate and verify hash-based message authentication codes (HMACs).
 - (Asymmetric keys only) **Sign and Verify**: In this key pair, the public key is used to sign while the private key is used to verify.

3. Enter a user-friendly **Alias** for the key. This helps uniquely identify a key.
4. (Optional) Provide a brief **Description** of the key.
5. Select the desired **Key Algorithm** from the drop-down list.

Note

The **Algorithm** field is not displayed for symmetric keys when their Key Usage is selected as **Encrypt and Decrypt**.

6. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.

CCKM allows the following characters in tag values:

- Alphanumeric characters
- Special characters **** _ . / = + - @ ****

- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

7. Click **Next**. The Key Policy screen is displayed.

On the **Key Policy** screen, you can specify a policy for the key. If you skip the Key Policy screen by clicking **Next**, a default policy will be applied to the key.

Key Policy

Specify a policy for the key. You can create a new policy or select an existing policy.

Click the desired tab to view the instructions.

Create New Policy

1. Select the **Create New Policy** option.
2. Select a policy view. You can select either the Basic view or Raw view. The default view is Basic.

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

In the Basic view, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.
- d. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

3. Select **Save this Key Policy** and **Enter Template Name**.
4. Click **Next**.

Select Saved Policy

1. Select the **Select Saved Policy** option.
2. Select a policy from the **Saved Policies** drop-down list.

You can also update the selected policy, and create a new policy using it.

- a. Select the policy view, you can select **Raw View** or **Basic View**. By default, the policy will open in the Raw View.
- b. Update the policy.

Raw View

Make the necessary changes in the policy.

Basic View

In the Basic View, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
 - b. Click the **Key Users** tab.
-

-
- c. Select the desired key users.

Note

If you make changes in one of the views and switch to the other view without updating the policy, you will see a **Warning** message. Click **Yes** to continue.

- c. Click **Update**. The **Update Policy** message is displayed.

To save the changes as a new policy, select the **DO NOT Push Changes** check box, and enter the **New Policy** name.

Note

If the selected policy is Unverified, the Update Policy message will not be displayed, a new policy will not be created.

- d. Click **Apply**.
- e. Click **Next**.

Build From Template

1. Select the **Build From Template** option.
2. **Select Template**.
3. (Optional) Make the necessary changes in the template.
4. Click **Next**.

The Add to Schedule screen is displayed.

Add to Schedule

1. From the **Rotation** drop-down list, select a schedule to apply.

Note

If you're creating an imported AES symmetric key and the selected schedule has "Rotate Material (imported AES keys)" enabled, the key material will be rotated automatically when the schedule runs.

For all other cases (including schedules without this option enabled), a new key will be created on rotation, and the alias will be reassigned to the new key.

2. Specify whether to **Disable Encrypt Permissions on Current Key**. By default, the encrypt permissions are disabled on the current key. Clear the check box to enable permissions.
3. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.
 - **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review And Add Key

This screen shows the key details that you have provided. These details are divided into KEY MATERIAL ORIGIN, KEY POLICY, DESTINATION KEY and KEY SCHEDULES sections. For a multi-region key, the DESTINATION KEY section shows Multi-Region Key as the Regionality. The KEY MATERIAL ORIGIN section shows the Source Type as External (BYOK) and Source as Decide Later.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the KEY MATERIAL ORIGIN, KEY POLICY, DESTINATION KEY and KEY SCHEDULES sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **OK**. The Add AWS Key wizard is closed.

The newly created key is displayed in the list of AWS keys. The origin of the key is **BYOK - External** and the key state is **PendingImport**.

Creating CloudHSM Keys

Before you can create a CloudHSM key, you must ensure your CloudHSM key store is configured. For more information, see [AWS CloudHSM Key Store Resources](#).

To create a CloudHSM key using CloudHSM Source

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**.
3. Click **Add Key**. The Key Material Origin screen of the Add AWS Key wizard is displayed.

Key Material Origin

1. Select the desired **AWS Account** from the drop-down list.
2. Select the desired **Region** from the drop-down list.
3. Select **AWS CloudHSM Key Store** from **Origin Type**.
4. Select the desired **CloudHSM Key Store** from the drop-down list.
5. Enter a user-friendly **Alias** for the key. This helps uniquely identify the key.
6. (Optional) Provide a brief **Description** of the key.
7. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.

CCKM allows the following characters in tag values:

- Alphanumeric characters
- Special characters **** _ . / = + - @ ****

- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

8. Click **Next**. The Key Policy screen is displayed.

On the **Key Policy** screen, you can specify a policy for the key. If you skip the Key Policy screen by clicking **Next**, a default policy will be applied to the key.

Key Policy

Specify a policy for the key. You can create a new policy or select an existing policy.

Click the desired tab to view the instructions.

Create New Policy

1. Select the **Create New Policy** option.
2. Select a policy view. You can select either the Basic view or Raw view. The default view is Basic.

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

In the Basic view, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.
- d. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

3. Select **Save this Key Policy** and **Enter Template Name**.
4. Click **Next**.

The Review And Add Key screen is displayed.

Select Existing Policy

1. Select the **Select Existing Policy** option.
2. Select a policy from the **Saved Policies** drop-down list.

To view the raw details of the selected policy, click **View Policy (Raw)**. To hide the raw details, click **Hide Policy (Raw)**.

3. Click **Next**.

The Review And Add Key screen is displayed.

Review And Add Key

1. Review the key details, and click **Add Key** to create.
2. Once creation completes, click **OK** to close the dialog.

Creating HYOK Keys

You must [prepare an external custom key store on CCKM](#) before you can create an HYOK key.

As part of creating an HYOK key, you create a new source key in either a Luna HSM partition or a CipherTrust Manager that is associated with the external custom key store. The source of the HYOK key must match the source of the [external custom key store](#).

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Configuration varies based on whether the HYOK is created in the [linked](#) or [unlinked](#) state.

If the HYOK key is created in the linked state, a KMS key is automatically created in the corresponding AWS KMS external key store.

If the HYOK key is created in the unlinked state, you must either [link it](#) to automatically create the corresponding KMS key on AWS KMS, or you must manually create the KMS key on AWS KMS. If you choose to manually create a KMS key, you can [refresh keys](#) to later detect and link the KMS key to its HYOK key on CCKM. Managing [AWS policies associated with the HYOK key on CCKM](#) is unsupported with unlinked keys.

Create a Linked HYOK Key

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**.
3. Click **Add Key**. The Key Material Origin screen of the Add AWS Key wizard is displayed.

Key Material Origin

1. Select the desired **AWS Account** from the drop-down list.
 2. Select **External Custom Key Store (HYOK)** from **Origin Type**.
 3. Select **CipherTrust (Local)** or **Luna HSM** from **Source**.
 4. Select **Linked Key** from the Linked State.
 5. Click **Next**. The Source Key screen is displayed.
-

Source Key

Caution

Modify AWS XKS (HYOK) only through the CipherTrust Cloud Key Management (CCKM) GUI, API, or CLI. Do not modify the source using the **Keys** menu, the `ksctl keys` command, or the `/v1/vault/keys2` endpoint. Such modifications might make the XKS (HYOK) key unavailable to the AWS cloud service.

1. Select an **External Key Store** from the drop down.

Note

Only key stores hosted on the local CCKM, and matching the selected **Source** are available.

2. Select the **Source Key Material**. This specifies how to create the key.

- **Create New Key**

- a. Click to create a fresh key.
- b. Specify the **Source Key Name** for the source key. This will be the name for the new key.

Note

For Luna HSM keys, key attributes are displayed. These values are not editable.

- **Copy Existing Key**

- a. Click to create a new key by copying an existing source key.
- b. Select a **CipherTrust (Local) Key** from the list. For Luna HSM, the list shows the available keys based on the Luna HSM connection.

Valid source keys must have specific attributes, depending on the source type.

Source Type	Required Attributes
CipherTrust Manager	• Not exportable • Not deletable • Usage Masks: Encrypt, Decrypt, Wrap, Unwrap

Source Type	Required Attributes
Luna HSM	• CKA_EXTRACTABLE = FALSE • CKA_SENSITIVE = TRUE • CKA_ENCRYPT = TRUE • CKA_DECRYPT = TRUE • CKA_WRAP = TRUE • CKA_UNWRAP = TRUE

3. Click **Next**. The Destination (AWS key) screen is displayed.

Destination (AWS) Key

1. Add an **Alias** for the new KMS key.
2. Optionally add a **Description** and **Tags**, if desired.
3. Click **Next** The Key Policy screen is displayed.

On the **Key Policy** screen, you can specify a policy for the key. If you skip the Key Policy screen by clicking **Next**, a default policy will be applied to the key.

Key Policy

Specify a policy for the key. You can create a new policy or select an existing policy.

Click the desired tab to view the instructions.

Create New Policy

1. Select the **Create New Policy** option.
2. Select a policy view. You can select either the Basic view or Raw view. The default view is Basic.

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

In the Basic view, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.
- d. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON

format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

3. Select **Save this Key Policy** and **Enter Template Name**.
4. Click **Next**.

Select Saved Policy

1. Select the **Select Saved Policy** option.
2. Select a policy from the **Saved Policies** drop-down list.

You can also update the selected policy, and create a new policy using it.

- a. Select the policy view, you can select **Raw View** or **Basic View**. By default, the policy will open in the Raw View.
- b. Update the policy.

Raw View

Make the necessary changes in the policy.

Basic View

In the Basic View, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.

Note

If you make changes in one of the views and switch to the other view without updating the policy, you will see a **Warning** message. Click **Yes** to continue.

- c. Click **Update**. The **Update Policy** message is displayed.

To save the changes as a new policy, select the **DO NOT Push Changes** check box, and enter the **New Policy** name.

Note

If the selected policy is Unverified, the Update Policy message will not be displayed, a new policy will not be created.

- d. Click **Apply**.
- e. Click **Next**.

Build From Template

1. Select the **Build From Template** option.
2. **Select Template**.
3. (Optional) Make the necessary changes in the template.
4. Click **Next**.

The Add to Schedule screen is displayed.

Add to Schedule

1. Select the desired **Rotation** schedule to apply to the key from the drop-down list. This is an optional step.

Note

If you're creating an imported AES symmetric key and the selected schedule has "Rotate Material (imported AES keys)" enabled, the key material will be rotated automatically when the schedule runs. For all other cases (including schedules without this option enabled), a new key will be created on rotation, and the alias will be reassigned to the new key.

2. Click **Next**. The Review And Add Key screen is displayed.

Review And Add Key

1. Review the key details, and click **Add Key** to create.

Creation completes and an **XKS ID** is generated for the new HYOK key.

2. Click **OK** to close the dialog.

Create an Unlinked HYOK key

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**.
3. Click **Add Key**. The Key Material Origin screen of the Add AWS Key wizard is displayed.

Key Material Origin

1. Select the desired **AWS Account** from the drop-down list.
2. Select **External Custom Key Store (HYOK)** from **Origin Type**.
3. Select **CipherTrust** or **Luna HSM** from **Source**.
4. Select **Unlinked Key** from the Linked State.

This means that the corresponding key is not automatically created in AWS KMS, and you must either link the HYOK after creation, or manually create the KMS key in AWS KMS.

5. Click **Next**. The Source Key screen is displayed.

Source Key

Caution

Modify AWS XKS (HYOK) only through the CipherTrust Cloud Key Management (CCKM) GUI, API, and CLI. Do not modify the source using the **Keys** menu, the `ksctl keys` command, or the `/v1/vault/keys2` endpoint. Such modifications might make the XKS (HYOK) key unavailable to the AWS cloud service.

1. Select an **External Key Store** from the drop down.

Note

Only local key stores matching the selected **Source** are available.

2. Select the **Source Key Material**. This specifies how to create the key. The source key must be a symmetric AES-256 key.

For Luna source key stores, the `CKA_SENSITIVE`, `CKA_ENCRYPT`, `CKA_DECRYPT`, `CKA_WRAP`, and `CKA_UNWRAP` attributes must be enabled, and `CKA_EXTRACTABLE` must be disabled.

The options are:

- **Create New Key**

- a. Click to create a fresh key.
- b. Specify the **Source Key Name** for the source key. This will be the name for the new key.

Note

For Luna HSM keys, key attributes are displayed. These values are not editable.

- **Copy Existing Key**

- a. Click to create a new key by copying an existing source key.
- b. Select a **CipherTrust (Local) Key** from the list. For Luna HSM, the list shows the available keys based on the Luna HSM connection.

3. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. Select the desired **Rotation** schedule to apply to the key from the drop-down list. This is an optional step.

Note

If you're creating an imported AES symmetric key and the selected schedule has "Rotate Material (imported AES keys)" enabled, the key material will be rotated automatically when the schedule runs. For all other cases (including schedules without this option enabled), a new key will be created on rotation, and the alias will be reassigned to the new key.

2. Click **Next**. The Review And Add Key screen is displayed.

Review And Add Key

1. Review the key details, and click **Add Key** to create.

Creation completes and an **XKS ID** is generated for the new HYOK key.

2. Click **OK** to close the dialog.
3. Create the corresponding KMS key. You can either:
 - [Link the new HYOK key in CCKM](#).

- Manually create the key in AWS KMS:
 - a. In CCKM, retrieve the **XKS ID** value from the [key details page](#).
 - b. On AWS KMS, navigate to the external key store which corresponds to the CCKM external custom key store you selected for the HYOK key.
 - c. Create the KMS key. Provide the **XKS ID** for any fields indicating an **External key ID**.
 - d. Follow AWS documentation to associate the desired AWS services with the KMS key.
 - e. If you want to later link the KMS key to its associated HYOK key, [refresh all AWS keys](#).

Encryption and decryption requests will now be executed with the source key.

If the source key is a Luna HSM key, the cryptographic requests are executed inside the Luna HSM.

If the source key is a CipherTrust Manager key, the cryptographic requests are executed inside the CipherTrust Manager.

Linking HYOK Keys

If you created an HYOK key in an unlinked state, you can link it after creation. Linking an HYOK key automatically creates a new corresponding KMS key in AWS KMS.

Note

To link an existing KMS key, [refresh all the keys](#).

To link an HYOK key

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.

You can filter this list on **Origin: HYOK-CCKM** to display only local HYOK keys.

3. Click the overflow icon () corresponding to the desired key and click **Link**.

The Configure AWS Key screen of the Link AWS Key wizard is displayed.

4. Optionally provide an **Alias** and **Tags** for the new KMS key.
5. Click **Next**. The AWS Key Policy screen is displayed.
6. Click the desired tab to view the instructions.

Create New Policy

- a. Select the **Create New Policy** option.
- b. Select a policy view. You can select either the Basic view or Raw view. The default view is Basic.

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

In the Basic view, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.
- d. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

- c. Select **Save this Key Policy** and **Enter Template Name**.
- d. Click **Next**.

Select Saved Policy

- a. Select the **Select Saved Policy** option.
- b. Select a policy from the **Saved Policies** drop-down list.

You can also update the selected policy, and create a new policy using it.

- a. Select the policy view, you can select **Raw View** or **Basic View**. By default, the policy will open in the Raw View.
- b. Update the policy.

Raw View

Make the necessary changes in the policy.

Basic View

In the Basic View, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.

Note

If you make changes in one of the views and switch to the other view without updating the policy, you will see a **Warning** message. Click **Yes** to continue.

- c. Click **Update**. The **Update Policy** message is displayed.

To save the changes as a new policy, select the **DO NOT Push Changes** check box, and enter the **New Policy** name.

Note

If the selected policy is Unverified, the Update Policy message will not be displayed, a new policy will not be created.

- d. Click **Apply**.
- e. Click **Next**.

Build From Template

- a. Select the **Build From Template** option.
- b. **Select Template**.
- c. (Optional) Make the necessary changes in the template.
- d. Click **Next**.

The Review and Add screen is displayed.

7. Review the **AWS Key** and **AWS Key Policy** details.
-

8. Click **Link** to accept, link the HYOK key, and create a new KMS key.
9. Click **OK**.

Viewing AWS Keys

To view an AWS key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed. The AWS Keys page displays following details:

Field	Description
Alias	Unique, user-friendly alias of the key. This is useful in searching for specific keys. The AWS Keys page shows the latest alias under the Alias field. Additional aliases for the key are displayed in braces. For example, if a key has three aliases and the latest alias is "latest-alias", the Alias field shows "latest-alias (+2 more)". Clicking the link takes to the Aliases section of the key in edit mode.
Key ID	Unique ID of the CipherTrust Manager key.
Account	AWS account name.
Region	AWS region.
Algorithm	Name of the algorithm. Supported algorithms are: • SYMMETRIC_DEFAULT • RSA_2048 • RSA_3072 • RSA_4096 • ECC_NIST_P256 • ECC_NIST_P384 • ECC_NIST_P521 • ECC_SECG_P256K1
Source Key	Name of the source key.
Source Type	Source of the key. • CipherTrust (Local): Local CipherTrust Manager • DSM: Vormetric Data Security Manager • Luna: HSM Luna • CipherTrust (External): External CipherTrust Manager.

Field	Description
Key State	State of the key. The state can be: • Enabled • Disabled • Deleted • PendingDeletion • PendingImport • Unavailable
Creation Date	Time when the key is created.
Origin	The origin of the key can be: • BYOK-CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud. • BYOK - External: Source of the key material is unknown. It is different than CCKM and the native cloud. • HYOK-CCKM: HYOK key exists on CCKM • HYOK-External: An external key from KMS, without the corresponding HYOK key on the local CCKM. This can mean that the HYOK key is present on a different CCKM not clustered with the local one, or that the HYOK key is present on another vendor's external key manager. NOTE: When the CipherTrust Manager is upgraded from 2.0 to 2.1, the Origin column appears blank. The column will be populated on next scheduled key synchronization or on-demand synchronization by clicking the Sync All button.
Expiration Date	Time when the key will expire. Expiration date is nonapplicable for some key types.
Expiration State	State of the expiration.
Regionality	Whether the key is a single-region key, multi-region primary key, or multi-region replica key.
Cloud	Name of the AWS cloud.
Key Usage	How the key is used - for example, to decrypt and encrypt or to sign and verify.
Source Key Container	Container of the source key. A container can be the CipherTrust Manager, DSM, or Luna HSM.
Key Store	(Applicable to HYOK keys) Name of the key store.

Some of these fields are not populated for HYOK keys. The non-populated fields are **Alias**, **Key ID**, **Key State**, and **Creation Date**.

The **Regionality**, **Cloud**, **Key Usage**, and **Source Key Container** columns are hidden by default. To show/hide a column, click the custom view icon (⚙️), select/clear the desired column, and click **OK**. The **Key Usage** column is not populated for HYOK keys.

Sometimes, you might notice certain keys are displayed as grayed out. This happens when the keys are no longer accessible. For example, when:

- Any cloud permissions on the keys are changed. The keys are no longer accessible from the AWS connection.
- Connection is changed in AWS KMS. The new connection does not have permissions to access the keys.
- When AWS regions are changed or removed. The keys from the configured region are no longer accessible.

Creating Replica of Multi-Region Keys

Note

This functionality is applicable to Native and BYOK keys.

A multi-region key can be replicated in multiple AWS regions. By default, the base key will be referred to as the multi-region primary key. Later, you can set any replica of the multi-region key as the primary key.

Note

Only one replica of a primary key can be created in one AWS Region. Moreover, a replica of a replica key cannot be created.

To add a replica:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the alias of the desired multi-region primary key. The detail view of the key is displayed.
4. Scroll down to the **REGIONALITY** section.
5. Click **Add Replicas**. The Add Replica Region screen of the Add Replica Keys dialog box is displayed.

Add Replica Region

1. From the **Select Replica Region** drop-down list, select the AWS region where you want to create the replica key.

-
2. Click **Next**. The Add Labels screen is displayed.

Add Labels

The fields on the Add Labels screen display the current values of the primary key, but you can edit them at any time. AWS KMS does not synchronize any changes to these values.

1. Enter a user-friendly **Alias** for the replica key. This helps uniquely identify the replica key.
2. (Optional) Provide a brief **Description** of the key.
3. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.

CCKM allows the following characters in tag values:

- Alphanumeric characters
- Special characters **** _ . / = + - @ ****

- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

4. Click **Next**. The Key Policy screen is displayed.

On the **Key Policy** screen, you can specify a policy for the key. If you skip the Key Policy screen by clicking **Next**, a default policy will be applied to the key.

Key Policy

Specify a policy for the key. You can create a new policy, select a saved policy, and build from template.

Click the desired tab to view the instructions.

Create New Policy

1. Select the **Create New Policy** option.
2. Select a policy view. You can select either the Basic view or Raw view. The default view is Basic.

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

In the Basic view, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.

- b. Click the **Key Users** tab.
- c. Select the desired key users.
- d. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

3. Select **Save this Key Policy** and **Enter Name**.
4. Click **Next**.

Select Saved Policy

1. Select the **Select Saved Policy** option.
2. Select a policy from the **Saved Policies** drop-down list.

Click **View Policy (Raw)** to view the policy in source JSON.

3. Click **Next**.

Build From Template

1. Select the **Build From Template** option.
2. **Select Template**.
3. (Optional) Make the necessary changes in the template.
4. Click **Next**.

The Review screen is displayed.

Review

This screen shows the replica key details that you have provided. These details are divided into REPLICA REGION, LABELS, AWS KEY POLICY, and CONFIRMATION sections.

Before adding the replica key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **REPLICA REGION** and **LABELS** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Under **CONFIRMATION**, select **I understand that the values I choose here are not synchronized with any other Multi-Region Key**.
3. Click **Add Replica Key**. Your key is successfully added. Close window to return to replica keys list.
4. Click **Close**. The Add Replica Keys wizard is closed.

The newly created replica key is displayed in the list of replica keys under the **REGIONALITY** section. The Regionality of a replica key is displayed as **REPLICA** and its Status moves from **Starting** to **PendingImport**.

Viewing Replicas of a Multi-Region Key

Note

This functionality is applicable to Native and BYOK keys.

To view the replicas of a multi-region key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the alias of the desired multi-region primary key. The detail view of the key is displayed. To view or edit an AWS key:
4. Scroll down to the **REGIONALITY** section. This section shows the replicas of the multi-region primary key. The section shows the following details:

Field	Description
Region	Region where the replica is created.
Key ARN	Amazon Resource Name (ARN) of the AWS replica.
Alias	Alias of the replica key.
State	State of the replica key.
Regionality	Regionality of the replica key is REPLICA .
Creation Date	Date and time when the replica is created.

Viewing or Editing AWS Key Details

To view or edit an AWS key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **View/Edit**.
4. For HYOK keys, you can obtain the following additional values:
 - The **XKS ID**. You can copy this value using the copy button, if you want to manually create a corresponding KMS key. In AWS, this value is called the **External key ID**.
 - The **Source**. This is a link to the CCKM-managed source key stored within either Luna HSM or CipherTrust Manager (depending on which key source you are using). This is important if you want to perform any key functions using either [Luna key functions](#) or [CipherTrust Manager key functions](#).

Note

You can view but not edit details for unlinked HYOK keys.

5. For Native, linked HYOK keys, and BYOK Keys, edit or configure the following fields and click **Update**:
 - **POLICY**: Grant access to external accounts, key administrators, and key users. Refer to [Adding/Editing Policies](#) for details.
 - **ALIASES**: Add or delete aliases of the key.
 - **LABELS**: Add tags to the key. Refer to [Add Labels](#) for characters allowed in AWS tag values.
 - **SCHEDULES**: Applies rotation schedule to the key. Refer to [Apply Key Rotation Schedule](#) for details.

Note

- You can only view and remove the schedule if you have assigned it to the existing AWS native key that has an algorithm as SYMMETRIC_DEFAULT. After you remove the schedule, this section will be hidden.
- If a new AWS native key is created that has an algorithm as SYMMETRIC_DEFAULT, the key will not have the SCHEDULES section.

- **AWS AUTOMATIC KEY ROTATION:** Automatically rotates AWS native key after a specified time period. To automatically rotate the key, enable **automatic key rotation** and enter the **Rotation period (in days)**.

Based on the entered rotation period, the next rotation date will be displayed.

Note

- The default rotation period is 365 days.
- Automatic key rotation only applies to the AWS native keys that have the algorithm as SYMMETRIC_DEFAULT.
- If the next rotation date is not displayed, click the **Refresh** icon.

- **ROTATION HISTORY:** Rotate the key. Refer to [Rotation History](#) for details.

Adding or Deleting Aliases

Note

This functionality is applicable to Native, BYOK, and CloudHSM keys. For HYOK keys, this functionality is applicable to linked keys only.

To **add** a new alias to the key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **View/Edit**.
4. Under **ALIASES**, click **Add Alias**.
5. Enter an **Alias Name**.
6. Click **Save**. The new alias is added to the key.

The AWS Keys page shows the latest alias under the Alias field. Additional aliases for the key are displayed in braces. For example, if a key has three aliases and the latest alias is "latest-alias", the Alias field shows "latest-alias (+2 more)". Clicking the link takes to the Aliases section of the key in edit mode.

To **delete** an alias of the key:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **View/Edit**.
4. Under **Aliases**, click the overflow icon () corresponding to the desired alias.
5. Click **Delete**. The alias is deleted.

Apply Key Rotation Schedule

Note

For HYOK keys, this functionality is applicable to linked keys only.

To apply a key rotation schedule to an AWS key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **View/Edit**.
4. Under **SCHEDULES**, from the **Rotation** drop-down list, select a schedule to apply. If applying a schedule to a CloudHSM key or linked HYOK key, proceed to step 7.

Note

If the selected scheduled rotation has Rotate Key Material (Imported AES Keys) enabled, you will see it alongside the rotation name.

5. (Optional) For BYOK or Native keys, you can grant or deny the encrypt permission to the key policy in the AWS account where the key is created in OR to the key policy in *all accounts* the key is shared in:
 - Select **Disable Encrypt Permissions on Current Key** to deny the encrypt permission to the key policy in the AWS account where the key is created in. Clear the check box to grant the encrypt permission. By default, **Disable Encrypt Permissions on Current Key** is selected, that is, the encrypt permission is denied to the current key.
 - Select **Disable Encrypt Permissions on All Accounts** to deny the encrypt permission to the key policy in all the AWS accounts the key is shared in. Clear the check box to grant the encrypt permission.
6. For BYOK or Native keys, select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust (External)**: External CipherTrust Manager.

- **CipherTrust (Local)**: Local CipherTrust Manager.
- **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
- **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.

Note

If the algorithm of the key is RSA, EC, or HMAC, you can only select **CipherTrust (Local)** as the Key Origin.

7. Click **Update**.

Note

A scheduled key rotation always creates a new key with a randomly generated UUID name.

Rotation History

You can view the details and perform a new rotation for Native and BYOK keys using the SYMMETRIC_DEFAULT algorithm.

- Native keys with the SYMMETRIC_DEFAULT algorithm: You can view the Key Material ID, Date, State, and Type of the key rotation.

To perform a new rotation:

- a. Click **Rotate Key**, the AWS On-Demand Key Rotation dialog box is displayed.
- b. Click **Rotate**.

Warning

As of now, you can only initiate AWS On-Demand Key Rotation 10 times by default.

Note

- On-demand key rotation applies to the AWS native keys that have the algorithm as

SYMMETRIC_DEFAULT.

- If the latest rotation history details are not displayed, click the **Refresh** icon.

- BYOK keys with the SYMMETRIC_DEFAULT algorithm: You can view the Key Material ID, Rotation Date, Key Material State, Import State, Source Type, Origin, Expiration Date, and Key Material Description of the key rotation.

To perform a new rotation, refer to the [Rotating Keys](#) section.

For BYOK keys, you can also promote a key material to current, and re-import, update, or delete it.

Promote to Current: To promote a key material to current:

- a. Click the overflow icon () next to the desired key rotation and select **Promote to Current**. The Promote to Current dialog box is displayed.
- b. Click **Rotate**.
- c. Click **OK**.

Note

You can promote a key material to current only when it's first in the list and its state is Pending Rotation.

Re-Import: To re-import a key material.

- a. Click the overflow icon () next to the desired key rotation and select **Re-Import Key Material**. The Re-Import Key Material dialog box is displayed.
- b. Click **Re-Import Key Material**.
- c. Click **Next**.
- d. (Optional) Select **Key Material Expiration Date**.
- e. (Optional) Enter **Description**.
- f. Click **Next**.
- g. Click **Import**.
- h. Click **OK**.

Note

- You can re-import key material after it has been deleted.
- When all key materials have been imported, the Key State becomes **Enabled**. If any key materials are pending import, the Key State remains **Pending import**.

Update: To update a key material.

- Click the overflow icon () next to the desired key rotation and select **Update Key Material**. The Update Key Material dialog box is displayed.
- (Optional) Select the **Set Expiration Date** check box.

Either enter the key material expiration time manually or select using the on-screen calendar. The time format is MM/DD/YYYY HH:MM, for example, September 16, 2020 1:50 PM.

To select a specific time, click **Time**, and select hours and minutes, from the GUI.

- (Optional) Enter **Description**.
- Click **Save**.

Delete: To delete a key material.

- Click the overflow icon () next to the desired key rotation and select **Delete Key Material**. The Delete Key Material dialog box is displayed.
- Click **Delete Key Material**.

Note

You can delete the imported key material.

Adding/Editing Policies

You can apply a new policy or edit the policy attached to an AWS key on the list view or the details view of the AWS Keys page.

On the list view of the AWS Keys page

Note

This functionality is applicable to all Native and BYOK keys. For HYOK keys, this functionality is applicable to linked keys only.

To add or edit key policy:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Add/Edit Policies**.

The Add/Edit Policies dialog box displays the attached policy, if any. Under **Unsaved Policy**, you can either:

- Update the policy locally. These are local changes only. They do not affect the original policy.

Alternatively, you can save the changes as a new policy. Click **Save to Policy List**, specify a **Policy Name**, and click **Save**. The newly created policy will be available as a saved policy for selection.

- Attach a new saved policy. Click **Select Saved Policy**, select a saved policy from the drop-down list, and click **Apply**.
- Select a template and use it to create a key policy. Click **Build from Template**, select a template from the drop-down list, and click **Apply**.

4. Click **Save**.

On the details view of the AWS Keys page

To add or edit key policy:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the **Alias** link of the key.
4. Navigate to the **POLICY** section.
5. Add or update the policy, as described [above](#).
6. Click **Update**.

Refreshing AWS Keys

Refreshing is the process of downloading keys created on the AWS KMS to CCKM.

Refreshing All Keys

You can refresh keys from all AWS KMS accounts at once.

Refreshing keys also refreshes [AWS CloudHSM key stores](#) and [external custom key stores](#).

Refreshing all keys can set unlinked HYOK keys to a linked state. This happens when CCKM detects a corresponding KMS key for an local unlinked HYOK key. As well, this operation can set unlinked external custom key stores to a linked state. This happens when CCKM detects a corresponding KMS external key store for an local unlinked external custom key store.

To refresh keys:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The AWS Keys page is displayed. This page displays the list of AWS keys.
3. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all keys is a time intensive operation that could take several hours or days to complete. It will continue running in the background. Do you want to continue?

4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > AWS > AWS Keys** page.

Refreshing a Desired Key

You can refresh a desired key from the list. To refresh an AWS key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**.
3. Click the overflow icon () corresponding to the desired alias and click **Refresh**.

A message 'Key refreshed successfully.' is displayed on the screen.

Disabling Keys

Note

This functionality is applicable to all Native, BYOK, and Cloud HSM keys. For HYOK keys, this functionality is applicable to linked keys only.

For HYOK keys, disabling has the same practical effect of disallowing cryptographic operations as **blocking** does. However, disabling changes the external key state on AWS KMS and blocking does not.

Caution

Take care when disabling a key. You cannot use this key in cryptographic operations and it may limit your access to certain resources that use this key. To reverse this action in the future, you can always choose to enable the key again.

To disable the key(s):

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Select the keys to be enabled from the list.
4. Click **Disable**. The Disable Key(s) dialog box is displayed.
5. Click **Disable**. The Job Created dialog box is displayed. A job is created to track the disabling of the keys.
6. Click **OK**.

You can check the job status on the **Bulk Operations** tab.

Enabling Keys

Note

This functionality is applicable to all Native, BYOK, and Cloud HSM keys. For HYOK keys, this functionality is applicable to linked keys only.

For HYOK keys, enabling has the same practical effect of allowing cryptographic operations as **unblocking** does. However, enabling changes the external key state on AWS KMS and unblocking does not.

Caution

Take care when disabling a key. You cannot use this key in cryptographic operations and it may limit your access to certain resources that use this key. To reverse this action in the future, you can always choose to enable the key again.

To enable the key(s):

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Select the keys to be enabled from the list.
4. Click **Enable**. The Enable Key(s) dialog box is displayed.
5. Click **Enable**. The Job Created dialog box is displayed. A job is created to track the enabling of the keys.
6. Click **OK**.

You can check the job status on the **Bulk Operations** tab.

Downloading Keys

Note

This functionality is applicable to asymmetric keys.

Asymmetric keys can be downloaded to your local machines. Symmetric keys cannot be downloaded.

To download an asymmetric key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Download Key**. The key is downloaded.

Importing Key Material

Note

This functionality is applicable to Native and BYOK keys.

You can create a key without key material and can later import the CipherTrust key material to the AWS KMS. As the key material is not created on the AWS KMS, its origin is external.

Note

You can only import AES keys with status **PendingImport** to the AWS KMS.

To import key material:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Import Material**. The Import Material dialog box is displayed.
4. **Select Import Type** (the desired key material source). The options are:
 - Import Using CipherTrust (External)
 - Import Using CipherTrust (Local)
 - Import Using Vormetric DSM
 - Import Using Luna HSM

Note

The Vormetric DSM key option will be available only for the keys that have the algorithm as SYMMETRIC_DEFAULT (AES).

Import Using CipherTrust (External)

When importing the key material from an external CipherTrust Manager, **Select Key Material Origin**. The options are:

-
- **Create New Key**
 - **Use Existing Key**

Create New Key

In this method, the external CipherTrust Manager creates the new key material locally.

1. Select **Create New Key**.
2. Click **Next**.
3. Enter **Key Name**.
4. **Select Domain** for the key. The drop-down list shows the domains of the external CipherTrust Manager linked with the configured connection.
5. (Optional) Select the **Key Material Expiration Date** from the on-screen calendar.
6. Click **Next**. The Review Key Details screen is displayed.
7. Review the key details. If the details are incorrect or you want to modify them, click **Back** and update the details.
8. Click **Import**.

The key import starts. An Import in Progress message is displayed on the screen. Leave the window open until the process is completed. When the status next to the SOURCE KEY section becomes Complete, the key material is imported successfully.

9. Click **OK**.

In this scenario, the CipherTrust creates a new key material and the key material is imported to the AWS KMS. The key state changes from PendingImport to Enabled. The Origin changes to **CCKM**.

Use Existing Key

In this method, the key material of an existing external CipherTrust Manager key is used.

1. Select **Use Existing Key**.
2. Click **Next**.
3. **Select Domain** from the drop-down list. The drop-down list shows the domains of the external CipherTrust Manager linked with the configured connection.
4. Select a **CipherTrust (External) Key** from the list.
5. (Optional) Select the **Key Material Expiration Date** from the on-screen calendar.

-
6. Click **Next**. The Review Key Details screen is displayed.
 7. Review the key details. If the details are incorrect or you want to modify them, click **Back** and update the details.
 8. Click **Import**.

The key import starts. An Import in Progress message is displayed on the screen. Leave the window open until the process is completed.

9. Click **OK**.

In this scenario, the existing key material of the external CipherTrust Manager is imported to the AWS KMS. The key state changes from PendingImport to Enabled. The Origin changes to **CCKM**.

Import Using CipherTrust (Local)

When importing the key material from CipherTrust, **Select Key Material Origin**. The options are:

- [Create New Key](#)
- [Use Existing Key](#)

Create New Key

In this method, CipherTrust Manager creates the new key material locally.

1. Select **Create New Key**.
2. Click **Next**.
3. Enter **Key Name**.
4. (Optional) Select the **Key Material Expiration Date** from the on-screen calendar.
5. Click **Next**. The Review Key Details screen is displayed.
6. Review the key details. If the details are incorrect or you want to modify them, click **Back** and update the details.
7. Click **Import**.

The key import starts. An Import in Progress message is displayed on the screen. Leave the window open until the process is completed. When the status next to the SOURCE KEY section becomes Complete, the key material is imported successfully.

8. Click **OK**.

In this scenario, the CipherTrust creates a new key material and the key material is imported to the AWS KMS. The key state changes from PendingImport to Enabled. The Origin changes to **CCKM**.

Use Existing Key

In this method, the key material of an existing CipherTrust Manager key is used.

1. Select **Use Existing Key**.
2. Click **Next**.
3. Select a **CipherTrust (Local) Key** from the list.
4. (Optional) Select the **Key Material Expiration Date** from the on-screen calendar.
5. Click **Next**. The Review Key Details screen is displayed.
6. Review the key details. If the details are incorrect or you want to modify them, click **Back** and update the details.
7. Click **Import**.

The key import starts. An Import in Progress message is displayed on the screen. Leave the window open until the process is completed.

8. Click **OK**.

In this scenario, the existing CipherTrust key material is imported to the AWS KMS. The key state changes from PendingImport to Enabled. The Origin changes to **CCKM**.

Import Using Vormetric DSM

When importing the key material from Vormetric DSM, **Select Key Material Origin**. The options are:

- [Create New Key](#)
- [Use Existing Key](#)

Create New Key

In this method, DSM creates the new key material locally.

1. Select **Create New Key**.
2. Click **Next**.
3. Enter a **DSM Key Name**.
4. Select the desired **DSM Domain**.
5. (Optional) Select the **Key Material Expiration Date** from the on-screen calendar.
6. Click **Next**. The Review Key Details screen is displayed.

7. Review the key details. If the details are incorrect or you want to modify them, click **Back** and update the details.
8. Click **Import**.

The key import starts. An Import in Progress message is displayed on the screen. Leave the window open until the process is completed. When the status next to the SOURCE KEY section becomes Complete, the key material is imported successfully.

9. Click **OK**.

In this scenario, the DSM creates a new key material and the key material is imported to the AWS KMS. The key state changes from PendingImport to Enabled. The Origin changes to **CCKM**.

Use Existing Key

In this method, the key material of an existing DSM key is used.

1. Select **Use Existing Key**.
2. Click **Next**.
3. Select a **DSM Key** from the list.
4. (Optional) Select the **Key Material Expiration Date** from the on-screen calendar.
5. Click **Next**. The Review Key Details screen is displayed.
6. Review the key details. If the details are incorrect or you want to modify them, click **Back** and update the details.
7. Click **Import**.

The key import starts. An Import in Progress message is displayed on the screen. Leave the window open until the process is completed.

8. Click **OK**.

In this scenario, the existing DSM key material is imported to the AWS KMS. The key state changes from PendingImport to Enabled. The Origin changes to **CCKM**.

Import Using Luna HSM

When importing the key material from Luna HSM, **Select Key Material Origin**. The options are:

- [Create New Key](#)
- [Use Existing Key](#)

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Create New Key

In this method, Luna HSM creates the new key material locally.

1. Select **Create New Key**.
2. Click **Next**. The Add Key Details page is disabled.
3. Enter a **Key Name**.
4. Select the desired **Partition ID**.
5. Select the desired **Key Attributes**.
6. (Optional) Select the **Key Material Expiration Date** from the on-screen calendar.
7. Click **Next**. The Review Key Details screen is displayed.
8. Review the key details. If the details are incorrect or you want to modify them, click **Back** and update the details.
9. Click **Import**.

The key import starts. An Import in Progress message is displayed on the screen. Leave the window open until the process is completed. When the status next to the SOURCE KEY section becomes Complete, the key material is imported successfully.

10. Click **OK**.

In this scenario, the Luna HSM creates a new key material and the key material is imported to the AWS KMS. The key state changes from PendingImport to Enabled. The Origin changes to **CCKM**.

Use Existing Key

In this method, the key material of an existing Luna HSM key is used.

1. Select **Use Existing Key**.
2. Click **Next**. The Add Key Details page is disabled.
3. Select an **HSM Key** from the list.
4. (Optional) Select the **Key Material Expiration Date** from the on-screen calendar.

5. Click **Next**. The Review Key Details screen is displayed.
6. Review the key details. If the details are incorrect or you want to modify them, click **Back** and update the details.
7. Click **Import**.

The key import starts. An Import in Progress message is displayed on the screen. Leave the window open until the process is completed.

8. Click **OK**.

In this scenario, the existing Luna HSM key material is imported to the AWS KMS. The key state changes from PendingImport to Enabled. The Origin changes to **CCKM**.

Deleting Key Material

Note

This functionality is applicable for BYOK keys, not HYOK keys. You can delete [unlinked HYOK keys](#) or [schedule key deletion for linked HYOK keys](#) through AWS menus, but not their source key material.

For HYOK keys, Luna source keys are managed through [CCKM Luna key functions](#), and CipherTrust Manager source keys are managed through [CipherTrust Manager key management menus](#).

To delete key material:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Delete Material**. The Delete Key Material dialog box is displayed.
4. Select **I wish to delete key material**.
5. Click **Delete Key Material**.

A message **AWS Key material deleted** is displayed on the screen. The key state changes to **PendingImport**.

Warning

Be extremely careful when deleting a key material from the AWS KMS. Once the key material is deleted, decryption of data cannot be performed using that key material. However, if needed, you can reimport the key material.

Scheduling Key Deletion

Note

This functionality is applicable to all Native, BYOK, HYOK keys and CloudHSM keys. For HYOK keys, this functionality is applicable to linked keys only.

Scheduled key deletion permanently removes the key from the AWS KMS (if a Native, linked HYOK, or BYOK key) or from the AWS CloudHSM (if a CloudHSM key) at the specified time. The AWS KMS enforces a waiting period of 7 to 30 days. You can cancel schedule deletion before the waiting period ends.

For linked HYOK keys, scheduling deletion removes the local CCKM HYOK key, and the remote KMS key at the specified time. Unlinked HYOK keys can only be [deleted manually](#), not at a scheduled time. Deleting an HYOK key does not delete the source key (stored in either an Luna HSM or CipherTrust Manager. You can later [create a new HYOK key using the source key](#).

Note

In regards to Native and BYOK keys, schedule key deletion is not allowed for multi-region primary keys that have replicas. To schedule deletion of such a key, delete its replica keys first.

Warning

Be extremely careful when scheduling key deletion. Once the key is deleted from the AWS KMS or from the AWS CloudHSM (if a CloudHSM key), it cannot be restored and the data encrypted with this key will be unrecoverable.

For HYOK keys, if you want to temporarily suspend AWS KMS access, you can instead [block the HYOK key](#).

Scheduling Deletion of an AWS Key

To schedule the deletion of an AWS key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Schedule Key Deletion**.
4. On the Schedule Key Deletion screen:
 - a. Select **I wish to delete this key**.

- b. Specify the **Waiting period (in Days)** after which the key will be deleted. The default value is 30.
- c. Click **Schedule Deletion**.

A message **Key <key_name> scheduled for deletion** is displayed on the screen. The key state changes to **PendingDeletion**.

Scheduling Deletion of Multiple AWS Keys

To schedule the deletion of multiple AWS keys:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Select the checkboxes next to the keys you want to schedule for deletion.
4. Click **Schedule Key Deletion**.
5. On the Schedule Key Deletion screen:
 - a. Select **I wish to delete this key**.
 - b. Specify the **Waiting period (in Days)** after which the key will be deleted. The waiting period must be between 7 and 30 days. The default value is 30.
 - c. Click **Schedule Deletion**. The Job Created dialog box is displayed. A job is created to track the deletion of the keys.
 - d. Click **OK**.

You can check the job status on the **Bulk Operations** tab.

Canceling Scheduled Deletion

This functionality is applicable to Native, BYOK, and CloudHSM keys.

Warning

Be extremely careful when scheduling key deletion. Once the key is deleted from the AWS KMS, it cannot be restored and the data encrypted with this key will be unrecoverable.

Canceling the Scheduled Deletion of an AWS Key

To cancel the scheduled deletion of an AWS key:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Cancel Deletion**.

A message Scheduled deletion cancelled is displayed on the screen. The key state changes to Disabled. You can enable the key, if you wish to use this key in the cryptographic operations.

Canceling the Scheduled Deletion of Multiple AWS Keys

To cancel the scheduled deletion of multiple AWS keys:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Select the checkboxes next to the keys whose scheduled deletion you want to cancel.
4. Click **Cancel Key Deletion**. The Cancel Schedule Key Deletion screen opens.
5. Click **Cancel Schedule Deletion**. The Job Created dialog box is displayed. A job is created to track the cancellation of scheduled deletion for AWS keys.
6. Click **OK**.

You can check the job status on the **Bulk Operations** tab.

Deleting Unlinked HYOK Keys

Deleting an unlinked HYOK Key deletes the XKS ID that is associated to a KMS key in AWS KMS. The corresponding KMS key is not deleted, but it can no longer perform cryptographic operations. For linked HYOK keys, you can [schedule a key deletion](#) to delete both the HYOK key on CCKM and the KMS key in AWS KMS.

Warning

Deleting an HYOK key permanently breaks the association between the corresponding KMS key and this particular XKS ID so that KMS key can no longer perform cryptographic operations. Any data encrypted with the KMS key cannot be decrypted. To temporarily suspend AWS KMS access to the HYOK key and its XKS ID, you can instead [block the HYOK key](#).

Deleting an HYOK key does not delete the source key (stored in either an Luna HSM or CipherTrust Manager). You can later [create a new HYOK key using the source key](#). Luna source keys are managed through [CCKM Luna key functions](#), and CipherTrust Manager source keys are managed through [CipherTrust Manager key management menus](#).

To delete an unlinked HYOK key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.

You can filter this list on **Origin: CCKM-HYOK** to display only HYOK keys.

3. Click the overflow icon () corresponding to the desired alias and click **Delete**. The Delete Key dialog box is displayed.
4. Click **Delete Key** to confirm.

Rotating Keys

Key rotation is a critical security best practice that involves updating the cryptographic material used by your keys. CCKM offers several ways to manage key rotation for your AWS keys.

Understanding Key Rotation Mechanisms in CCKM

CCKM supports different approaches to key rotation depending on the key type and origin:

- **Material Rotation (Retaining Key ID and ARN):** This is the primary rotation method facilitated by CCKM's "Rotate Key" feature for symmetric single-region BYOK keys and Native AWS keys, where algorithm is SYMMETRIC_DEFAULT. It involves associating new cryptographic material with the *existing* AWS KMS Key ID and ARN. The key's identity in AWS remains the same, but the underlying material changes. This is particularly relevant for BYOK symmetric keys.
- **AWS-Managed Automatic Rotation:** For Native AWS keys using the SYMMETRIC_DEFAULT algorithm, AWS KMS can automatically rotate the key material. CCKM allows you to enable and manage this feature.
- **HYOK Key Versioning:** For HYOK keys, rotation (termed "Add Version (Rotate)") involves associating a new version of the source key (from CipherTrust Manager or a new Luna HSM key) with the existing HYOK entity in CCKM. The HYOK key in AWS KMS then uses this new material for future operations.
- **CloudHSM Key Rotation:** For keys stored in an AWS CloudHSM custom key store, CCKM triggers a rotation command that instructs AWS CloudHSM to generate new key material for the existing key within the HSM.
- **Manual Alias-Based Rotation (New Key ID and ARN):** As a general key management strategy, for any key type, you can manually achieve a form of rotation by:
 - a. Creating an entirely new AWS key with fresh key material (see [Creating Native Key Material](#) or [Adding Key Material Using External \(BYOK\) Source](#)).
 - b. Removing the alias from the old key.
 - c. Adding that alias to the newly created key.

This method results in a new AWS KMS Key ID and ARN. It's a manual process and distinct from CCKM's "Rotate Key" feature which often aims to retain the existing Key ID.

General Steps to Initiate Key Rotation via CCKM UI

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.
3. Locate the key you wish to rotate. Click the overflow icon () corresponding to its alias.
 - For **Native**, **BYOK**, or **CloudHSM** keys, click **Rotate Key**.
 - For **HYOK** keys, click **Add Version (Rotate)**.

The subsequent steps depend on the type of key you are rotating, as detailed in the sections below.

Rotating Native AWS Keys

The process for rotating Native AWS keys varies based on their configuration.

1. Native Keys with SYMMETRIC_DEFAULT Algorithm (AWS-Managed Rotation)

If you are rotating an AWS native key that uses the **SYMMETRIC_DEFAULT** algorithm, CCKM can initiate an AWS on-demand key rotation or manage AWS's automatic rotation schedule.

- **On-Demand Rotation**
 - After clicking **Rotate Key**, an "AWS On-Demand Key Rotation" dialog box may appear.
 - Click **Rotate** in the dialog, then **OK** to confirm.
 - Alternatively, this option might be available in the key's "View/Edit" screen under "ROTATION HISTORY". Refer to [ROTATION HISTORY](#) for details.

Warning

AWS KMS may have a default limit on how many times you can initiate on-demand key rotation (for example, 10 times by default). Consult AWS documentation for the most current information.

- **Scheduled Automatic Rotation:** You can enable AWS automatic key rotation and set a rotation period via the key's "View/Edit" screen under "AWS AUTOMATIC KEY ROTATION".

Note

For these AWS-managed rotations, you do not select new key material from an external source like CipherTrust Manager; AWS manages the generation of the new key material.

2. Native Asymmetric Keys or Other Native Key Types

For other types of Native AWS keys (such as asymmetric keys, HMAC keys, etc.), key rotation will create a new BYOK key and reassign the alias from the existing Native AWS key to the newly created BYOK key. Follow the instructions for [Material Rotation \(BYOK and applicable Native Keys\)](#) below.

Rotate BYOK Keys

The process for rotating AWS BYOK keys varies based on their configuration.

1. Single region BYOK keys with SYMMETRIC_DEFAULT Algorithm

When you rotate BYOK keys, CCKM's Rotate Key feature helps update the cryptographic material for the existing AWS Key ID and ARN. You provide new key material from a supported source. Follow the instructions for [Material Rotation \(BYOK and applicable Native Keys\)](#) below.

Alternatively, this option might be available in the key's "View/Edit" screen under "ROTATION HISTORY". Refer to [ROTATION HISTORY](#) for details.

Warning

AWS KMS may have a default limit on how many times you can initiate on-demand key rotation (for example, 10 times by default). Consult AWS documentation for the most current information.

Note

- **Key Material Requirement:** The new key material you upload (whether created as a new source key in CipherTrust Manager/HSM or by selecting an existing source key from these systems) **must not have been previously imported or associated with this specific AWS KMS key**. Each rotation requires distinct cryptographic material for the target AWS KMS key.
- If you prefer the traditional method of creating an entirely new key with a new ID/ARN, or if specific configurations necessitate it, you can achieve rotation manually as described in [Understanding Key Rotation Mechanisms in CCKM](#) under "Manual Alias-Based Rotation".

2. Other BYOK Keys

For other types of BYOK keys (such as asymmetric keys, multi-region symmetric keys, or HMAC keys), key rotation creates a new BYOK key and moves the alias from the existing key to the new one.

Steps for Material Rotation (BYOK and applicable Native Keys)

1. After clicking **Rotate Key** (as described in [General Steps to Initiate Key Rotation](#)), the **Select Material Origin**

screen appears.

2. Choose the source for the new key material. Options typically include:
 - **Upload New CipherTrust (External) Key:** Create a new key on an external CipherTrust Manager.
 - **Upload Existing CipherTrust (External) Key:** Use an existing key from an external CipherTrust Manager.
 - **Upload New Vormetric DSM Key:** Create a new key on a Vormetric DSM. (Available for SYMMETRIC_DEFAULT/AES keys)
 - **Upload Existing Vormetric DSM Key:** Use an existing key from a Vormetric DSM. (Available for SYMMETRIC_DEFAULT/AES keys)
 - **Upload New Luna HSM Key:** Create a new key on a Luna HSM.
 - **Upload Existing Luna HSM Key:** Use an existing key from a Luna HSM.
 - **Upload New CipherTrust (Local) Key:** Create a new key on the local CipherTrust Manager.
 - **Upload Existing CipherTrust (Local) Key:** Use an existing key from the local CipherTrust Manager.

Note

When rotating BYOK keys with the SYMMETRIC_DEFAULT algorithm, do not reuse any previously used key material.

Select your desired option and click **Next**.

3. Configure source key details. The next screen will prompt you to either create a new source key or select an existing one, based on your choice.
 - If you selected **Upload New ... Key:**
 - For CipherTrust (External) Key:
 - a. **Select Domain** from the drop-down list.
 - b. Specify a unique **Key Name** for the new source key.
 - For CipherTrust (Local) Key: Specify a unique **Key Name** for the new source key.
 - For Vormetric DSM Key:
 - a. Specify a unique **DSM Key Name**.
 - b. Select the desired **DSM Domain**.

- For Luna HSM Key:

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

- Specify a unique **Key Name** for the new source key.
 - Select the desired **Partition ID**.
 - Review the displayed **Key Attributes**.
- If you selected **Upload Existing ... Key**:
 - For CipherTrust (External) Key: Select the desired **CipherTrust (External) Key** from the list.
 - For CipherTrust (Local) Key: Select the desired **CipherTrust (Local) Key** from the list.
 - For Vormetric DSM Key: Select the desired **DSM Key** from the list.
 - For Luna HSM Key:

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Select the desired **HSM Key** from the list.

Click **Next**.

- Configure destination key properties (Add Labels screen).
- Click the desired tab to view the instructions.

Single-region BYOK keys with the SYMMETRIC_DEFAULT algorithm

- (Optional) Provide a **Description** for this rotation instance or update the key's description.
- (Optional) Set or update a key **Expiration Date** for the new material:
 - Select the **Expiration Date** check box.

-
- b. Enter the date and time (for example, **MM/DD/YYYY HH:MM**) or use the calendar.
 - c. Click **Next**.
 - d. Review and update the rotate key material details that you have provided. These details are divided into MATERIAL ORIGIN and CONFIGURATION sections.
 - e. Before rotating the key, review all details. After the key is rotated, certain features will no longer be editable.
 - f. Click **Rotate**.
 - g. Click **OK**. The Rotate Key Material wizard is closed.

Other BYOK Keys

The **Alias** of the current AWS key being rotated is pre-populated.

- a. (Optional) Provide a **Description** for this rotation instance or update the key's description.
 - b. (Optional) Set or update a key **Expiration Date** for the new material:
 - a. Select the **Expiration Date** check box.
 - b. Enter the date and time (for example, **MM/DD/YYYY HH:MM**) or use the calendar.
 - c. (Optional) For BYOK or Native keys, you can grant or deny the encrypt permission to the key policy in the AWS account where the key is created in OR to the key policy in *all accounts* the key is shared in:
 - Select **Disable Encrypt Permissions on Current Key** to deny the encrypt permission to the key policy in the AWS account where the key is created in. Clear the check box to grant the encrypt permission. By default, **Disable Encrypt Permissions on Current Key** is selected, that is, the encrypt permission is denied to the current key.
 - Select **Disable Encrypt Permissions on All Accounts** to deny the encrypt permission to the key policy in all the AWS accounts the key is shared in. Clear the check box to grant the encrypt permission.
 - d. (Optional) Select the **Apply Gravestone Alias on Current Key** check box to retain the current key's alias (with a timestamp) on the archived key material/version after rotation. If not selected, the alias is not retained on the old key version.
 - e. Click **Save**. A message will confirm successful key material creation (if applicable) and key rotation for the AWS key.
-

Rotating CloudHSM Keys

Rotating a CloudHSM key involves CCKM instructing AWS CloudHSM to generate new key material for the existing key within the CloudHSM cluster. The Key ID in AWS KMS remains the same.

1. After clicking **Rotate Key** (as described in General Steps to Initiate Key Rotation), the Rotate Key screen (or a similar confirmation screen) appears.
2. (Optional) Provide a **Description** for this rotation instance.
3. Disable encrypt permissions on the current key (old material).
 - a. (Optional) For BYOK or Native keys, you can grant or deny the encrypt permission to the key policy in the AWS account where the key is created in OR to the key policy in *all accounts* the key is shared in:
 - b. Select **Disable Encrypt Permissions on Current Key** to deny the encrypt permission to the key policy in the AWS account where the key is created in. Clear the check box to grant the encrypt permission. By default, **Disable Encrypt Permissions on Current Key** is selected, that is, the encrypt permission is denied to the current key.
 - c. Select **Disable Encrypt Permissions on All Accounts** to deny the encrypt permission to the key policy in all the AWS accounts the key is shared in. Clear the check box to grant the encrypt permission.

By default, encrypt permissions are disabled on the current (old) key material after rotation. Clear the check box to keep them enabled on the old material.
4. (Optional) Select the **Apply Gravestone Alias on Current Key** checkbox if you wish to retain the key alias (with a timestamp) on the archived key material *after* rotation. If not selected, the alias is not retained on the old key version.
5. Click **Save** (or **Rotate**). A message will confirm successful key rotation.

Rotating HYOK Keys (Add Version / Rotate)

Rotating a Hold Your Own Key (HYOK) involves associating new cryptographic material (either a new source key or a new version of an existing source key from your CipherTrust Manager or Luna HSM) with the HYOK entity in CCKM. This new material will be used for future encryption operations by the corresponding AWS KMS key. Decryption operations on data encrypted with previous key material will continue to use the older material.

Note

After rotation, the HYOK key in CCKM (and by extension, the AWS KMS key) is associated with multiple source keys/versions.

- **New Encrypt Operations:** Will use the newly rotated (latest) key material.
- **Decrypt Operations:** Will use the key material version that originally encrypted the data.

Warning

- **Blocking/Unblocking:** To temporarily suspend AWS KMS access to an HYOK and its source key(s), block the HYOK key. This is reversible.
- **Deleting HYOK Key in CCKM:** Deleting the HYOK key in CCKM or scheduling its deletion can make data encrypted with the corresponding AWS KMS key unrecoverable.
- **Deleting Source Key:** Deleting the underlying source key directly in Luna HSM or CipherTrust Manager will also render data unrecoverable. Manage source keys via their respective management interfaces (Luna HSM keys or CipherTrust Manager keys).

Steps

After clicking **Add Version (Rotate)** (as described in General Steps to Initiate Key Rotation), the **Select Material Origin** screen (or a direct confirmation for CM key versioning) appears.

Option 1: Create New CipherTrust Manager Key Version (for CipherTrust Manager sourced HYOK keys)

If your HYOK key's source is a CipherTrust Manager key, rotation typically involves creating a new version of that existing source key.

1. From the "Add Version (Rotate) Key" screen (this might be a direct confirmation dialog if only this option is available for the key), click **Rotate Key** (or similar confirmation) to proceed.
2. Leave the dialog open until the rotation operation completes successfully.

Option 2: Create a New Luna HSM Key (for Luna HSM sourced HYOK keys)

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

You can create an entirely new source key in your Luna HSM to associate with the HYOK.

1. On the Select Material Origin screen, select **Create New Luna HSM Key**.
2. Click **Next**. The Configure Source Key screen is displayed.
3. Provide a new **Source Key Name** for the key to be created in the Luna HSM.

Note

Key attributes for the new Luna HSM key will be displayed and are typically not editable, ensuring they meet HYOK requirements.

4. Click **Next** to proceed to the **Review and Rotate Key** screen.
5. Review the key details and click **Rotate** to confirm.
6. Leave the dialog open until the rotation operation completes successfully.

Option 3: Use Existing Luna HSM Key (for Luna HSM sourced HYOK keys)**Note**

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

You can use another existing AES-256 key from your Luna HSM as the new cryptographic material for the HYOK.

1. On the Select Material Origin screen, select **Use Existing Luna HSM Key**.
2. Click **Next**. The Configure Source Key screen is displayed.
3. Select the desired key from the **Select Luna Key** drop-down list.

Note

The existing Luna HSM key must be AES-256 and have the following attributes:

- CKA_EXTRACTABLE = FALSE
- CKA_SENSITIVE = TRUE
- CKA_ENCRYPT = TRUE
- CKA_DECRYPT = TRUE
- CKA_WRAP = TRUE
- CKA_UNWRAP = TRUE

4. Click **Next** to proceed to the **Review and Rotate Key** screen.
5. Review the key details and click **Rotate** to confirm.
6. Leave the dialog open until the rotation operation completes successfully.

Blocking and Unblocking HYOK Keys

Blocking and unblocking an HYOK key is a way to temporarily suspend and restore AWS KMS's access to the HYOK key and its source key. This way, you do not [permanently delete the HYOK key and the XKS ID](#) which AWS KMS needs to communicate to the HYOK and source key.

Block a Key

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.

You can filter this list on **Origin: HYOK-CCKM** to display only local HYOK keys.

3. Click the overflow icon () corresponding to the desired key and click **Block Key**.
4. Confirm you wish to block the key by clicking **Block** in the Block Key dialog.

Unblock a Key

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. The list of available AWS keys is displayed.

You can filter this list on **Origin: HYOK-CCKM** to display only local HYOK keys.

3. Click the overflow icon () corresponding to the desired key and click **Unblock Key**.
4. Confirm you wish to unblock the key by clicking **Unblock** in the Block Key dialog.

Scheduling Operations

CCKM allows you to periodically refresh or rotate keys in the background. To refresh or rotate keys, you need to create schedule configuration. A schedule configuration defines when to run a refresh or rotation job. You can use either basic format or cron format to specify the time when the rotation or refresh job will run.

Specify basic format in the following order:

[Repeats, at]

Where,

- Repeats: frequency of the scheduler. Possible values are daily, weekly, monthly, and yearly.

- **at:** specific time at which the rotation or refresh operation will be performed. Possible value lies between 12:00-11:00 AM/PM UTC.

Specify cron format in the following order:

```
" * * * * * "
```

These five values indicate when the job should be executed. These values are mandatory and must be specified in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

Field	Allowed values	Allowed special characters
Minute	0-59	* / , -
Hour	0-23	* / , -
Day of month	1-31	* / , - ?
Month	1-12 or JAN-DEC	* / , -
Day of week	0-6 or SUN-SAT	* / , - ?

Adding Key Rotation Schedule

A key rotation schedule replaces selected keys with new cryptographic keys at the specified time.

To add key rotation schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. **Select Schedule Type as Key Rotation.**
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - From the **Cloud Name** drop-down list, select **AWS**.
 - Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. Set the values on the **Schedule Config** screen.

You can schedule configuration for:

- **All Keys:** Refer to [Configuring Rotation Schedule for All Keys](#) for details.
- **Only Expiring Keys:** Refer to [Configuring Rotation Schedule for Expiring Keys](#) for details.
- **All Keys Based on Creation/Last Rotation Date:** Refer to [Configuring Rotation Schedule for All Keys Based on Creation or Last Rotation Date](#) for details.

Configuring Rotation Schedule for All Keys

To configure rotation schedule for all keys:

1. Configure **DURATION** for the scheduler. Specify the following values:

- a. **Schedule Starts**
- b. **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.

2. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either the **Basic** or the **Raw (Cron)** format to specify the value.
3. (Optional) Set the expiry of the new key that will be created through rotation. If not specified, the new key material never expires. For example, if you set the expiry time to **6 Hours**, the key material of the new key will expire in 6 hours.
4. (Optional) Enable **Rotate Key Material (Imported AES Keys)**.

Warning

Key material rotation is only available for imported symmetric AES single-region keys (BYOK). Other key types can be rotated by creating a new key and reassigning the alias.

5. Select **Apply Gravestone Alias on Current Key** to retain the key alias with timestamp on the archived key after rotation. The format of the Gravestone alias is `<current alias>-rotated-<timestamp>`. The alias will only be applied to keys with an existing alias.
6. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Configuring Rotation Schedule for Expiring Keys

To configure rotation schedule for expiring keys:

1. In the **SELECTION** section, select the keys to be rotated based on their expiry time. In the **Only rotate keys within _ of material expiration** field, select or enter the key expiry time. For example, if the value of **Only rotate keys within _ of material expiration** field is set to **4 Days**, then the rotation schedule is created for all the keys that will expire in 4 days.
2. Configure **DURATION** for scheduler. Specify the following values:
 - a. **Schedule Starts**
 - b. **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.
3. (Optional) Set the expiry of the new key that will be created through rotation. If not specified, the new key material never expires. For example, if you set the expiry time to **6 Hours**, the key material of the new key will expire in 6 hours.
4. (Optional) Enable **Rotate Key Material (Imported AES Keys)**.

Warning

Key material rotation is only available for imported symmetric AES single-region keys (BYOK). Other key types can be rotated by creating a new key and reassigning the alias.

5. Select **Apply Gravestone Alias on Current Key** to retain the key alias with timestamp on the archived key after rotation. The format of the Gravestone alias is `<current alias>-rotated-<timestamp>`. The alias will only be applied to keys with an existing alias.
6. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Configuring Rotation Schedule for All Keys Based on Creation or Last Rotation Date

To configure rotation schedule for all keys based on their creation or last rotation date:

1. In the **SELECTION** section, select the keys to be rotated based on their creation or last rotation date.

In the **Only rotate keys after _ of creation/last rotation date** field, select or enter the number of days, weeks, months, or years. For example, if the value of **Only rotate keys after _ of creation/last rotation date** field is set to **4 Days**, then all the keys will rotate after 4 days of their creation or last rotation date. The default duration is 7 Days.
2. Configure **DURATION** for scheduler. Specify the following values:

a. **Schedule Starts**

b. **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.

3. (Optional) Set the expiry of the new key that will be created through rotation. If not specified, the new key material never expires. For example, if you set the expiry time to **6 Hours**, the key material of the new key will expire in 6 hours.
4. (Optional) Enable **Rotate Key Material (Imported AES Keys)**.

Warning

Key material rotation is only available for imported symmetric AES single-region keys (BYOK). Other key types can be rotated by creating a new key and reassigning the alias.

5. Select **Apply Gravestone Alias on Current Key** to retain the key alias with timestamp on the archived key after rotation. The format of the Gravestone alias is `<current alias>-rotated-<timestamp>`. The alias will only be applied to keys with an existing alias.
6. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Adding Key Refresh Schedule

Note

With support for on-demand rotation of both native and imported BYOK keys, the refresh operation now takes longer than in previous releases. This is because detailed rotation information must be synchronized from AWS KMS to CCKM.

To add a key refresh schedule:

1. Open the **Cloud Key Manager** application.
 2. In the left pane, click **Schedules**. The list of available schedules is displayed.
 3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
 4. Select **Schedule Type** as **Key Refresh**.
-

-
5. Click **Next**.
 6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **AWS** from the **Cloud Name** drop-down list.
 - Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
 7. On the **Schedule Config** screen, enter or select the following details.
 - a. Configure **Duration** for the scheduler. Specify the following values:
 - **Schedule Starts**
 - **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.
 - b. Configure **Frequency** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
 - c. Click **Next**.
 8. On the **Add Accounts** screen:
 - a. Select the account name from the **Refresh Keys from Selected Account(s)** drop-down list.
 - b. Click **+** button.
 - c. Similarly, you can add more accounts.
 - d. Select the **Refresh All Accounts** option if you want to refresh all the existing accounts.
 - e. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Automatic Cloud Key Discovery

A KMS container schedule automatically detects the KMS containers based on the AWS Cloud connection and adds them to the CipherTrust Cloud Key Manager.

Note

When adding a KMS container schedule, select a connection that has access to the management account of your AWS organization. The connection must have permissions to list all organizational units (OUs) and accounts within the organization.

Refer to [configurations to discover all AWS accounts](#) for details.

Adding a KMS container (vault or subscription) to CCKM consumes one CCKM license count.

To add a KMS container schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Select Schedule Type screen of the Add New Schedule wizard is displayed.

Schedule Type

1. Select Schedule Type as **Add KMS Container**.
2. Click **Next**. The General Info screen is displayed.

General Info

1. Enter a unique **Schedule Name**.
2. Select **AWS** from the **Cloud Name** drop-down list.
3. (Optional) Add **Description** for the scheduler in a maximum of 250 characters.
4. (Optional) Select **Enable Schedule** to enable the schedule.
5. Select a job type from **Define Job Type**. The options are:
 - **Discover Only**: Lists all the discovered containers.
 - **Discover and Add**: Adds all the discovered containers to the CCKM.
6. (Optional) Enable **Archive Accounts**.
7. Click **Next**. The Connection screen is displayed.

Connection

1. Select a **Connection**.
2. Enter a **Role**.

3. (Optional) Enter a **AWS Role External ID**.

4. (Optional) Add the Filters.

a. Select a **Name**.

You can select only one of the following names: [Root] Arn, [Root] Id, [Root] Name, [Root] Tag, [OrganizationalUnit] Arn, [OrganizationalUnit] Id, [OrganizationalUnit] Name, [OrganizationalUnit] Tag, [Account] Arn, [Account] Email, [Account] Id, [Account] Name, and [Account] Tag.

b. Enter a **Value**.

c. Click **+** (plus).

Note

- A name can have only one value.
- You can add multiple filters.

5. Click **Next**. The Region & ACLs screen is displayed.

Region & ACLs

1. (Optional) In Select Regions section, add regions to the schedule.

a. Select the desired regions.

b. Click right arrow (). The selected regions move to the **All Selected Regions** list.

2. (Optional) In the ACCESS CONTROL section.

a. Click **Keys**, **Certificates**, **Secrets**, or **Reports** tab.

b. Click the **All Keys**, **BYOK Keys**, **Native Keys**, **HYOK Keys**, **Cloud HSM Keys**, **Custom Key Store**, or **Reports** tab.

c. Click **Assign User/Group**. The Assign User/Group dialog box is displayed.

d. Select the desired user or group from the **User/Group** drop-down list.

e. Click **Save**. The newly added user/group is displayed under **Name** in the ACCESS CONTROL section.

f. Grant the permissions to the user/group, as appropriate.

Granting additional permissions

To grant additional permissions, select the check box under the desired operation corresponding to the

desired users or groups. You can grant the following permissions.

- **All Keys:** Cancel Key Delete, Schedule Key Delete, Delete Material, Import Material, Rotate Key, Refresh Key, and Edit Key.
- **BYOK Keys:** View Key and Add Key.
- **Native Keys:** View Key and Add Key.
- **HYOK Keys:** View Key, Add Key, Block/Unblock, Link Key, and Delete Key.
- **Cloud HSM Keys:** View Key, Add Key, and Delete Key.
- **Custom Key Store:** View, Add, Edit, Delete, Block, Unblock, Connect, Disconnect, and Link.
- **Reports:** View Report, Add Report, Download Report, and Delete Report.

To remove a permission, clear the check box under the desired operation.

g. Remove a user or group.

- a. Under **Unassign**, click **X** corresponding to the desired user/group. The Remove Group dialog box is displayed.
- b. Click **Remove**.

3. Click **Next**. The Schedule Config screen is displayed.

Schedule Config

1. Configure **DURATION** for the scheduler. Specify the schedule start and end time:

- **Schedule Starts:** Specify time when the schedule starts.
- **Schedule Ends:** Unavailable by default, that is, the scheduler never expires.
- **Never:** Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.

2. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.

3. Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created schedule is displayed in the schedules list.

After the schedule is run successfully, AWS KMS Accounts will be added to the CCKM. You can see them in the Vaults section, of the AWS KMS Accounts page in the CCKM.

Adding Credential Rotation Schedule (AWS XKS only)

A credential rotation schedule auto rotates the selected key store's credentials with new credentials. You can assign a credential rotation schedule for the credentials of a given external custom key store in CCKM. You must first create one or more rotation schedules for the key store's credentials. Thereafter, these schedules are made available within a list under **Schedules** of the details page for the given custom key store, which you can then assign to the key store.

To add credential rotation schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. **Select Schedule Type as Credential Rotation (AWS XKS only).**
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. On the **Schedule Config** screen, enter or select the following details.
 - a. Configure **Duration** for the scheduler. Specify the following values:
 - **Schedule Starts**
 - **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.
 - b. Configure **Frequency** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
 - c. Click **Save**.

Viewing/Editing Schedules

To view/edit a schedule:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **Schedules**. The Schedules page displays following details:

Field	Description
Name	Unique name of the scheduler configuration.
Schedule Type	Type of the schedule. Possible types are: • Key Rotation • Key Refresh • Add KMS Container (GCP, Azure, and AWS only) • Credential Rotation (AWS XKS only)
Cloud Name	Name of the cloud. Note: For the schedule type of Credential Rotation, only the AWS cloud is supported.
Last Modified	Time when the schedule is modified.
Frequency	Frequency of the scheduler configuration.
Start Date	Creation time of the scheduler configuration.
End Date	Expiry time of the scheduler configuration.
Status	Status of the scheduler configuration. Possible values are: • Enabled • Disabled

3. Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.
4. Edit or configure the following fields:
- Description.
 - Status of the scheduler configuration.
 - Connection settings, refer to [Connection](#) for details.
 - (Applicable to KMS container schedule) Regions and Access Control settings. Refer to [Region & ACLs](#) for details.
 - Scheduler configuration parameters, such as duration and frequency. Refer to [Schedule Config](#) for details on key rotation schedule fields.
5. Click **Update** to save the changes.

JOB HISTORY

The Schedules page also contains a section named JOB HISTORY. Every time a schedule is run, a Job is created.

-
- This section displays information related to a job such as **Run Date**, **Job ID**, and **Status**.
 - To view the details of the job, click the **Job ID** link corresponding to the desired job. The Job details page is displayed. It shows the list of AWS Accounts along with their Health and Status.
 - The health can be:
 - **Error**: Failed to manage the KMS accounts.
 - **Warning**: KMS accounts managed partially.
 - **Healthy**: Successfully managed all the KMS accounts.
 - The status can be:
 - **FailedToAccess**: KMS account can't fetch AWS regions because it doesn't have access to regions.
 - **AlreadyManaged**: Regions are already managed by a KMS container within the KMS account.
 - **NewlyManaged**: Regions are newly managed by a KMS container within the KMS account.
 - **FailedToManage**: Regions couldn't be managed by a KMS container within the KMS account.
 - **NotManaged**: None of the regions is managed by a KMS container within the KMS account.
 - **Archived**: All regions are archived by a KMS container within the KMS account.

Note

- You can view the details of successfully completed jobs that belong to a KMS container schedule.

- To download the reports of AWS Accounts, go to the top left of the Job details page, click **Download**, and select an **AWS Accounts** from the drop-down menu.

Disabling Schedules

To disable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Disable**. The Disable Schedule message is displayed.
4. Click **Disable Schedule**.

A message **Successfully disabled the schedule** is displayed on the screen.

Enabling Schedules

To enable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Enable**. The Enable Schedule message is displayed.
4. Click **Enable Schedule**.

A message **Successfully enabled the schedule** is displayed on the screen.

Manually Running Schedules

To manually run a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Run Now**.

The Run Now schedule is started in the background. A message **Schedule is running now. It will take a few seconds to finish.** is displayed on the screen. After the schedule is run successfully, a message **Successfully finished running the schedule** is displayed.

Deleting Schedules

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Delete Schedule**. The Delete Schedule message is displayed.
4. Click **Delete**.

A message **Successfully deleted schedule** is displayed on the screen.

Managing AWS Reports

CCKM provides options to generate key visibility reports based on:

- Key-related activities between CCKM and AWS KMS
- Track keys by their expiration dates

- What applications are using the keys

AWS reports are categorized as:

- **Key Activity Report:** Inspect individual AWS key histories by operations, for example, when they were refreshed, rotated, edited, or deleted. Also, use this report to compare key activities between CCKM and AWS KMS.
- **Key Aging Report:** Track keys by their expiration dates. Audit a range of dates, from past material deletions to future scheduled deletions, within the selected AWS KMS account.
- **Service/Usage Report:** Monitor key usage by tracking services and applications consuming the keys. View when and where a service requests the use of each key.

Prerequisites

For generating CCKM reports, logs are fetched from AWS using the CloudTrail and CloudWatch services. You need to configure these services on the **AWS console**. This includes specifying trail name, storage location, log group name, IAM role, and event types.

Attach the following policy to the IAM role linked with the log group:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:DescribeLogGroups",
        "logs:FilterLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Creating AWS Reports

CCKM users with the **Add Report** permission can create reports.

To create an AWS report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > AWS**. The AWS Reports page is displayed.
3. Click **Add Report**. The Choose Report Type and Name screen of the Add AWS Report wizard is displayed.
4. Under Select Report Type, select a report type. The options are:
 - Key Activity Report

- Key Aging Report
- Service/Usage Report

- Specify a **Report Name**. This is a mandatory field.
- Click **Next**. The Select Log groups screen of the wizard is displayed.
- Select an **AWS KMS** from the drop-down list.
- (Not applicable to Key Aging Report)* Select a **Log Region** from the drop-down list. The drop-down list shows the log regions of the selected AWS KMS account.

When a log region is selected, its log groups (if any) are populated in the Log Group Name drop-down list.

- (Not applicable to Key Aging Report)* Select a **Log Group Name** from the drop-down list.
- Click **Add**. The selected log group is displayed under Selected Log Groups. Add more log groups, if required.
- Click **Next**. The Set Start and End Dates screen of the wizard is displayed.
- In the **Include Entries From** field, specify the start date and time for the report.
 - Click the field and select the date and time on the on-screen calendar.
 - Alternatively, enter the time in **MM/DD/YYYYHH:MM** format.
- In the **To** field, specify the end date and time for the report.
- Click **Save**.

A success message **Created report** is displayed. The newly created report is displayed under the AWS reports list.

Viewing AWS Reports

CCKM users with the **View Report** permission can view reports.

The AWS Reports page displays the list of existing reports. Filter the reports by their names.

To view existing AWS reports:

- Open the **Cloud Key Manager** application.
- In the left pane, click **Reports > AWS**. The list of available reports is displayed. The following details of the reports are displayed:

Column Name	Description
Report Name	Name of the report.

Column Name	Description
Run Date	When the report is run.
Type	Type of the report. The type can be: • Key Activity Report • Key Aging Report • Service/Usage Report
Start Date	Start date from when the report is generated. If a report is run now, its status becomes Running Now.
End Date	End date for the report.

Viewing Details of an AWS Report

CCKM users with the **View Report** permission can view reports.

To view the details of an AWS report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > AWS**. The list of available reports is displayed.
3. Click the desired **Report Name** link. The details of report are displayed.

Alternatively, click the overflow icon () corresponding to the desired alias and click **View**.

Downloading AWS Reports

CCKM users with the **Download Report** permission can download reports.

To download an AWS report as a CSV file:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > AWS**. The list of available reports is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Download**.

Alternatively, click the desired report under **Report Name**, and click **Download CSV** in the detail view.

The report is downloaded as a CSV file.

Deleting AWS Reports

CCKM users with the **Delete Report** permission can delete reports.

To delete an AWS report:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **Reports** > **AWS**. The list of available reports is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Delete**. The Confirm Delete Report message is displayed.
4. Click **Delete Report**.

The report is removed from the list of reports.

Managing Policies

Policies can be applied to new AWS keys during and after creation. Saved policies (policy templates) are supported for all Native, BYOK, and CloudHSM keys.

For HYOK keys, policy templates are applicable to linked keys only.

Note

- Consult the [AWS Key Management Service Developer Guide](#) for details on the effects of AWS key policies.
- The `ListResourceTags`, `TagResource`, and `UntagResource` permissions are needed to manage saved policies.

This section covers the following topics:

- [Creating New Policies](#)
- [Viewing Saved Policies](#)
- [Viewing Details of Saved Policies](#)
- [Updating Saved Policies](#)
- [Deleting Saved Policies](#)

Creating New Policies

You must have the **Add Key** (BYOK, HYOK, or Native) or **CloudHSM - Add Key** permission to create a policy.

To create a policy:

1. Open the **Cloud Key Manager** application. By default, the AWS Keys tab is selected.
2. Click the **Saved Policies** tab.
3. Click **Create Policy**. The Select Account and Name Policy screen is displayed.

Select Account and Name Policy

1. Specify a unique policy **Name**.
2. Select the **AWS KMS Account**. The policy will be added to this account.
3. (Optional) Enable **Create Policy at Account ID Level**.

Note

You can create the policy with a KMS Account or Account ID.

4. Click **Next**. The Configure Policy screen is displayed.

Configure Policy

Select the key admins, key users, and additional accounts for the policy. The **Key Admins** and **Key Users** tabs display the list of available AWS users and roles.

You can configure policy in either the Basic view or Raw view. The default view is Basic.

In the Basic view, Key Admins is the default tab.

1. On the **Key Admins** tab, select the desired key admins.
2. Click the **Key Users** tab.
3. Select the desired key users.
4. Add external accounts in the **Add Accounts** field and click **+**.

These accounts have user permissions. Similarly, you can add more external accounts. To remove an account, click the close (X) icon in the account name.

To switch to the Raw view, click **Switch to policy (Raw) View**. In the Raw view, paste a policy in JSON format in the **Raw Policy** field.

Note

Any edits made in the policy (Raw) view will be retained exclusively in the policy (Raw) view.

5. Click **Next**. The Review and Add screen is displayed.

Review and Add

Before adding the policy, review all the details. After the policy is added, certain features cannot be edited.

1. Review the policy and account details. If the details are incorrect or you want to modify them, click **Back** and update the details.
2. Click **Add Policy**.

The newly created policy appears on the Saved Policies tab. The status of the policy is Unverified.

Viewing Saved Policies

The Saved Policies tab of the AWS Keys page shows the list of existing key policies. Filter the policies by Policy Name, Account Name, or Account ID.

You must have the **View Key** (BYOK, HYOK, or Native) / **CloudHSM - View Key** permission to view policies.

To view the saved policies:

1. Open the **Cloud Key Manager** application. By default, the AWS Keys tab is selected.
2. Click the **Saved Policies** tab. The list of available policies is displayed. The tab shows the following details:

Column Name	Description
Policy Name	Name of the policy.
Status	Status of the policy - Verified or Unverified. If a policy is linked with an AWS key, its status is Verified, otherwise, the status is Unverified.
Account Name	Name of the AWS KMS account.
Account ID	ID of the AWS KMS account.
Cloud	Name of the AWS cloud - AWS, AWS Gov Cloud, or AWS China.
Creation Date	Time when the policy is created.

Viewing Details of Saved Policies

The detail view of a policy shows the associated keys and the policy.

You must have the **View Key** (BYOK, HYOK, or Native) / **CloudHSM - View Key** permission to view the policies.

To view the details of a policy:

1. Open the **Cloud Key Manager** application.
2. Click the **Saved Policies** tab. The list of available policies is displayed.
3. Click the **Policy Name** link of the desired policy.

Alternatively, click the overflow icon () corresponding to the desired policy and click **View Details**.

The detail view shows the details of the associated keys and the policy under KEYS and POLICY sections.

Updating Saved Policies

You must have the **Edit Key** (BYOK, HYOK, or Native) or **CloudHSM - Add Key** permission to create a policy.

To update saved policies:

1. Open the **Cloud Key Manager** application.
2. Click the **Saved Policies** tab. The list of available policies is displayed.
3. Click the overflow icon () corresponding to the desired Saved Policy and click **View Details**.

Alternatively, click the saved policy link under the Policy Name field to view the Saved Policy details.

4. Scroll down to the POLICY section, and select the view to update the policy, the options are **Raw View** and **Basic View**.
5. Click the desired tab to view the instructions.

Raw View

Make the necessary changes in the policy.

Basic View

In the Basic View, Key Admins is the default tab.

- a. On the **Key Admins** tab, select the desired key admins.
- b. Click the **Key Users** tab.
- c. Select the desired key users.

Note

If you make changes in one of the views and switch to the other view without updating the policy, you will see a **Warning** message. Click **Yes** to continue.

6. Click **Update**. The **Update Policy** message is displayed.

Note

To save the changes as a new policy, select the **DO NOT Push Changes** check box, and enter **New Policy**.

-
7. Click **Apply**.

Deleting Saved Policies

You must have the **Schedule Key Delete**, **Delete Key (HYOK)**, or **CloudHSM - Delete Key** permission to delete a policy.

Only the policies with the status **Unverified** can be deleted. If a policy is associated with (or being used by) an AWS key, it cannot be deleted. So, before deleting a policy, make sure that is not associated with any AWS key.

To delete a policy:

1. Open the **Cloud Key Manager** application. By default, the **AWS Keys** tab is selected.
2. Click the **Saved Policies** tab.
3. Click the overflow icon () corresponding to the desired policy.
4. Click **Delete Policy**. A message box appears prompting to confirm the policy deletion.

Deleting a policy is permanent and cannot be undone. After deletion, the policy cannot be retrieved.

5. Click **Delete Policy**. The policy is deleted.

Managing AWS Templates

The templates store and manage AWS policy templates (in JSON format) to be used as a base for creating key policies.

This section covers the following topics:

- [Creating Templates](#)
- [Viewing Templates](#)
- [Updating Templates](#)
- [Deleting Templates](#)

Creating Templates

You must have the **Add Key** permission to create a template.

To create a template:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. By default, the **Keys** tab is selected.
3. Click the **Templates** tab.

4. Click **Create Template**. The Create Template wizard is displayed.
5. Enter **Name**.
6. Enter **JSON**.
7. Click **Save**.

A success message is displayed on the screen.

Viewing Templates

To view the templates:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. By default, the Keys tab is selected.
3. Click the **Templates** tab. The list of available templates is displayed. The tab shows the following details:

Column Name	Description
Template Name	Name of the template.
Creation Date	Time when the template is created.

Updating Templates

You must have the **Edit Key** permission to update a template.

To update the templates:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > AWS**. By default, the Keys tab is selected.
3. Click the **Templates** tab. The list of available templates is displayed.
4. Click the **Template Name** link of the desired template. The View/Edit Template wizard is displayed.
5. Update the **JSON** details.
6. Click **Update**.

A success message is displayed on the screen.

Deleting Templates

You must have the **Schedule Key Deletion** permission to delete a template.

To delete a template:

-
1. Open the **Cloud Key Manager** application.
 2. In the left pane, click **Cloud Keys > AWS**. By default, the Keys tab is selected.
 3. Click the **Templates** tab. The list of available templates is displayed.
 4. Click **Delete Template** corresponding to the desired template. The Delete Template dialog box is displayed.

Deleting a template is permanent and cannot be undone. Keys using this template will not be affected. Are you sure you want to delete this template?

5. Click **Delete Template**.

A success message is displayed on the screen.

Migrating to IAM Roles Anywhere Connections

To migrate from an access key and secrets-based connection to an IAM Roles Anywhere-based connection:

1. Upgrade to the CipherTrust Manager to the latest version. Refer to [System Upgrade/Downgrade](#) for details.
2. On the CipherTrust Manager, create an AWS connection with IAM Roles Anywhere enabled. Refer to the following sections for details:
 - a. [Prerequisites](#)
 - b. [Additional Configuration for AWS IAM Roles Anywhere](#)
 - c. [Connection Manager](#)
3. Open the **Cloud Key Manager** application.
4. In the left pane, click **KMS Containers > AWS KMS Accounts**.
5. Update the desired AWS KMS account with the AWS IAM Roles based connection (created above).
6. Verify that all the KMS operations are working as expected.

Downloading AWS Metadata

Downloading AWS Keys Metadata

CCKM users with permissions to view the AWS keys can export and download the keys metadata displayed on the CCKM GUI. Filter the keys metadata to display on the GUI based on your requirements, for example, all keys created in the past 12 months, all keys residing in a specific AWS account, and all keys based on a specific algorithm. You can then download the filtered metadata as a CSV file.

To download the AWS keys metadata visible:

1. Open the **Cloud Key Manager** application.
-

2. In the left pane, click **Cloud Keys > AWS**. The AWS Keys tab of the AWS Keys page is displayed.
3. Filter the keys metadata based on your requirements.

Use the Search box to filter the keys. To further filter the metadata, show/hide the desired columns and use the column-level filters.

4. Click **Download as CSV**.

The filtered AWS keys metadata is downloaded as a CSV file.

Downloading AWS KMS Accounts Metadata

CCKM users with permissions to view the KMS accounts can export and download the accounts metadata displayed on the CCKM GUI. Filter the KMS accounts metadata to display on the GUI based on your requirements. You can then download the filtered metadata as a CSV file.

Metadata of a maximum of 100 KMS accounts can be exported and downloaded.

To download the AWS KMS accounts metadata visible:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Filter the accounts metadata based on your requirements.

Use the Search box to filter the KMS accounts. To further filter the metadata, show/hide the desired columns and use the column-level filters.

4. Click **Download as CSV**.

The filtered AWS KMS accounts metadata is downloaded as a CSV file.

AWS CloudHSM Key Store Resources

Within AWS CloudHSMs, AWS provides cloud-based hardware security modules (HSMs), which are used to perform cryptographic operations and store cryptographic keys securely. Using an AWS CloudHSM, you can create and control your encryption keys within the AWS Cloud. The HSMs reside in a cluster of HSMs, which are automatically synchronized whenever an operation is performed on one. You own and manage these HSMs as a single-tenant user of the device, which you run on your own AWS Virtual Private Cloud (VPC).

In this documentation, AWS CloudHSM is also referred to as AWS CloudHSM key store.

CCKM integration with AWS CloudHSM enables you to perform the following from the CCKM UI:

- Create and manage CloudHSM key stores
- Connect to a CloudHSM key store
- Create and manage CloudHSM keys

Prerequisites for AWS CloudHSM

Prior to creating an AWS CloudHSM key store from CCKM, ensure to perform the prerequisite steps within AWS. For more information, refer to AWS CloudHSM documentation including the *AWS Key Management Service Developer Guide* and *AWS CloudHSM User Guide*.

Setting up a CloudHSM Key Store from CCKM

After you have performed the prerequisite steps including creating an AWS CloudHSM cluster, proceed to CCKM to create a CloudHSM key store and then connect it to the cluster you wish to use for this key store.

To set up a CloudHSM key store:

1. From CCKM, create a key store of type CloudHSM. See [Create a CloudHSM Key Store](#).
2. From CCKM, connect your CloudHSM key store to its associated AWS CloudHSM cluster. See [Connect to a CloudHSM Key Store](#).

Managing an AWS CloudHSM Key Store from CCKM

This section describes how to perform the following steps from CCKM:

- [Create a CloudHSM Key Store](#)
- [View and Edit Details of a CloudHSM Key Store](#)
- [Delete a CloudHSM Key Store](#)
- [Connect to a CloudHSM Key Store](#)
- [Disconnect a CloudHSM Key Store](#)

Create a CloudHSM Key Store

From CCKM, you have the ability to create an AWS CloudHSM key store within AWS Cloud. After the store is created, you can proceed to creating your CloudHSM keys within the key store from CCKM. For information, see [Creating CloudHSM Keys](#).

Note

A newly created key store will be in a disconnected state.

Note

You can only use an unused AWS CloudHSM cluster to create an AWS CloudHSM key store. For more

information, refer to AWS CloudHSM documentation.

Note

Ensure to download the trust anchor certificate of the AWS CloudHSM prior to creating a CloudHSM key store and save it to a secure location. For more information, refer to the *AWS CloudHSM User Guide*.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that will own the key store. The account details page is displayed.
4. Under **CloudHSM Key Stores**, click the **+ Add CloudHSM Key Store** button. You might have to expand the **CloudHSM Key Stores** section to view the button. The **Add CloudHSM Key Store** screen displays.
5. In the **Add CloudHSM Key Store** screen, provide the following settings:
6. Select the **Region** in which the store will reside.
7. Enter the **Key Store Name** using a friendly name for the key store.
8. Select the **AWS CloudHSM Cluster** from the drop-down list to which to associate with the key store.

Note

Only *unused* CloudHSM clusters display in the drop-down list.

9. Click **Choose File**, select the trust anchor certificate file of the AWS CloudHSM cluster. This certificate file was created during the initialization of the cluster. AWS KMS will use the trust anchor certificate to connect the key store to the cluster. For more information, refer to the *AWS CloudHSM User Guide*.
10. Enter the **Password** of the `kmsuser` crypto user (CU) account. When AWS KMS logs into the cluster, it uses this user to manage key material on your behalf.
11. Click **Save**

The new CloudHSM key store is displayed under **CloudHSM Key Stores**.

View and Edit Details of a CloudHSM Key Store

You have the option to edit the key store name, AWS CloudHSM cluster, and password of an existing CloudHSM key store.

Note

A CloudHSM key store must be in a disconnected state before you can edit its key store name, associated cluster, or password. **Be sure to disconnect the key store from the associated CloudHSM cluster before editing the key store.**

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **CloudHSM Key Stores**, find the desired CloudHSM key store name.
5. From the **CloudHSM Key Stores** display, click the ellipsis icon associated with the desired CloudHSM key store.
6. Click **View/Edit Details**. The **General Info** section of the view/edit details page displays.
7. (Optional) Enter the new key store name in **Key Store Name**, if you wish to use a different name.
8. (Optional) Select another **AWS CloudHSM Cluster** to which to associate with the key store from the drop-down list, if you wish to use a different cluster.
9. (Optional) Enter your new **Password** of the `kmsuser` CU account, if you wish to change this password.
10. Click **Update**.

Delete a CloudHSM Key Store

Before proceeding to delete a key store, be sure to delete all of the CloudHSM keys within it. You will have schedule to delete the keys. For more information, see [Schedule Deletion of a Key](#). Thereafter, disconnect the key store from its associated AWS CloudHSM cluster. You can then proceed to deleting the key store after disconnecting the key store.

Warning

Deleting a CloudHSM key store is a irreversible operation, which permanently deletes the store from the AWS Cloud.

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **CloudHSM Key Stores**, find the desired CloudHSM key store name.
5. From the **CloudHSM Key Stores** display, click the ellipsis icon associated with the desired CloudHSM key store.
6. Click **Delete Store**.
7. You are prompted to confirm the deletion. Enable the **I wish to delete this Key Store** checkbox and click **Delete**.

Connect to a CloudHSM Key Store

A newly created key store will be in a disconnected state. Connect the key store to the AWS CloudHSM cluster you wish to use for the store *before* creating keys within your key store.

Note

After a successful connection between a key store and a CloudHSM cluster, AWS KMS changes the password of the **kmsuser** CU account (associated with the key store) for security reasons. **Before** attempting to connect again, reset this password.

Note

Ensure that the **kmsuser** CU is not be logged into the key store's associated AWS CloudHSM cluster. Otherwise, the AWS KMS is prevented from logging into the account in which the store is added.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **CloudHSM Key Stores**, find the desired CloudHSM key store name.
5. From the **CloudHSM Key Stores** display, click the ellipsis icon associated with the desired CloudHSM key store.
6. Click **Connect**. The Connect Key Store page displays.
7. You are prompted to confirm that you wish to connect to the key store. Enter the password to use to connect to the key store and then click **Connect**. This is the password for the **kmsuser** CU account, which you

entered when you created the CloudHSM key store.

Disconnect a CloudHSM Key Store

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **CloudHSM Key Stores**, find the desired CloudHSM key store name.
5. From the **CloudHSM Key Stores** display, click the ellipsis icon associated with the desired CloudHSM key store.
6. Click **Disconnect**. The Disconnect Key Store page displays.
7. You are prompted to confirm that you wish to disconnect the key store. Click **Disconnect**.

AWS External Key Store Resources

External Key Store (XKS) resources for integration with Amazon Web Services Key Management Service (AWS KMS) allow you to manage keys held in Luna Network HSMs (Hardware Security Module) or a CipherTrust Manager, and allows AWS KMS to use the keys for cryptographic operations on demand.

The external custom key store entity on CipherTrust Cloud Key Manager (CCKM) provides access to AWS KMS to use source key material stored in HSM partitions or a CipherTrust Manager, while preserving end user control to manage those source keys outside of AWS KMS.

The external custom key store contains Hold Your Own Key (HYOK) keys, which acts as an intermediary to the source key material stored in Luna HSM or CipherTrust Manager. Luna or CipherTrust Manager (depending on which key source you are using) executes the cryptographic operations.

External custom key stores and HYOK keys on CCKM can be created as linked or unlinked to an external key store in AWS KMS. Linked key stores and keys automatically synchronize with objects in AWS KMS, and unlinked key stores do not. Unlinked key stores and HYOK keys require you to either link those objects, which automatically creates corresponding external key stores and KMS keys on AWS KMS, or to manually create corresponding objects in AWS KMS.

Required Resources

- If using a Luna HSM as a key source, then ensure to have at least one [Thales Luna Network HSM \(v7.4 or higher\)](#), with at least one initialized network trust link (NTLS) or Secure Trusted Channel (STC) partition. Preferably, you should have two HSMs for redundancy. When you connect multiple Luna Network HSM partitions to CipherTrust Manager, the partitions automatically become HA (High Availability) enabled and replicate keys to each other.

Warning

HSMs configured for Functionality Modules (FMs) are not compatible with this integration.

- An AWS Account.

Suggested Deployment Architectures

We recommend deploying at least two CipherTrust Managers in [cluster](#) to provide proper redundancy and availability to AWS KMS. All XKS resources and configurations are replicated across all nodes. If using Luna HSM as a key source, then we recommend connecting two or more Luna Network HSM partitions for redundancy. When you connect multiple partitions, they automatically become HA enabled and replicate keys to each other.

If using Luna HSM as a key source, you must establish connections with the CipherTrust Manager instances and the AWS account that manages corresponding KMS keys and external key stores, as described in [Managing External Key Store Resources](#).

If using CipherTrust Manager as a key source, you must establish connections between the CipherTrust Manager instances and the AWS account.

Deployment topologies vary based on usage of AWS Virtual Private Cloud (VPC). The following are the supported deployment topologies:

- Host the CipherTrust Manager instances on a VPC
- Host the CipherTrust Manager instances on premises with a VPC connection
- Host the CipherTrust Manager instances on premises and make them public routable and available over the Internet

In these topologies, the key source is either Luna HSM or CipherTrust Manager. When using Luna HSMs, these are deployed on premises.

Deployment Prerequisites

- The CipherTrust Manager must be reachable through a Fully Qualified Domain Name (FQDN).
 - In VPC deployments, you provide this value to set the **Private DNS Name** in the VPC endpoint service. If you are operating an unlinked key store and manually creating an external key store on AWS KMS, you also provide this value as the **Proxy URI endpoint**.
 - You provide this as the **Proxy URI endpoint** for public routable deployments without a load balancer.
- You can change the default port of 443 for the CipherTrust Manager web interface. One would have to ensure to change it **BEFORE** configuring the given cloud service on CCKM. However, in the case of AWS XKS, AWS KMS only accepts and sends requests on the default port of 443. If you wish to use a different port for the web interface other than the default, add a network component, such as a load balancer or a firewall, in front of CipherTrust Manager. Then configure port mapping on this component by mapping port 443 to the non-default

port on the CipherTrust Manager you plan to use. For more information, refer to [Support for Changing the Default Port of Web Interface Setting](#).

- The web interface must have a [TLS certificate signed by an external Certificate Authority \(CA\)](#) trusted by AWS. AWS trusts certificates issued by well-known public CAs such as Verisign.
- The Common Name (CN) in the web interface certificate must be set to a specific value.
 - A good default value for the CN is the CipherTrust Manager FQDN.
 - For VPC deployments with more than one AWS KMS external key store, the CN should include the wildcard * asterisk character.
 - If you are using a load balancer with a publicly routable deployment, and the load balancer is set to do SSL passthrough, include the wildcard * asterisk character in the CN.
- We recommend setting up all networking elements before creating the external custom key store on CCKM. This is especially important when you create an external custom key store in the linked state. Successful validation of the connection between AWS KMS and the new external custom key store must take place before a linked key store can be created. For example, for [CipherTrust Manager hosted on premise with VPC connection](#), configure VPC endpoints as described in AWS documentation, before creating the external custom key store on CCKM.
- The Health Check Key must be created before creating a key store.

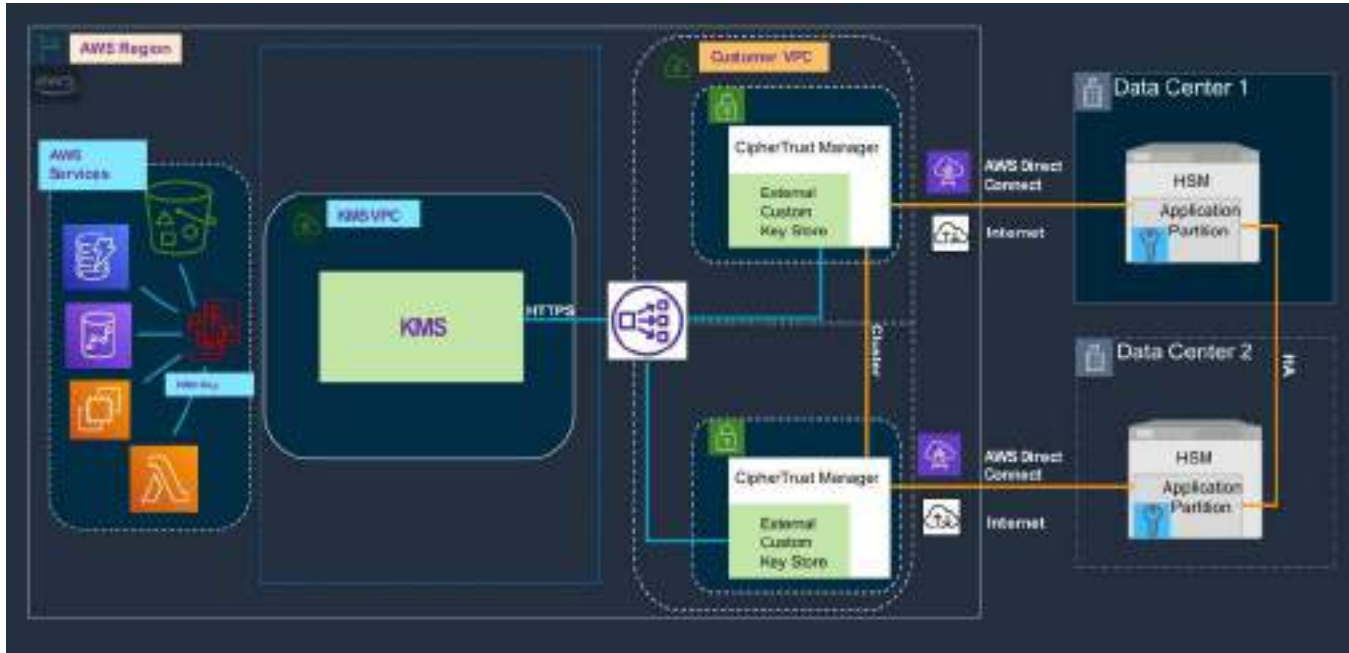
Depending on the location of the key store, the health check is initiated.

- If the key store is on Luna HSM, the Health Check Key must have the following attributes:
 - `CKA_EXTRACTABLE = FALSE`
 - `CKA_SENSITIVE = TRUE`
 - `CKA_ENCRYPT = TRUE`
 - `CKA_DECRYPT = TRUE`
 - `CKA_WRAP = TRUE`
 - `CKA_UNWRAP = TRUE`
- If the key store is on CipherTrust Manager, the Health Check Key must have the following attributes:
 - `Key Not Exportable`
 - `Key Not Deletable`
 - `Usage Masks - Encrypt, Decrypt, Wrap, Unwrap`

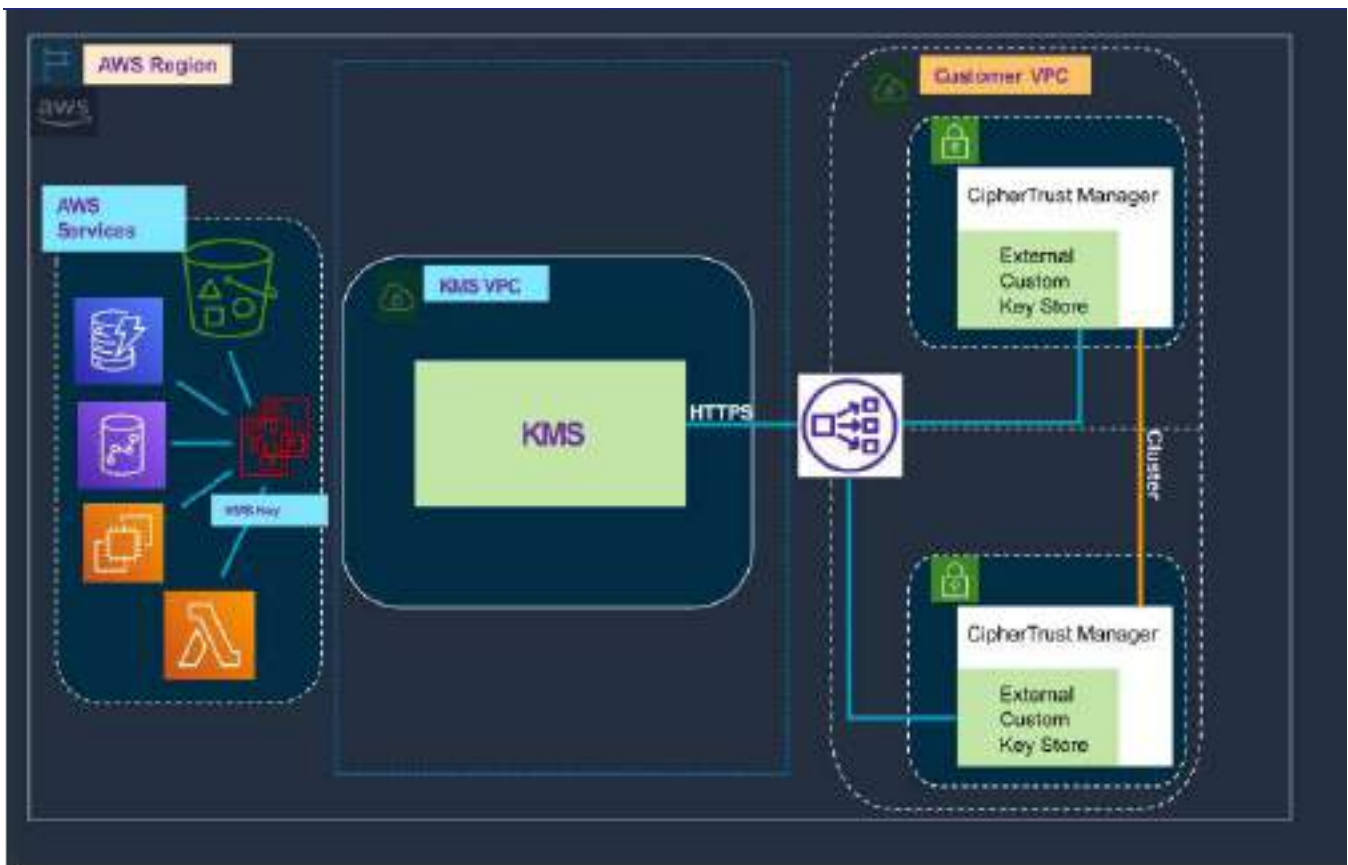
CipherTrust Manager Hosted on VPC

CipherTrust Manager deployment inside a VPC is described in the [Amazon Web Services deployment page](#). Detailed VPC configuration, such as availability zones, is described in [AWS Key Management Service Deployment Guide](#).

CipherTrust Manager Hosted on VPC Using Luna HSM as a Key Source



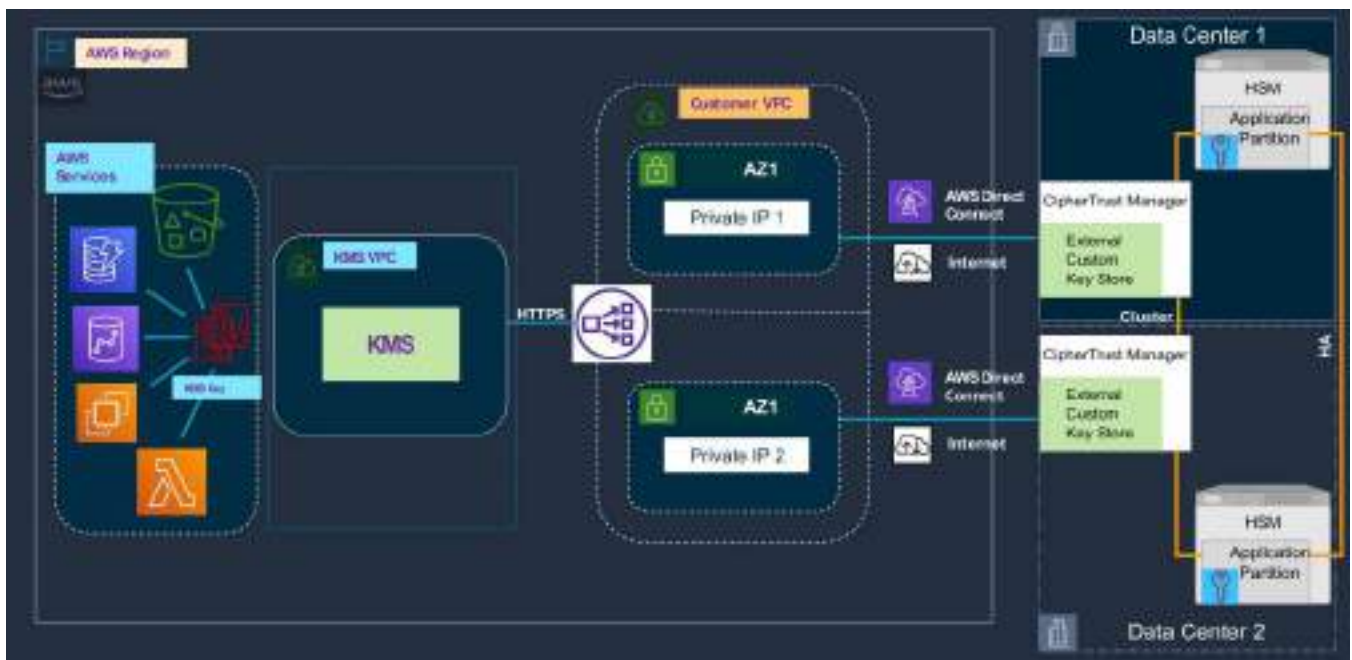
CipherTrust Manager Hosted on VPC Using CipherTrust Manager as a Key Source



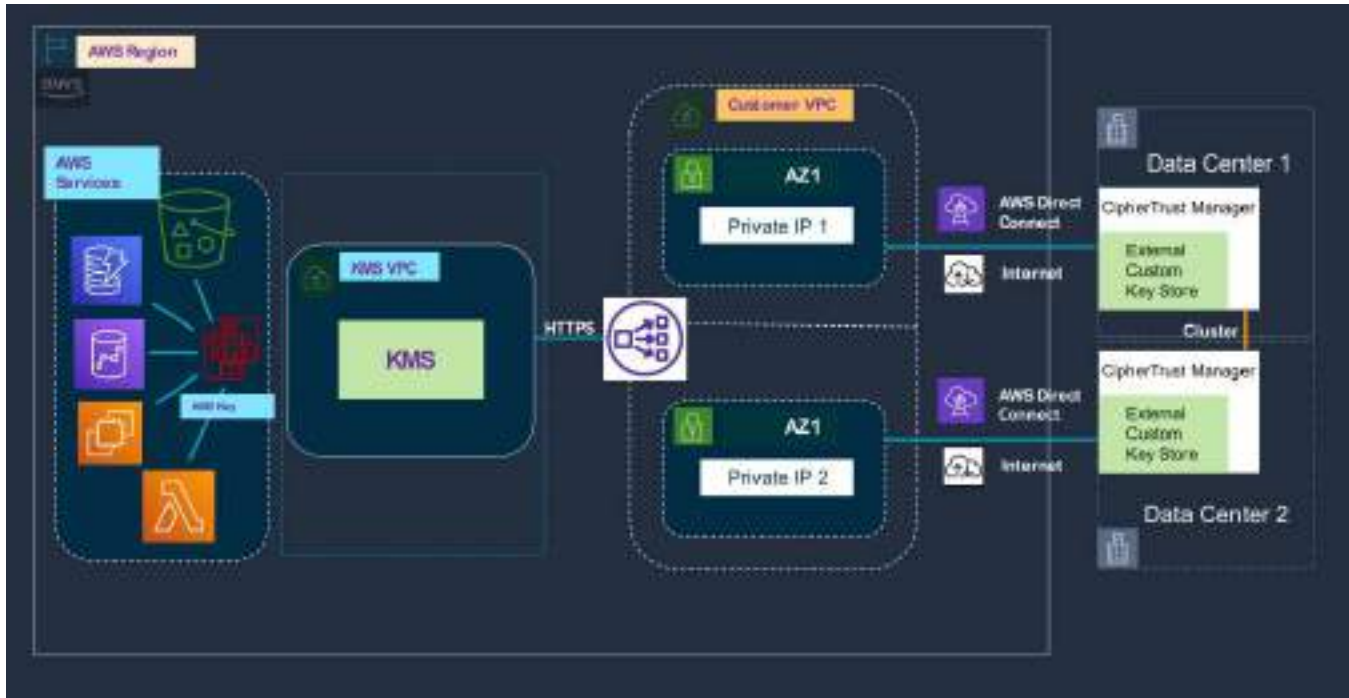
CipherTrust Manager Hosted on Premise with VPC Connection

Configuration of VPC endpoints is described in [AWS Key Management Service Deployment Guide](#).

CipherTrust Manager Hosted on Premise Using Luna HSM as a Key Source

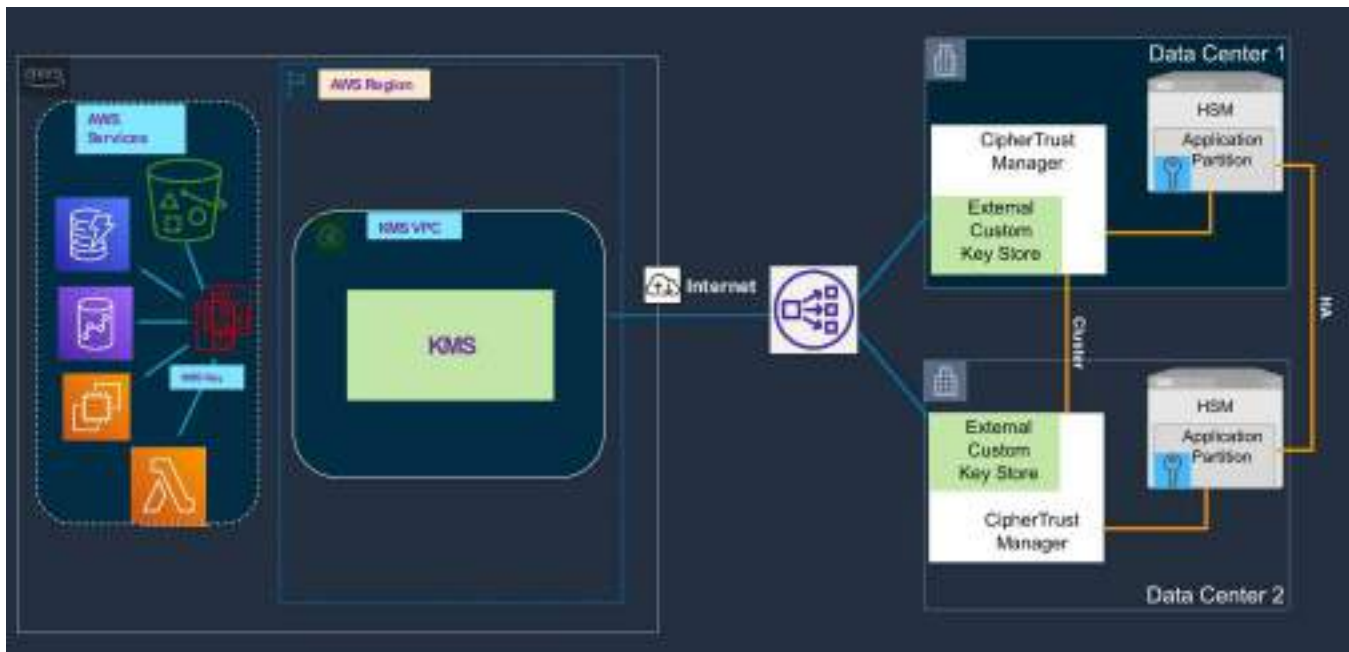


CipherTrust Manager Hosted on Premise Using CipherTrust Manager as a Key Source

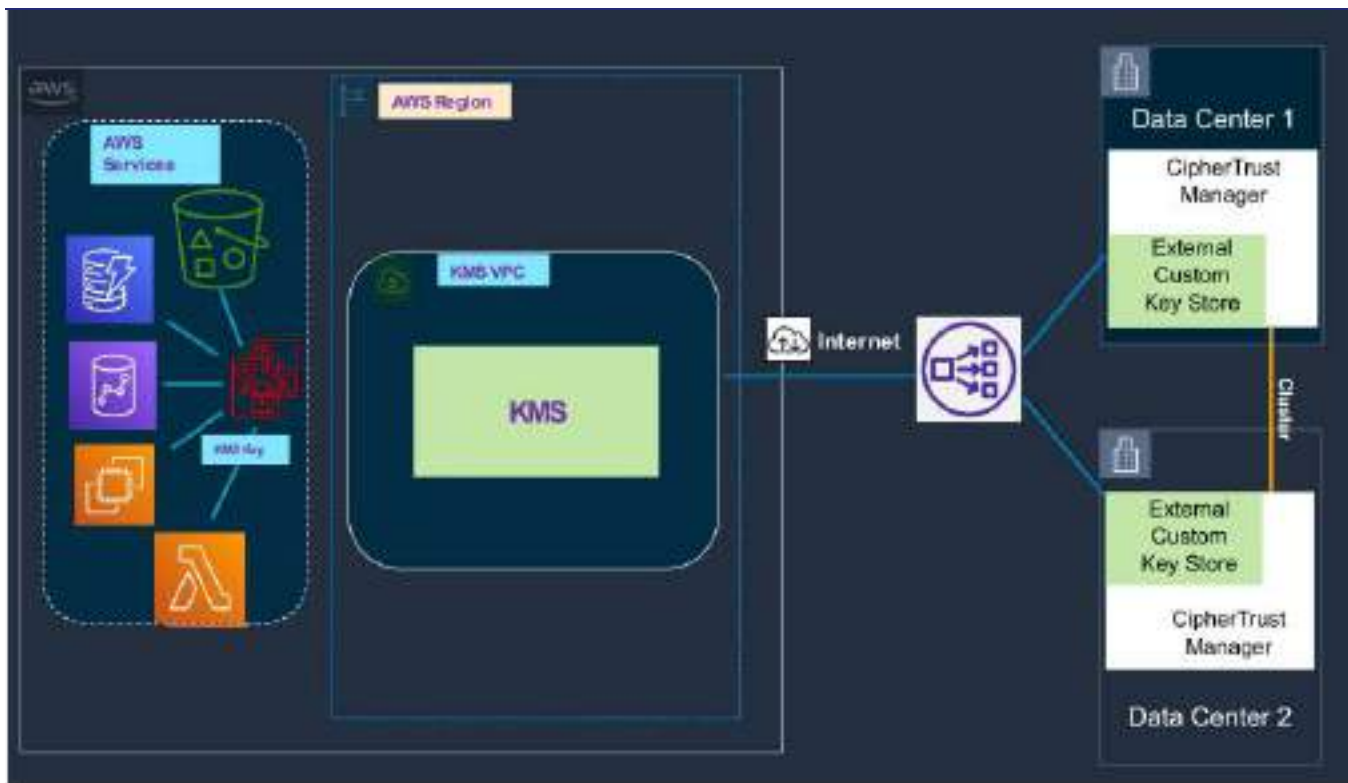


CipherTrust Manager on Premise with Public Routable Connection

CipherTrust Manager on Premise with Public Routable Connection Using Luna HSM as a Key Source



CipherTrust Manager on Premise with Public Routable Connection Using CipherTrust Manager as a Key Source



Network Considerations for Optimal Stability and Performance

There are some important network considerations for deployments of an external key store.

- AWS KMS, Luna HSM appliances (if using Luna HSMs as a key source), and CipherTrust Manager instances should all be as geographically close together as possible. The supported AWS regions are described in [AWS Key Management Service Deployment Guide](#).
- When Luna HSM is the key source, we recommend a network latency of round-trip communication of 25 ms or less between CipherTrust Manager and the HSM.
- When CipherTrust Manager is the key source, we recommend a network latency of round-trip communication of 35 ms or less between AWS KMS and the CipherTrust Manager.
- We recommend preparing only one or two external key stores per deployment.
- For a summary of AWS XKS performance, refer to [AWS XKS Performance Summary](#).

High Level Integration Process Using Luna HSM as a Key Source

These are the high level steps to allow AWS KMS to begin making cryptographic requests to source keys through external custom key stores within CCKM. Unlinked external custom key stores and HYOK keys require additional steps.

1. Deploy two CipherTrust Manager instances in keeping with the [deployment prerequisites](#). Cluster the instances together.

-
2. Deploy one or more Luna Network HSM partitions with initialized NTLS or STC partitions, as described in [Luna Network HSM documentation](#). Preferably, you would have two HSM servers with at least one partition each.
 3. If needed for a VPC connection, configure VPC endpoints as described in the [AWS Key Management Service Developer Guide](#).
 4. Prepare an External Custom Key Store.

As part of this process, you establish CipherTrust Manager connections to Luna HSM partitions and to the AWS account, and create an external custom key store on CCKM.

5. If you created an unlinked key store on CCKM, you must either link the key store or create a corresponding external key store on AWS KMS as described in [AWS Key Management Service Deployment Guide](#).
6. Create an AWS HYOK key.
 - In the CipherTrust Manager web console for [AWS keys](#), you create a Hold Your Own Key (HYOK) key. An associated backing Luna HSM source key is implicitly created.
 - In the CipherTrust Manager [REST API](#), you create a Luna HSM source key, provide its value to create a virtual key, and then create an HYOK key associated with the virtual key.
7. If you have created an unlinked HYOK key on CCKM, either link the key or create a KMS key in the AWS external key store, associated with the the HYOK key on CCKM.

High Level Integration Process Using CipherTrust Manager as a Key Source

These are the high level steps to allow AWS KMS to begin making cryptographic requests to CipherTrust Manager source keys through external custom key stores within CCKM. Unlinked external custom key stores and HYOK keys require additional steps.

1. Deploy two CipherTrust Manager instances in keeping with the [deployment prerequisites](#). Cluster the [instances](#) together.
2. If needed for a VPC connection, configure VPC endpoints as described in the [AWS Key Management Service Developer Guide](#).
3. Prepare an External Custom Key Store.

As part of this process, you establish a CipherTrust Manager connection to the AWS account and create an external custom key store on CCKM.

4. If you created an unlinked key store on CCKM, you must either link the key store or create a corresponding external key store on AWS KMS, as described in [AWS Key Management Service Deployment Guide](#).
5. Create an AWS External Custom Key Store on CCKM.

6. If you have created an unlinked HYOK key on CCKM, either link the key or create a KMS key in the AWS external key store, associated with the the HYOK key on CCKM.

AWS and CCKM Terminology

Terms used in AWS documentation, API, or AWS Console can differ from terms used in CCKM documentation, CipherTrust Manager REST API, and CipherTrust Manager web console for the same concepts. The following table provides equivalencies.

AWS term(s)	CCKM term(s)	Notes
External key	Luna HSM source key or CipherTrust Manager source key	Generally refers to the key material which performs cryptographic operations.
External Key ID, XksKeyId	XKS ID, XksKeyConfiguration:Id, HYOK ID	This is the key identifier which AWS KMS specifies in cryptographic requests. Important Note: This XKS ID is associated with the HYOK. It is not the source CipherTrust Manager or Luna HSM key's ID.
External Key Store	External Custom Key Store, Custom Key Store	CCKM documentation refers to the CCKM object as external custom key store and the AWS object as external key store . Product interfaces may use these terms interchangeably.
Proxy URI endpoint	XKS Proxy URI Endpoint	This value is the hostname of the server entity which AWS KMS first connects to for cryptographic requests. In VPC deployments, this is the Private DNS Name set for the VPC endpoint service. In non-VPC deployments, this is usually a load balancer.
Proxy URI Path	XKS Proxy URI Path	This is the path associated with an external custom key store in CCKM.
XKS proxy configuration file	Credentials file	The downloadable file available in CCKM when you create a new external custom key store or rotate the credentials for an external custom key store. This file contains the external custom key store's Proxy URI path prefix , Access key ID , and Secret access key values.

Managing External Key Store Resources

Deployment Prerequisites

- The CipherTrust Manager must be reachable through a Fully Qualified Domain Name (FQDN).
 - In VPC deployments, you provide this value to set the **Private DNS Name** in the VPC endpoint service. If you are operating an unlinked key store and manually creating an external key store on AWS KMS, you also provide this value as the **Proxy URI endpoint**.
 - You provide this as the **Proxy URI endpoint** for public routable deployments without a load balancer.
- You can change the default port of 443 for the CipherTrust Manager web interface. One would have to ensure to change it **BEFORE** configuring the given cloud service on CCKM. However, in the case of AWS XKS, AWS KMS only accepts and sends requests on the default port of 443. If you wish to use a different port for the web interface other than the default, add a network component, such as a load balancer or a firewall, in front of CipherTrust Manager. Then configure port mapping on this component by mapping port 443 to the non-default port on the CipherTrust Manager you plan to use. For more information, refer to [Support for Changing the Default Port of Web Interface Setting](#).
- The web interface must have a **TLS certificate signed by an external Certificate Authority (CA)** trusted by AWS. AWS trusts certificates issued by well-known public CAs such as Verisign.
- The Common Name (CN) in the web interface certificate must be set to a specific value.
 - A good default value for the CN is the CipherTrust Manager FQDN.
 - For VPC deployments with more than one AWS KMS external key store, the CN should include the wildcard ***** asterisk character.
 - If you are using a load balancer with a publicly routable deployment, and the load balancer is set to do SSL passthrough, include the wildcard ***** asterisk character in the CN.
- We recommend setting up all networking elements before creating the external custom key store on CCKM. This is especially important when you create an external custom key store in the linked state. Successful validation of the connection between AWS KMS and the new external custom key store must take place before a linked key store can be created. For example, for [CipherTrust Manager hosted on premise with VPC connection](#), configure VPC endpoints as described in AWS documentation, before creating the external custom key store on CCKM.

- The Health Check Key must be created before creating a key store.

Depending on the location of the key store, the health check is initiated.

- If the key store is on Luna HSM, the Health Check Key must have the following attributes:
 - **CKA_EXTRACTABLE = FALSE**

-
- `CKA_SENSITIVE = TRUE`
 - `CKA_ENCRYPT = TRUE`
 - `CKA_DECRYPT = TRUE`
 - `CKA_WRAP = TRUE`
 - `CKA_UNWRAP = TRUE`
 - If the key store is on CipherTrust Manager, the Health Check Key must have the following attributes:
 - `Key Not Exportable`
 - `Key Not Deletable`
 - Usage Masks - Encrypt, Decrypt, Wrap, Unwrap

Permissions to Manage External Custom Key Stores

By default, users in the [CCKM Admins group](#) can manage external custom key stores. This group includes the root `admin` account on CipherTrust Manager.

CCKM administrators can grant users assigned to the [CCKM users group](#) permission to manage external custom key stores. CCKM manages these permissions through the [Access Control List](#) associated with the [AWS KMS account](#).

Linked and Unlinked States

External custom key stores are created in either a **Linked** or **Unlinked** state.

When you create a linked key store on CCKM, a corresponding external key store is automatically created on AWS KMS.

After you create an unlinked key store on CCKM, you must either link it to automatically create a corresponding key store on AWS KMS, or you must manually create the key store on AWS KMS.

If you manually create an external key store on AWS KMS, you can later detect and link it to a CCKM local external custom key store by [refreshing external custom key stores](#), or [refreshing AWS keys](#).

While an external custom key store is unlinked, some operations to later update an unlinked external custom key store require manual configuration on AWS. These update operations include:

- [Creating Keys](#)
- [Rotating Credentials](#)

Prepare an External Custom Key Store

Before you create an external custom key store, decide whether it will be [linked](#) or [unlinked](#).

To prepare an external custom key store and provide access to AWS KMS, you follow the following stages in

order:

1. If using CipherTrust Manager as a key source, [establish an AWS Connection](#). If using Luna HSM as a key source, [establish both AWS and Luna HSM Connections](#).
2. [Create an External Custom Key Store on CCKM](#).
3. Set up the corresponding AWS KMS external key store to communicate with the new CCKM external custom key store. Steps depend on whether the external custom key store is linked or unlinked.
 - **Linked key stores:** [Connect the new external key store](#).
 - **Unlinked key stores:** [Create the Corresponding Key Store on AWS KMS](#).
4. [Create Keys for the External Custom Key Store](#).

The configuration and new objects you create on CCKM, connections to AWS and Luna HSM or CipherTrust Manager, AWS account information, the external custom key store, and AWS keys are replicated across a cluster. For multiple CipherTrust Managers in [cluster](#), you need to only perform this configuration on one node.

Establishing an AWS Connection

If you are using CipherTrust Manager as a key source, be sure to establish a connection between AWS and CipherTrust Manager for HYOK integrations.

1. [Establish an AWS connection](#).

Note

If your CipherTrust Manager is deployed inside a VPC, and the AWS account is in a region other than `us-east-1`, there is an AWS STS Regional Endpoints setting required.

2. Test the connection by finding the new **AWS** connection in the **Connections Table**, clicking the ellipsis button in the row, and selecting **Test Connection**.

A **Ready** status indicates that the connection is working correctly.

3. Add an AWS account, as described in [Managing AWS Accounts](#).

Caution

When you select regions, make sure they are compatible for external key stores. Compatible regions are described in [AWS Key Management Service Deployment Guide](#).

Tip

CCKM Admins can allow CCKM users to manage external custom key stores through the Access Control List associated with the AWS account.

Establishing AWS and Luna HSM Connections

If you are using Luna HSM as a key source, be sure to establish AWS and Luna HSM connections for HYOK integrations.

The [Connection Manager](#) page gives more details about managing the AWS and Luna HSM connections.

1. Establish an AWS connection.

Note

If your CipherTrust Manager is deployed inside a VPC, and the AWS account is in a region other than `us-east-1`, there is an AWS STS Regional Endpoints setting required.

2. Test the connection by finding the new **AWS** connection in the **Connections Table**, clicking the ellipsis button in the row, and selecting **Test Connection**.

A **Ready** status indicates that the connection is working correctly.

3. Add a HSM Server connection and a Luna Network HSM Connection to establish access to a Luna Network HSM partition.
4. Add another HSM server connection and Luna Network HSM connection to add a partition from a second Luna Network HSM.
5. Test the HSM connections in the **Connections Table**, clicking the ellipsis button in the row, and selecting **Test Connection**.

A **Ready** status indicates that the connection is working correctly.

6. Add an AWS account, as described in [Managing AWS Accounts](#).

Caution

When you select regions, make sure they are compatible for external key stores. Compatible regions

are described in [AWS Key Management Service Deployment Guide](#).

Tip

CCKM Admins can allow CCKM users to manage external custom key stores through the Access Control List associated with the AWS account.

Creating an External Custom Key Store on CCKM

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that will own the key store. The account details page is displayed.
4. Under **External Custom Key Stores**, click the **+ Add External Key Store** button. You might have to expand the **External Custom Key Stores** section to view the button.
5. Under **Config and Add Key Store**, provide the following settings:
 - **Key Store Name**. A friendly name for the Key Store.
 - **Region**. Compatible regions are listed in [AWS Key Management Service Deployment Guide](#).
 - **Enable success audit events**. Use this toggle to enable or disable audit recording of successful operations within an external key store. The toggle is turned off by default.

Note

This toggle displays only if the key store is of type "local".

Note

Enabling the **Enable success audit events** toggle significantly impacts the performance of the key store. Enable this feature only for the purpose of troubleshooting or presenting a demo.

- **Enable Mutual TLS.** Use this toggle to enable or disable TLS client-side certificate verification where the CipherTrust Manager authenticates the AWS KMS client. The toggle is turned off by default.

Note

If you plan to enable mutual TLS on an AWS external custom key store, and there is a load balancer in front of CCKM, then configure the load balancer so that SSL termination does not take place on it thereby allowing SSL/TLS Passthrough. This configuration is required for CCKM and AWS KMS to successfully exchange certificates.

Note

If enabling TLS client-side certificate verification, ensure to perform the following prerequisite steps:

- a. Create a CA signed certificate for one of the AWS Trusted CAs for the Web interface of your CipherTrust Manager.
- b. Download and add the AWS root CA certificate from the same AWS trusted CA to External CA on CipherTrust Manager. For more information, refer to [Creating an External CA](#).
- c. Add the CA signed certificate to the CipherTrust Manager web interface. For more information, refer [Using an externally generated server certificate for an interface](#).

- **Source.** Either CipherTrust Manager or Luna HSM.

Caution

The source you select at this time permanently affects the Hold Your Own Key (HYOK) and health check keys you can add to the external custom key store later. If you select Luna HSM, only Luna HSM keys can be used for health check, and as source keys to HYOK keys. If you select CipherTrust Manager, only CipherTrust Manager keys can be used for health check, and as source keys to HYOK keys. You cannot mix Luna HSM and CipherTrust Manager source keys or health check keys within the same external custom key store.

- **Key Store Container.** If using Luna HSM, select the [desired partition](#) that will store the source key(s) for the key store.
- **Link State.** Select **Unlinked** to create a key in the AWS External Key Store that is in an unlinked state. This means that the corresponding key store is not automatically created in AWS KMS, and you must

create it. Select **Linked** to create an AWS External Key Store that is linked to the CCKM external custom key store. If you select **Linked**, and there is at least one credential rotation schedule already added within **Schedules**, then you have the option to assign any of the available credential rotation schedules to this key store. Proceed to Step 6 and then Step 7 to assign a credential rotation schedule to this key store *after* entering all of the settings in this step. Otherwise, skip to Step 8 *after* entering all of the settings in this step. See [Linked and Unlinked States](#) for more information on the effects of these choices. See [Adding Credential Rotation Schedule \(AWS XKS only\)](#) for more information about adding a credential rotation schedule for an AWS External Key Store.

- **XKS Location.** Select **Amazon VPC**, if you will use an Amazon VPC to connect to the key store. Then enter the **VPC Endpoint Service Name** and the **XKS Proxy URI Endpoint**. Select **Public**, if you will use the Internet to connect to the key store. Enter the **XKS Proxy URI Endpoint** thereafter.

Note

VPC Endpoint Service Name can be obtained from AWS configuration.

XKS Proxy URI Endpoint refers to the hostname of the server entity which AWS KMS first connects to for cryptographic requests.

- For **Amazon VPC** deployments, this is the Private DNS Name set for the VPC endpoint service.
 - For Public deployments, this is usually the load balancer host name.
 - If AWS is connecting to CipherTrust Manager directly, this is the CipherTrust Manager fully qualified domain name.
- **Health Check Key.** From the drop-down list, select the **Key ID** of the symmetric AES-256 key to use for performing the health checks on the given key store. AWS initiates this health check on a given key store to check whether the store is up and running and can perform cryptographic operations. When CCKM receives this health check request from AWS, it performs the health check using the Key ID.

Note

The Health Check Key must be created before creating a key store.

Depending on the location of the key store, the health check is initiated.

- If the key store is on Luna HSM, the Health Check Key must have the following attributes:
 - **CKA_EXTRACTABLE = FALSE**

-
- `CKA_SENSITIVE = TRUE`
 - `CKA_ENCRYPT = TRUE`
 - `CKA_DECRYPT = TRUE`
 - `CKA_WRAP = TRUE`
 - `CKA_UNWRAP = TRUE`
 - If the key store is on CipherTrust Manager, the Health Check Key must have the following attributes:
 - `Key Not Exportable`
 - `Key Not Deletable`
 - Usage Masks - `Encrypt, Decrypt, Wrap, Unwrap`
 - **Max Credentials.** Enter the maximum number of valid credentials allowed at any given time. These credentials are used to authenticate KMS accessing the external custom key store. The valid values are from 2 to 20 (inclusive). Once a key store reaches the maximum number of credentials, the oldest credential is deleted. For more information about the use of the AWS credentials to access external key stores, refer to [AWS Key Management Service Deployment Guide](#).

6. Click **Next**.
7. (Optional) Under **Add Credential Rotation Schedule** and from the drop-down list, select the credential rotation schedule to assign to this key store.
8. Click **Add Key Store** to continue.

For linked key stores, CCKM tests that the AWS KMS account can reach CCKM with the provided connection and network parameters. If this connection test fails, you cannot create the new key store.

A summary of the settings is displayed on the **Retrieve Credentials** page.

- If desired, you can copy the **Health Check URI Path** value (using the copy button) at this time to check the health of a given CipherTrust Manager node. For each of your CipherTrust Manager nodes, create a routable endpoint URL by adding "https://(ip address or hostname of node)" as a prefix to this URI path. You can then use this end point to validate whether a custom key store is ready to handle requests. Load balancers can also be configured to call this endpoint for a health check. To obtain the status of a CipherTrust Manager node that is behind a load balancer, configure the node's health check URL in the load balancer. You can also retrieve the **Health Check URI Path** value later from the details page of the external custom key store.

Note

The **Health Check URI Path** value is only available for a custom key store that is a local key store (meaning the store is locally hosted on the given CCKM instance).

Note

If you added a credential rotation schedule for the new key store, the job config ID associated with the schedule displays under CREDENTIALS.

9. Click **Download** to download the credentials file required by AWS KMS.

Caution

This credentials file includes a secret access key string which AWS KMS requires, and you cannot access again. You must download the file, or copy the **secret access key** string before closing the **Add External Key Store** wizard, or you will need to [rotate credentials](#) to generate new credentials for AWS.

10. Click **Close**.

The new external custom key store is displayed under **External Custom Key Stores**.

11. Set up the corresponding AWS KMS external key store to communicate with the new CCKM external custom key store. Steps depend on whether the external custom key store is linked or unlinked.
 - **Linked key stores:** [connect](#) the new external key store.
 - **Unlinked key stores** [create the Corresponding Key Store on AWS KMS](#).

Creating the Corresponding External Key Store on AWS KMS for Unlinked Key Stores

If your new external customer key store is unlinked, this means that you must either [link the external custom key store](#) to create the corresponding external key store on AWS, or [manually create](#) the key store on AWS KMS through its console, API, or CLI.

Link an Unlinked External Custom Key Store

If you created an unlinked external custom key store, you can link it after creation. Linking an external custom key store automatically creates a corresponding external key store in AWS KMS.

Note

You cannot use this operation to link to an existing external custom key store on AWS KMS. To link an existing external custom key store, [refresh all external key stores for an AWS account](#).

If you want to set up [CipherTrust Manager hosted on premise with VPC connection](#), configure VPC endpoints as described in [AWS Key Management Service Deployment Guide](#) if you have not already. VPC endpoints are associated with an external custom key store.

To link an unlinked external custom key store:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **External Custom Key Stores**, find the desired external custom key store name.
5. From the **External Key Stores** display, click the ellipsis icon associated with the desired external custom key store.
6. Select **Link**.

The Link Key Store dialog displays.

7. Review the auto-populated networking details.

These include the **XKS Location** selected during [key store creation](#).

8. Edit networking details if needed.

For **Public** XKS Location, you can edit the **XKS URI Proxy Endpoint**.

For **Amazon VPC** XKS Location, you can edit **VPC Endpoint Service Name** and the **XKS Proxy URI Endpoint**

9. Click **Link** to confirm.

An external key store is created on AWS KMS.

10. Proceed to [connect](#) the newly linked key store.

Manually Create Key Store on AWS KMS

If you want to set up [CipherTrust Manager hosted on premise with VPC connection](#), configure VPC endpoints as described in [AWS Key Management Service Deployment Guide](#) if you have not already. VPC endpoints are associated with an external custom key store.

The following CipherTrust Manager and CCKM values are required for creating the external key store on AWS KMS:

- The CipherTrust Manager **FQDN**, in most deployments.

For VPC deployments, the CipherTrust Manager FQDN is provided for the **Proxy URI Endpoint** when you create the External Key Store on AWS KMS.

If you have a non-VPC deployment, and are not operating behind a load balancer, this is the value needed for **Proxy URI Endpoint** in AWS console.

- The load balancer hostname, if applicable for non-VPC deployments. This is used as the **Proxy URI Endpoint**.
- The credentials file downloaded during [External Custom Key Store Creation on CCKM](#) or when [rotating credentials](#). This file contains **Proxy URI path prefix**, **Access key ID**, and **Secret access key** values, which you can directly upload to AWS console.

Note

There is an additional field in the AWS console to create the external key store, **VPC endpoint service name**, for VPC deployments. This is the VPC endpoint service which you have associated with the CipherTrust Manager and custom external key store.

If you do not wish to provide the credentials configuration file, you can manually input the following values into AWS Console:

- The **Secret access key**.

This value is only available on first [creating the external custom key store creation on CCKM](#), or when you [rotate credentials](#) for an external custom key store. There are options during those operations to copy the secret access key string.

- **The XKS Proxy URI Path.** This value is available through the **External Custom Key Stores** details:
 - a. Open the **Cloud Key Manager** application.
 - b. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
 - c. Click the name of the account that owns the external custom key stores.
 - d. Under **External Custom Key Stores**, find the desired external custom key store name.
 - e. For the **XKS Proxy URI Path** for the external custom key store, click the copy button to copy the path.

The full path is copied. The AWS console already includes the **/kms/xks/v1** suffix, so you must delete

that part of the path when pasting.

- The **Access key ID**

- a. From the **External Key Stores** display, click the ellipsis icon associated with the desired external custom key store.
- b. Select **View/Edit Details**.
- c. Click the copy button next to the desired **Access Key ID** to copy this value.

After creating the external key store on KMS, you must connect it on KMS to begin sending cryptographic requests to CCKM.

Note

If you later [refresh all external custom key stores for an AWS account](#) on CCKM, this external key store on KMS will be detected, and linked to the corresponding external custom key store on CCKM.

Connect or Disconnect an External Custom Key Store

You can connect or disconnect AWS KMS external key stores. Linked external key stores are created in the **Disconnected** state, and must be connected either through CCKM or AWS console to begin sending requests to the CCKM external custom key store.

As a convenience, you can connect or disconnect linked external custom key stores on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **External Custom Key Stores**, find the desired external custom key store name.
5. From the **External Key Stores** display, click the ellipsis icon associated with the desired external custom key store.
6. Select **Connect** to connect or **Disconnect** to disconnect.

Connecting or disconnecting an external custom key store through CCKM results in the corresponding external key store on AWS changing state to **connected** or **disconnected**. The transient states of **connecting** and **disconnecting** display during the transition. Consult [AWS Key Management Service Deployment Guide](#) for details on the allowed operations associated with these states.

Creating Keys for the External Custom Key Store

After configuring external custom key stores in both CCKM and AWS, you can create the keys necessary to perform cryptographic operations in response to AWS KMS. This is described in [Creating Hold Your Own Key](#)

(HYOK) Keys.

Block or Unblock an External Custom Key Store

Blocking and unblocking an external custom key store is a way to temporarily suspend and restore AWS KMS's access to the key store and its keys.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **External Custom Key Stores**, find the desired external custom key store name.
5. From the **External Key Stores** display, click the ellipsis icon associated with the desired external custom key store.
6. Click **Block** or **Unblock**, depending on whether the key store is in a blocked or unblocked state.
7. You are asked to confirm the action. Click **Block** or **Unblock** again.

The status in the **Blocked** column updates on the table.

Rotate Credentials

Periodically rotating credentials for the external custom key store is a security best practice. CCKM retains and recognizes the number of credentials set as the Max Credentials for the key store, which is between 2 and 20 credentials (default 20 credentials). Once a key store reaches the maximum number of credentials, the oldest credential is deleted. At this time, AWS only retains one set of credentials for each external key store.

Caution

When you create a new credential for an unlinked key store, you must immediately copy the **secret access key** string or download the credentials file containing this string. There will not be another opportunity to retrieve this value. You must manually provide the credentials in AWS console.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **External Custom Key Stores**, find and click the desired external custom key store name.

The details page of the external custom key store is displayed.

5. Check that the key store is **unblocked**. For linked key stores, also check that the state is either **connected** or

disconnected. If the state is in one of the transient states of **connecting** or **disconnecting**, wait until the transition is completed before proceeding to the next step.

6. Click **Add Credential** to add a new credential.
7. Click **Download** to download the file containing the new credentials, including the **Access key ID** and **Secret access key**.

Caution

This credentials file includes a **secret access key** string which you cannot access again. You must download the file, or copy the **secret access key** string before closing the **Add Credential** dialog, or you cannot add the credentials to AWS console.

A new credential is displayed.

8. If you have a linked key store, [connect the key store](#).
9. If you have an unlinked key store, edit the corresponding external key store on AWS console to upload the new credentials as described in [AWS Key Management Service Deployment Guide](#).
10. If desired, you can delete existing credentials with the **Delete Credential** button. CCKM will no longer accept requests using these older credentials.

Assign Credential Rotation Schedule

You have the ability to assign a credential rotation schedule for the credentials of a given external custom key store in CCKM. The store **must** be a local and linked key store. With a linked key store, the auto rotated (new) credentials are updated in AWS KMS.

To assign a credential rotation schedule to a store, you must first create one or more rotation schedules for the key store's credentials in the Add New Schedule wizard within **Schedules** of the **Cloud Key Manager** application. Thereafter, these schedules are made available within a list under **Schedules** of the details page for the given custom key store.

Note

A credential (new or existing) that is associated with the key store continues to be operational until either you delete it manually or it is automatically rotated out and then deleted based on the configured maximum number of valid credentials allowed at any given time. The oldest credential is deleted after the key store reaches the maximum number of credentials.

1. Open the **Cloud Key Manager** application credentials to be updated in AWS KMS.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.

3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **External Custom Key Stores**, find and click the desired external custom key store name.

The details page of the external custom key store is displayed.

5. Under Schedules, click **Credential Rotation Schedule** drop-down list and select a rotation schedule to assign to the key store.
6. Click **Update** to apply the schedule to the key store.

The selected rotation schedule is displayed showing when the next credentials rotation is to take place.

Unassign Credential Rotation Schedule

In CCKM, you also have the ability to unassign a credential rotation schedule for a given external custom key store.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **External Custom Key Stores**, find and click the desired external custom key store name.

The details page of the external custom key store is displayed.

5. Under Schedules, select the **X** icon (Unassign button) to unassign the selected rotation schedule for the key store.

Delete an External Custom Key Store

Warning

Deleting an external custom key store is a irreversible operation which permanently deletes unique identifiers required by the corresponding AWS KMS external key store.

You must [delete all HYOK keys in an external custom key store](#) before you can delete the external custom key store.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **External Custom Key Stores**, find the desired external custom key store name.

5. From the **External Key Stores** display, click the ellipsis icon associated with the desired external custom key store.
6. Click **Delete Store**.
7. You are asked to confirm the action. Enable the **I wish to delete this Key Store** checkbox and click **Delete**.

Update General Information for an External Custom Key Store

From the details page of an external custom key store, you can update general information for the key store.

You can update the following attributes of all external custom key store:

- Key Store Name
- XKS Location

You can update the following attributes for local external custom key stores only:

- Health Check Key
- Max Credentials
- Enable success audit events (by enabling or disabling this toggle depending on the current setting)
- Enable Mutual TLS (by enabling or disabling this toggle depending on the current setting)

You cannot edit health check key or max credentials for a remote type external custom key stores, which can appear after [refreshing external custom key stores for an AWS account](#).

Note

The **XKS Proxy URI Endpoint** value displays on the details page of an external custom key store. If desired, you can copy this value, which is uneditable, by clicking the copy button. If the custom key store is local, the **Health Check URI Path** value also displays on the details page of an external custom key store. If desired, you can copy this value, which is uneditable, by clicking the copy button.

The steps to update this information depends on whether the external custom key store is [linked](#) or [unlinked](#).

To update the key store

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Under **External Custom Key Stores**, find and click the desired external custom key store name.

The details page of the external custom key store is displayed.

5. Check that the state is **unblocked**. For linked key stores, also check that the state is **disconnected**.
6. In the General Info section, enter a new **Key Store Name**, if desired.
7. Change the **XKS Location** settings if desired to update networking details.

- Select **Amazon VPC**, if you will use an Amazon VPC to connect to the key store. Then enter the **VPC Endpoint Service Name** and the **XKS Proxy URI Endpoint**.

Caution

To avoid interruption in service, configure VPC endpoints in AWS **before** updating Amazon VPC details on an external custom key store in CCKM.

- Select **Public**, if you will use the Internet to connect to the key store. Enter the **XKS Proxy URI Endpoint**.

8. Select a new **Health Check Key**, if desired.

The health check key must be a symmetric AES-256 key.

The health check key must come from the key store's existing **Key Source Container**, CipherTrust Manager or Luna HSM connection.

9. Select a new number of **Max Credentials**, if desired. Valid values are between 2-20.
10. Change the setting of the **Enable success audit events** toggle to either enable or disable this feature, if desired. This toggle displays only if the key store is of type "local".
11. Change the setting of the **Enable Mutual TLS** toggle to either enable or disable this feature, if desired.

Note

If enabling TLS client-side certificate verification, ensure to perform the following prerequisite steps:

- a. Create a CA signed certificate for one of the AWS Trusted CAs for the Web interface of your CipherTrust Manager.
- b. Download and add the AWS root CA certificate from the same AWS trusted CA to External CA on CipherTrust Manager. For more information, refer to [Creating an External CA](#).
- c. Add the CA signed certificate to the CipherTrust Manager web interface. For more information, refer to [Using an externally generated server certificate for an interface](#).

12. Click **Update** to apply the new settings.

13. If you have a linked key store, [connect the key store](#).

Refresh All External Custom Key Stores and Keys for an AWS Account

This operation detects and imports all external key stores and external keys present on the KMS account.

This can include external key stores that don't have an external custom key store on the local CCKM. These could be associated with an external custom key store on another CCKM that is not clustered to the local CCKM, or to another vendor's external key manager. These key stores are display with the type **Remote**.

Refreshing all external custom key stores can set unlinked external custom key stores to a linked state. This happens when CCKM detects a corresponding KMS external key store for an local unlinked external custom key store. This operation can also set unlinked [HYOK keys](#) to a linked state.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > AWS KMS Accounts**. The AWS KMS Accounts page is displayed.
3. Click the name of the account that owns the key store. The account details page is displayed.
4. Next to the **External Custom Key Store** table, click **Refresh All**.

The **This may take a while...** message is displayed.

5. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

AWS XKS Performance Summary

We have tested different environments to capture performance metrics for the CipherTrust Cloud Key Manager (CCKM) AWS External Key Service integration. The performance results provided within this document demonstrate the effects of deployment choices on throughput for AWS HYOK key encrypt operations for one key within one key store. This can help you plan your CipherTrust Manager deployment to meet your performance needs for AWS XKS integration. There is an AWS requirement that encrypt and decrypt requests must be completed within 250 ms. Thus, the results are presented to show the throughput possible before meeting that threshold.

All environments used an open source k6.io tool (<https://k6.io/>) as the REST client to simulate encrypt requests from AWS KMS. The client ran on a virtual machine with 50 GB system volume, 8 CPUs, and 64 GB of memory. For the deployments using LUNA HSM as a key source, the HSM model of Luna K7 was employed.

Note

Actual performance numbers in your environment may be different. The results can vary based on factors, such as how and where the CipherTrust Manager is deployed, CipherTrust Manager resources, the location of cloud KMS, the key source of your choice, the network connectivity between the CipherTrust Manager, AWS Cloud and your key source, as well as how the traffic is load balanced.

Tested Environments

The following deployments were tested:

- Luna Network HSM as a key source (on premises or GCP):
 - Single node of CipherTrust Manager on premises, K6 client on premises, single node of HSM
 - Single node of CipherTrust Manager on premises, K6 client on premises, two nodes of HSM in HA mode
 - Single node of CipherTrust Manager on premises, K6 client on premises, HSM in Export mode and Clone mode
 - Single node of CipherTrust Manager on GCP, K6 client on AWS, single node of HSM
 - Single node of CipherTrust Manager on GCP, K6 client on AWS, two nodes of HSM in HA mode
 - Single node of CipherTrust Manager on GCP, K6 client on AWS, HSM in Export mode and Clone mode
 - Two Cluster nodes of CipherTrust Manager on GCP, K6 client on AWS, Two nodes of HSM in HA mode
- CipherTrust Manager as a key source (on premises and on AWS):
 - Single node of CipherTrust Manager on premises, K6 client on premises
 - Single node of CipherTrust Manager on GCP, K6 client on AWS
 - Single node of CipherTrust Manager on AWS, K6 client on AWS
 - Two clustered CipherTrust Manager nodes on AWS, K6 client on AWS

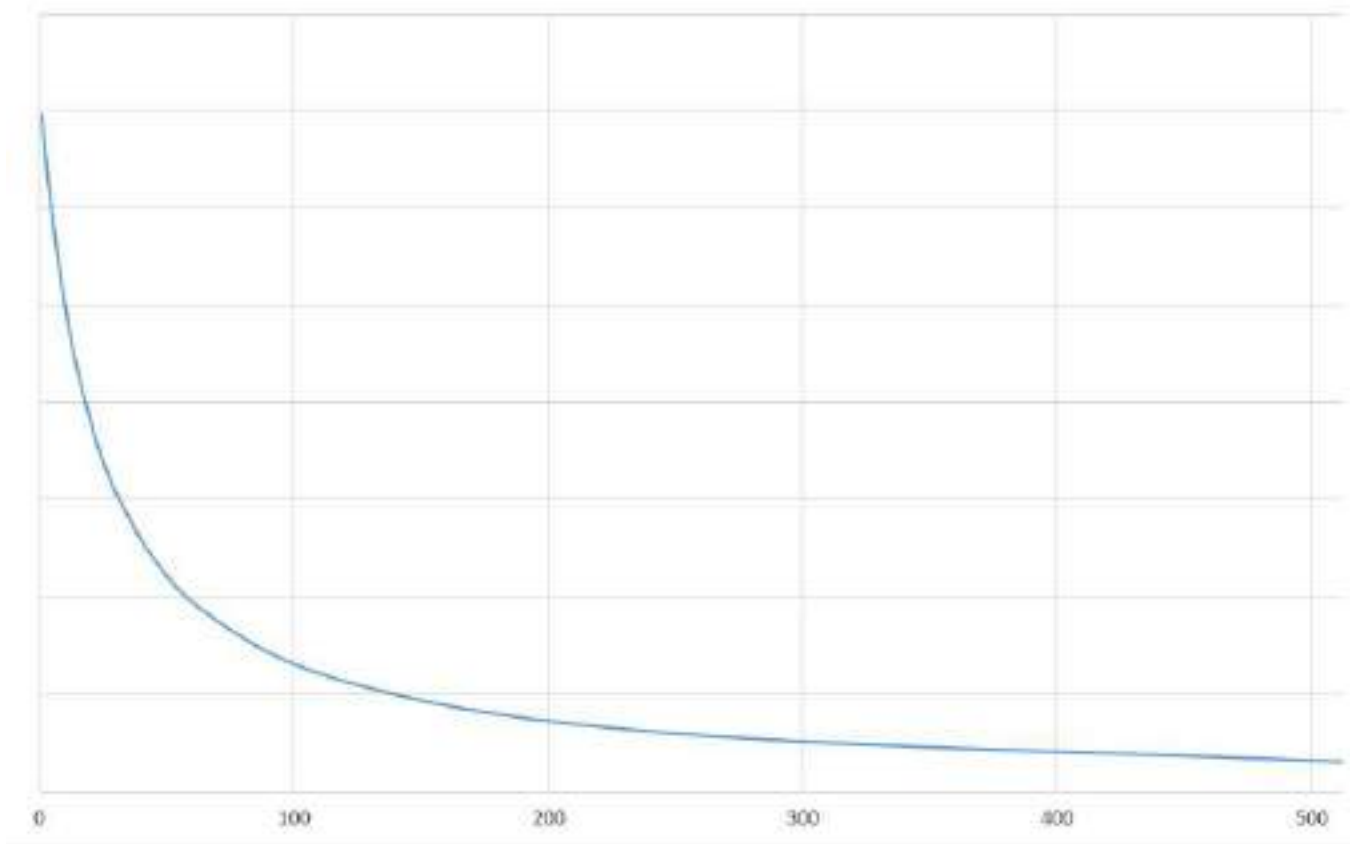
The following CipherTrust Manager release version was employed in the tested deployments:

Description	Value
Release version	2.11.1+9742

CipherTrust Managers were deployed as geographically close to the K6 client as possible to avoid potential network latencies, which can occur when crossing geographic regions. In your CCKM deployment, we similarly recommend deploying the virtual CipherTrust Manager instance geographically close to one of the AWS KMS regions where you intend to set up the AWS XKS. We recommend a network latency of round-trip communication of 35 ms or less between AWS KMS and the CipherTrust Manager. Also, if you are using Luna Network HSM as your key source for AWS HYOK, ensure the CipherTrust Manager and the HSM are geographically close and has the lowest possible latency. We recommend a network latency of round-trip communication of 25 ms or less between the CipherTrust Manager and the HSM.

The following graph was generated using a model of XKS performance. Simulated data is displayed. In this model, the internal latency of the CCKM was pre-set and the network latency between CipherTrust Manager/CCKM and the HSM was varied. The resulting data displayed in the graph captures the rapid degradation of performance with increasing network latency between CipherTrust Manager/CCKM and the HSM. In other words, increasing the latency to the HSM results in an (undesirable) non-linear degradation in performance.

Encrypt/Decrypt performance (Y axis) vs CipherTrust Manager-to-HSM latency (X axis, milliseconds)



Network Requirements

The following ports were opened to ensure CipherTrust Manager communication:

Used to communicate with CipherTrust Manager:

Type	Protocol	Port Number
SSH	TCP	22
HTTPS	TCP	443
postgresql (for cluster)	TCP	5432

Used for Connection to LUNA HSM:

Type	Protocol	Port Number
Secure Trusted Channel (STC)	TCP	5656
network trust link service (NTLS)	TCP	1792

Test process

The test consisted of starting a given number of virtual users to perform encrypt operations on the AWS XKS/HYOK key. Each user simulated a separate thread. Total test duration was 40 seconds for each reading. The test was divided into the following increments:

- Ramp-up time was 5 seconds. Virtual users were started.
- Test duration was 30 seconds for each reading. Virtual users make wrap requests during this time.
- Ramp-down time was 5 seconds. Virtual users were stopped until there were zero active virtual users.

Repeating the test in your environment

We have published the scripts used with `k6 .io` on [Github](#), for you to repeat the tests in your own environments as desired.

AWS XKS deployment results

- LUNA HSM as a Key Source
- CipherTrust Manager as a Key Source

LUNA HSM as a key source

LUNA HSM details

HSM Details

Description	Value
Firmware	7.3.3
HSM Model	Luna K6
Authentication Method	Password

Single node of CipherTrust Manager on premises, K6 client on premises, single node of HSM

4 CPU - 64 GB RAM

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on premises
- Single node of HSM on premises

Here are details of this deployment:

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	SanJose Lab ESXi
CM-2.11.1	50	4	16	SanJose Lab ESXi

The network latency between the CipherTrust Manager and the K6 client is **0.730** ms and between the CipherTrust Manager and the HSM is **0.230** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	14822	370.488464	44.19
20	19368	484.096976	58.21
30	18080	451.987655	100.13
40	18487	462.077984	130.53
50	19410	485.225354	154.07
60	18850	471.240975	195.45
70	18407	460.053475	226.3
80	20024	500.194051	252.84
90	19460	486.481693	280.93
100	18914	472.811809	326.98
110	19690	491.827879	372.5
120	18689	467.217404	401.62
130	19925	498.072093	382.72
140	19534	488.014201	456.43
150	20676	516.853259	507.51

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **500.194051** per second.

8 CPU - 64 GB RAM

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on premises
- Single node of HSM on premises

Here are details of this deployment:

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	SanJose Lab ESXi
CM-2.11.1	50	8	64	SanJose Lab ESXi

The network latency between the CipherTrust Manager and the K6 client is **0.730** ms and between the CipherTrust Manager and the HSM is **0.230** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	21940	548.107347	22.66
20	33727	843.16724	31.42
30	32958	823.825018	56.02
40	34005	849.923406	74.02
50	39687	992.116055	84.37
60	35144	878.444167	116.89
70	37021	925.508381	139.69
80	40590	1014.721688	145.48
90	36727	918.072933	179.38
100	36870	921.729155	205.61
110	41180	1029.403083	206.39
120	38085	952.035047	240.58
130	38755	968.739411	252.78
140	40895	1021.885088	265.68
150	40398	1009.86079	289.82
160	37406	934.97733	301.1
170	39361	983.550359	330.49
180	41499	1037.318017	353.11
190	38867	971.597279	354.4
200	37825	945.438374	399.83
210	42467	1061.660795	410.44
220	37750	943.64758	436.47
230	38568	964.13714	456.74
240	42287	1057.154044	483.3
250	38878	971.678607	506.67
260	38787	969.601137	506.7

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
270	41452	1036.028547	552.53
280	39361	983.826515	572.48
290	37992	949.628518	614.47
300	41278	1031.763766	617.22

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **968.739411** per second.

Single node of CipherTrust Manager on premises, K6 client on premises, two nodes of HSM in HA mode

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on premises
- Two nodes of HSM (in HA mode) on premises

Here are details of this deployment:

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	SanJose Lab ESXi
CM-2.11.1	50	8	64	SanJose Lab ESXi

The network latency between the CipherTrust Manager and the K6 client is **0.730** ms and between the CipherTrust Manager and the HSM is **0.366** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
50	42222	1055.399696	78.81
60	41850	1046.185283	98.45
70	44097	1102.260952	112.03
80	44598	1114.91903	133.04
90	43213	1080.301058	157.67
100	43963	1098.998373	176.29
110	43768	1094.192473	196.37
120	44191	1104.649837	211.35
130	45776	1144.330515	226.49
140	46178	1154.400332	242.02
150	45366	1134.078952	267.5

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
160	44746	1118.625993	292.36
170	43144	1078.4994	327.32
180	40015	1000.137207	370.45
190	47183	1179.329372	332.62
200	46888	1172.027319	360.34
210	44994	1124.722164	396.35
220	44728	1118.090287	425.53
230	45386	1134.351503	432.92
240	43351	1083.695209	472.14
250	48298	1207.424591	450.29
260	46645	1166.046241	487.5
270	46273	1156.613088	513.93
280	45939	1148.383001	534.94
290	45849	1146.062519	570.05

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **1134.078952** per second.

Single node of CipherTrust Manager on premises, K6 client on premises, HSM in Export mode and Clone mode

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on premises
- Two nodes of HSM (in Export mode and Clone mode) on premises

Here are details of this deployment:

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	SanJose Lab ESXi
CM-2.11.1	50	8	64	SanJose Lab ESXi

The network latency between the CipherTrust Manager and the K6 client is **0.730** ms and between the CipherTrust Manager and the HSM is **0.366** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
50	36653	916.252126	92.19
60	39611	988.931681	104.81
70	39942	998.44485	116.75
80	39077	976.81008	147.17
90	37210	928.869035	159.43
100	39449	986.073871	172.35
110	39829	995.497357	204.91
120	38793	969.736887	230.87
130	45039	1125.671565	216.52
140	42000	1049.847137	242.99
150	40730	1018.119718	272.74
160	43965	1099.099161	284.36
170	40373	1008.48485	304.81
180	40497	1012.157095	310.06
190	45421	1135.420918	324.31
200	41430	1035.709383	367.68
210	41805	1043.670215	380.37
220	41419	1035.446943	362.59
230	41486	1037.122511	411.73
240	42016	1050.17003	436.73
250	46029	1150.664962	436.38
260	41707	1042.623747	483.66
270	41874	1046.80338	506.85
280	46216	1155.302438	489.6
290	43389	1084.552883	503.54
300	41946	1048.570249	567.91

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **1018.119718** per second.

Single node of CipherTrust Manager on GCP, K6 client on AWS, single node of HSM

This deployment includes the following:

- Single node of CipherTrust Manager on GCP Cloud

- K6 client on AWS
- Single node of HSM on premises

Here are details of this deployment:

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	AWS (us-east-1)
CM-2.11.1	50	4	16	GCP (us-central-1)

The network latency between the CipherTrust Manager and the K6 client is **29.476** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
50	9583	239.413903	254.37
60	9582	239.396327	313.11
70	9409	235.152891	382.31
80	9685	241.639279	414.86
90	9899	247.195932	438.65
100	9239	230.564444	512.74
11	987	246.283066	532.34
120	9946	248.564488	587.08

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **239.413903** per second.

Single node of CipherTrust Manager on GCP, K6 client on AWS, two nodes of HSM in HA mode

This deployment includes the following:

- Single node of CipherTrust Manager on GCP Cloud
- K6 client on AWS
- Two nodes of HSM (in HA mode) on premises

Here are details of this deployment:

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	AWS (us-east-1)
CM-2.11.1	50	4	16	GCP (us-central-1)

The network latency between the CipherTrust Manager and the K6 client is **29.476** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
50	8473	211.770214	293.19
60	8796	219.767535	344.01
70	9030	225.36262	401.19
80	9119	227.559996	461.01
90	9251	231.20168	530.24
100	9019	225.17429	573.95

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **211.770214** per second.

Single node of CipherTrust Manager on GCP, K6 client on AWS, HSM in Export mode and Clone mode

This deployment includes the following:

- Single node of CipherTrust Manager on GCP Cloud
- K6 client on AWS
- Sing node of HSM (in Export mode and Clone mode) on premises

Here are details of this deployment:

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	AWS (us-east-1)
CM-2.11.1	50	4	16	GCP (us-central-1)

The network latency between the CipherTrust Manager and the K6 client is **29.476** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
50	8855	221.314997	285.33
60	9569	238.829389	315.33
70	9709	242.311532	364.37
80	9772	243.883417	424.64
90	9715	242.555781	479.29
100	9403	234.881415	533.49
110	9885	246.753465	583.31

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **221.314997** per second.

Two Cluster nodes of CipherTrust Manager on GCP, K6 client on AWS, two nodes of HSM in HA mode

This deployment includes the following:

- Two cluster nodes of CipherTrust Manager on GCP Cloud
- K6 client on AWS
- Two nodes of HSM (in HA mode) on premises

Here are details of this deployment:

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	AWS (us-east-1)
CM-2.11.1	50	4	16	GCP (us-central-1)
CM-2.11.1	50	4	16	GCP (us-central-1)

The network latency between the CipherTrust Manager and the K6 client is **29.476** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
50	8073	201.491849	323.26
60	12028	300.535836	289.25
70	16026	400.02639	209.51
80	16259	405.714295	242.97
90	16314	407.633572	277.7
100	17391	434.59756	314.73
110	17116	427.846397	320.92
120	17738	442.825828	336.91
130	18734	467.888171	368.49
140	18682	466.437955	396.66
150	18697	466.81158	427.09
160	19161	478.306436	453.75
170	18373	459.156292	477.44
180	19321	482.126969	526.24
190	19268	480.678308	561.78
200	18930	472.625649	594

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **407.633572** per second.

CipherTrust Manager as a key source

Single node of CipherTrust Manager on premises, K6 client on premises

4 CPU - 16 GB RAM

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on premises

Here are details of this deployment:

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	SanJose Lab ESXi
CM-2.11.1	50	4	16	SanJose Lab ESXi

The network latency between the CipherTrust Manager and the K6 client is **0.730** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
90	46726	1168.127735	106.25
100	45656	1141.353196	115.56
110	47627	1190.614976	121.09
120	47388	1184.652163	135.07
130	45875	1146.845218	150.92
140	47694	1192.265223	154.17
150	46099	1152.391072	176.23
160	46753	1168.772465	185.03
170	46862	1171.457966	193.13
180	46777	1169.389317	201.95
190	47740	1193.454638	214.36
200	45410	1135.215976	237.02
210	47217	1180.386041	245.45
220	46165	1154.094363	253.37
230	46833	1170.802745	260.4
240	46756	1168.802403	266.43

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
250	45592	1139.766613	293.12
260	47483	1187.019384	291.09
270	46340	1158.41622	302.89
280	47134	1178.260181	304.82
290	47486	1187.052589	331.72
300	46093	1152.273184	347.54

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **1154.094363** per second.

8 CPU - 32 GB RAM

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on premises

Here are details of this deployment:

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	SanJose Lab ESXi
CM-2.11.1	50	8	32	SanJose Lab ESXi

The network latency between the CipherTrust Manager and the K6 client is **0.730** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
200	108572	2714.132852	101.53
210	107513	2687.632044	107.13
220	107969	2699.092208	110.98
230	108460	2710.938208	116.41
240	105612	2640.272269	127.9
250	108884	2721.909181	128.57
260	105142	2628.295135	137.59
270	107586	2689.522924	136.49
280	108137	2703.191081	138.57
290	108689	2717.04585	143.47
300	106471	2661.621086	155.66

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
310	107243	2680.951166	155.33
320	104490	2611.856954	165.83
330	108011	2700.101438	163.48
340	106931	2673.031387	189.71
350	106606	2665.063073	176.93
360	106912	2672.510226	177
370	103426	2585.588873	201.11
380	105846	2646.028172	189.64
390	107108	2677.637119	195.95
400	106064	2651.43707	201.99
410	106670	2666.640371	200.42
420	107517	2687.715315	205.94
430	106777	2669.35678	212.39
440	106254	2656.301849	218.38
450	105980	2649.265159	226.94
460	108678	2716.786957	223.25
470	105867	2646.405144	229.43
480	106560	2663.778913	239
490	104493	2612.032179	245.18
500	107428	2685.476036	245.48
510	107797	2694.760117	241.88
520	105859	2646.351737	254.93
530	108447	2710.858349	253.88
540	107775	2693.815777	260.54
550	103867	2596.150634	289.24
560	91027	2275.505976	329.24
570	98877	2471.649406	302.61
580	104215	2605.251268	289.67
590	106646	2665.986521	291.77
600	105292	2631.839692	293.82

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **2646.351737** per second.

16 CPU - 32 GB RAM

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on premises

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	SanJose Lab ESXi
CM-2.11.1	50	16	32	SanJose Lab ESXi

The network latency between the CipherTrust Manager and the K6 client is **0.730** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
450	237073	5926.600974	105.09
500	237891	5946.404505	112.32
550	232386	5809.427737	125.85
600	239223	5979.227757	130.02
650	237962	5948.605547	141.17
700	239296	5981.690633	150.2
750	238349	5958.170732	159.48
800	241055	6025.977057	169.32
850	235010	5874.795624	182.45
900	225869	5646.398802	210.82
950	238279	5956.255117	198.68
1000	237937	5947.811578	207.82
1050	237716	5942.213984	218.33
1100	238211	5954.88409	226
1150	240137	6003.072035	236.04
1200	242326	6057.87667	244.45
1210	242763	6068.837597	246.14
1220	243201	6079.798524	247.82
1250	244514	6112.681305	252.87

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **6112.681305** per second.

Single node of CipherTrust Manager on GCP, K6 client on AWS

This deployment includes the following:

- Single node of CipherTrust Manager on GCP Cloud
- K6 client on AWS

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	AWS (us-east-1)
CM-2.11.1	50	4	16	GCP (us-central-1)

The network latency between the CipherTrust Manager and the K6 client is **29.476** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
50	21884	547.093178	113.14
60	22434	560.710964	132.19
70	22457	561.273174	157.98
80	23081	576.869185	168.56
90	23086	576.986894	196.33
100	23084	576.615912	229.77
110	23611	589.93677	244.65
120	23612	389.971801	264.87
130	23119	577.695201	300.05
140	23127	577.954035	318.79
150	23266	581.336016	358.63
160	24791	619.637581	344.87
170	24067	601.346708	408.58
180	24292	607.122047	412.1
190	23905	597.135154	429.96
200	23974	599.1375	455.52
210	22225	555.446348	500.47
220	24882	621.629199	471.24
230	25116	627.414426	482.22
240	25387	634.487322	510.06
250	24851	389.19375	561.02
260	24889	622.076392	581.02

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **389.971801** per second.

Single node of CipherTrust Manager on AWS, K6 client on AWS

This deployment includes the following:

- Single node of CipherTrust Manager on AWS
- K6 client on AWS

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	AWS (us-east-1)
CM-2.11.1	50	4	16	AWS (us-east-1)

The network latency between the CipherTrust Manager and the K6 client is **0.515** ms.

Results:

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
50	37535	938.334429	73.35
100	38745	968.593287	140.72
150	38386	959.62642	211.53
200	37192	929.770284	298.63
250	37392	934.748604	369.35
300	36731	918.200794	450.55
350	37259	931.380753	503
400	37976	949.319603	550.38

Observation:

For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **929.770284** per second.

Two clustered CipherTrust Manager nodes on AWS, K6 client on AWS

This deployment includes the following:

- Two clustered nodes of CipherTrust Manager on AWS
- K6 client on AWS

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
k6 Client	50	8	64	AWS (us-east-1)

Setup	System Volume (GB)	CPUs	Memory (GB)	Location
CM-2.11.1	50	4	16	AWS (us-east-1)
CM-2.11.1	50	4	16	AWS (us-east-1)

The network latency between the CipherTrust Manager and the K6 client is **0.515** ms.

Results:

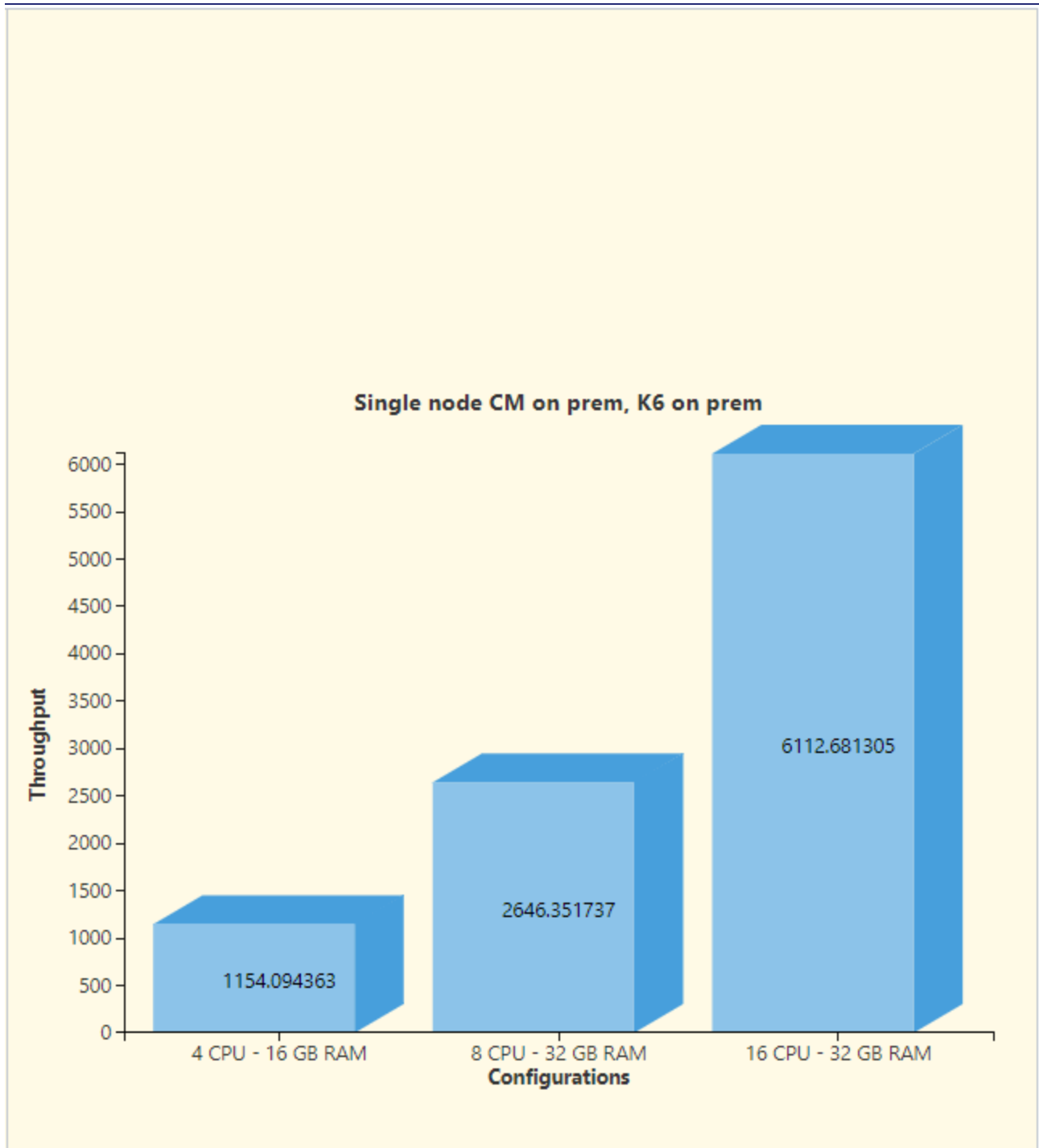
Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
50	75488	1887.087646	37.04
100	75724	1893.008797	74.1
150	78373	1959.190548	104.05
200	78286	1957.096186	136.88
250	76990	1924.686689	174.14
300	78426	1960.606549	208.27
350	77211	1930.219812	250.07
400	76152	1903.644125	282.48
450	78339	1958.447676	319.52
500	77479	1936.947471	348.31
550	77423	1935.50092	379.68
600	78721	1967.922819	412.16
650	78226	1955.612616	435.96
700	77420	1935.364058	477.81
750	78340	1958.367054	515.7
800	75553	1888.684818	550.19

Observation:

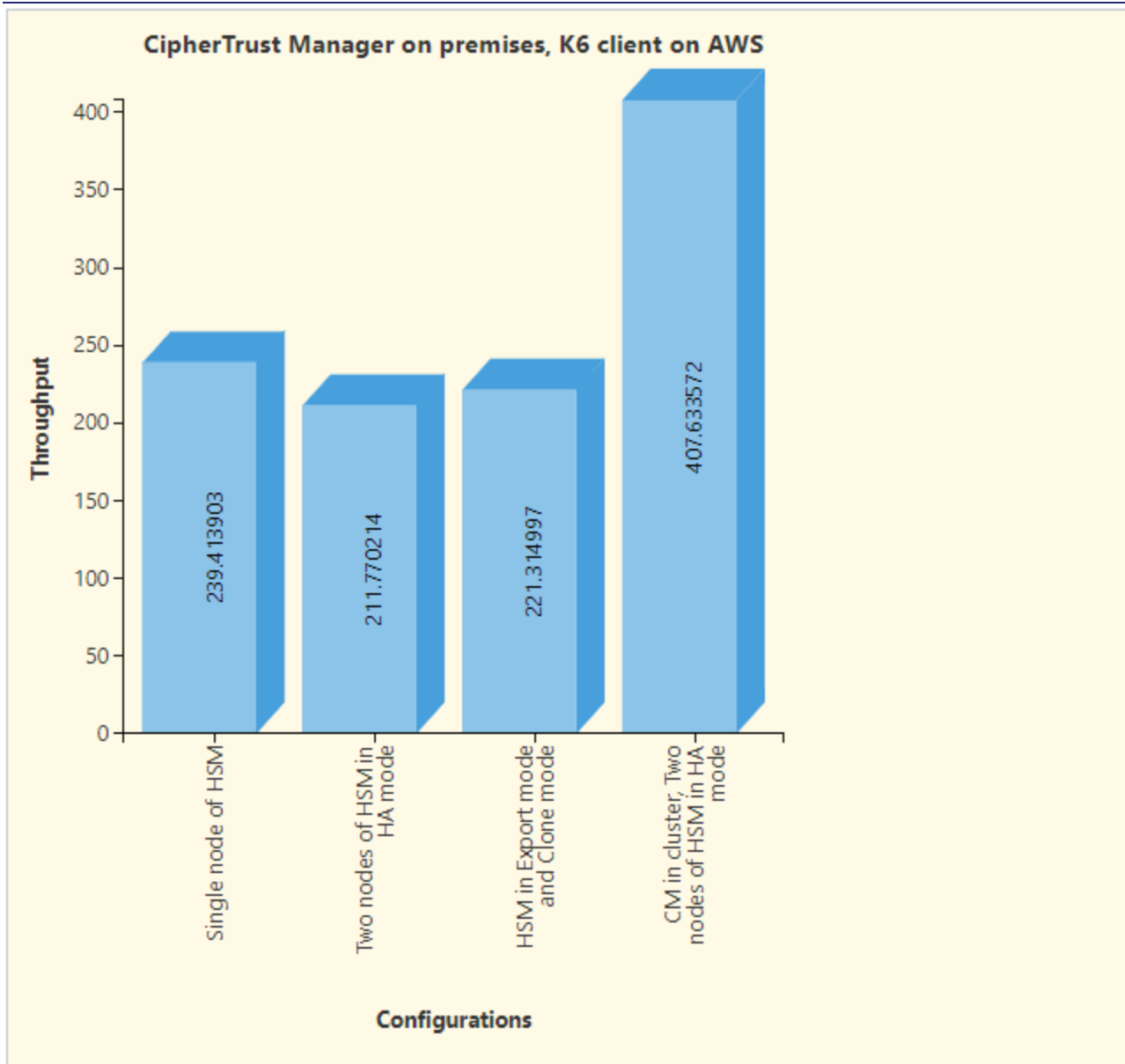
For this deployment, a response time compliance of 250 ms was met for a maximum throughput of **1930.219812** per second.

Conclusion

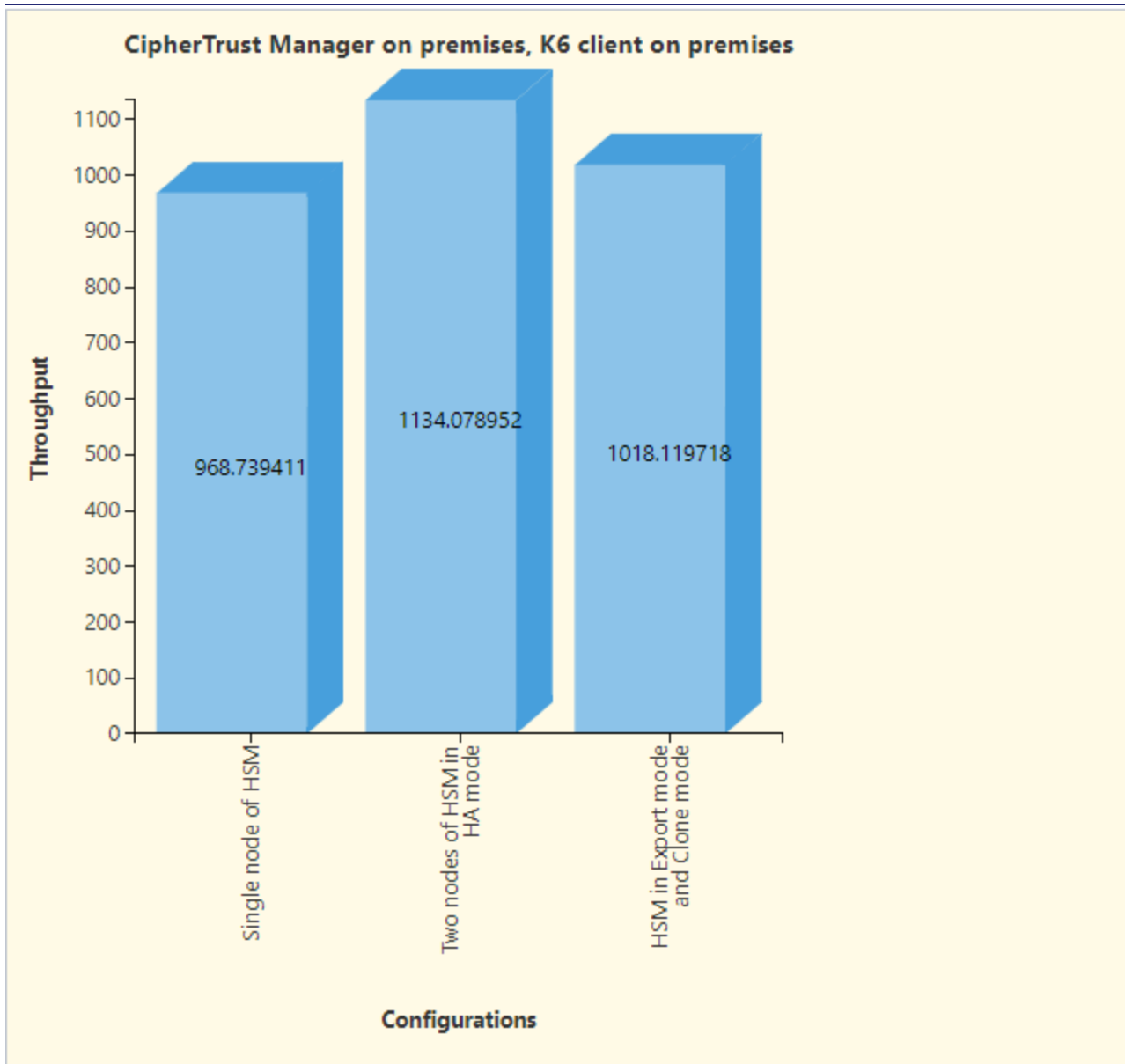
- The performance improves with higher number of CPUs. The minimum number of CPUs and RAM required: 4 CPUs and 16 GB RAM.



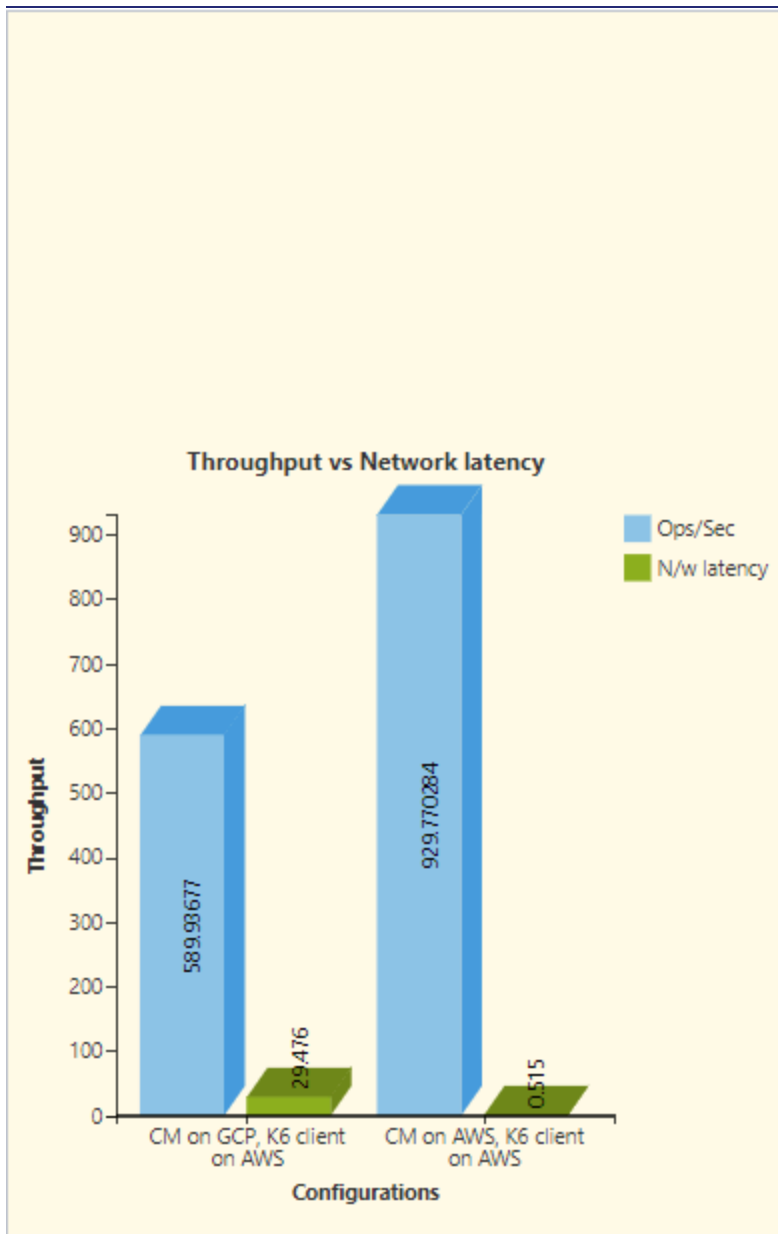
- Performance linearly improves with adding more CipherTrust Manager nodes.

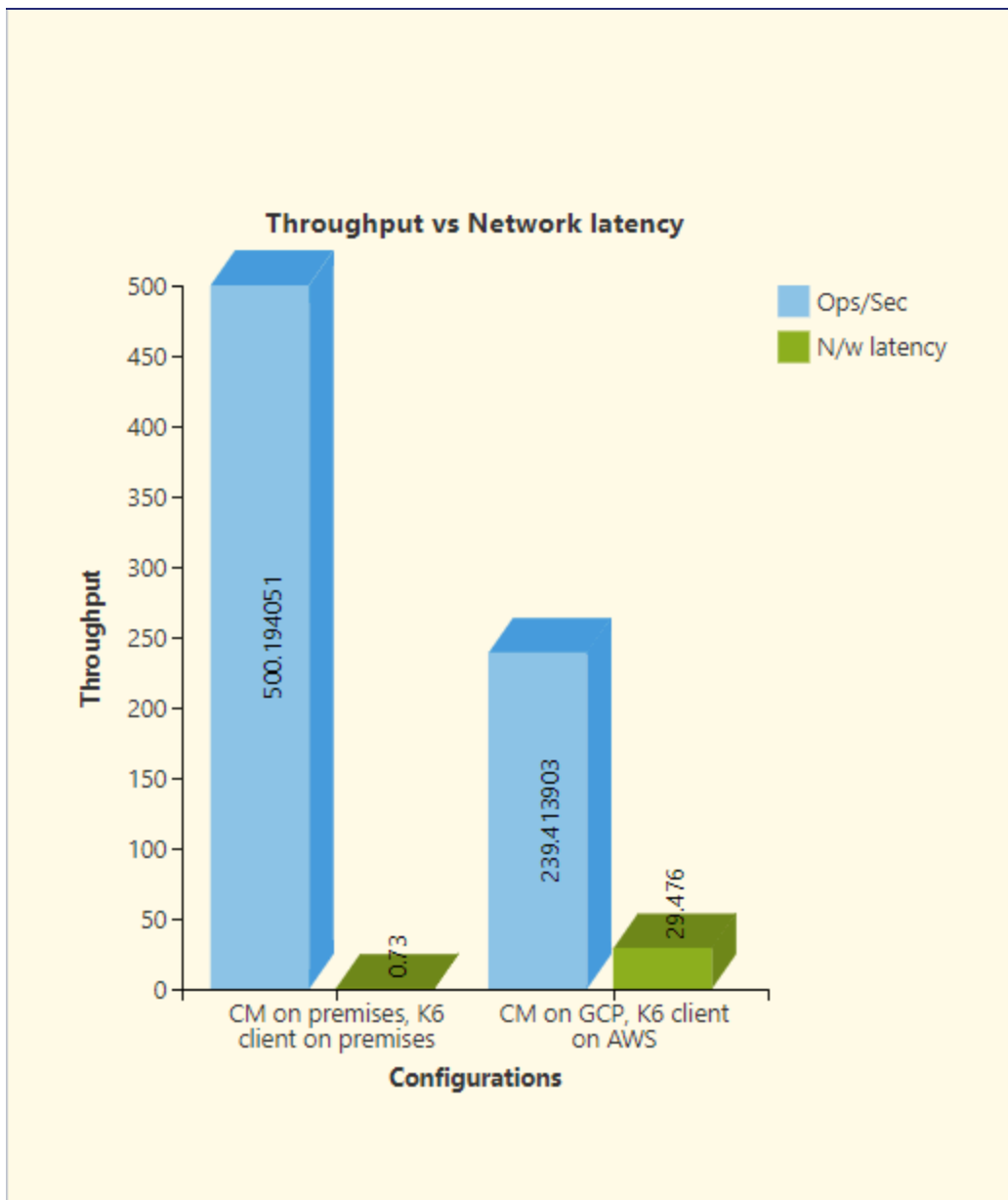


- Partition mode (Export vs Clone) does not have any impact on performance.
- Changing HSM connections (STC vs NTLS) does not have significant affect on performance.



- If you are using Luna Network HSM as your key source for AWS HYOK, ensure the CipherTrust Manager and the HSM are geographically close and has the lowest possible latency. Increasing the latency to the HSM results in degradation in performance.





- Different models of Luna 7 HSMs have their maximum throughput. Depending on your requirement, ensure to choose a model that can meet your needs. Ensure the HSM throughput does not become a bottle neck for the deployment. Refer to Luna HSM documentation for more information. CCKM uses the AES-GCM small packet encryption algorithm on a Luna HSM for AWS XKS operations. Refer to the **Luna Network HSM 7 Specifications** section under the **Specifications** tab (next to **Overview** and **Features**) within **Thales LUNA HSM** to view the performance numbers for the various Luna HSM models using the AES-GCM small packet encryption algorithm.
- The results captured was for one HYOK key in one key store, which represents the total throughput. When the number of key stores is higher, the total throughput will be cumulative across key stores.

Microsoft Azure Resources

Refer to the following sections to manage Azure resources on CCKM:

- Prerequisites
- Managing Azure Vaults
- Managing Azure Keys
- Managing Azure Certificates
- Managing Azure Secrets
- Scheduling Operations
- Managing Azure Reports
- Viewing Azure Subscriptions
- Allowing AD Users to Manage Azure Vaults
- Configuring Connection to Azure Using Private Link
- Downloading Azure Metadata

Prerequisites

This section describes the prerequisites to manage Azure resources on CCKM.

- Prerequisites for Azure Cloud
- Prerequisites for Azure Stack Cloud with Azure AD
- Prerequisites for Azure Stack Cloud with Azure ADFS
- Azure Key Backup Storage Requirement

Note

- To use Luna HSM as a key source, a Luna HSM connection must already exist on the CipherTrust Manager. The Luna HSM partition must already be added in CCKM. Refer to [Prerequisites](#).
- FM-enabled Luna HSM is not supported as a key source.

Note

To use DSM as a key source, a DSM connection must already exist on the CipherTrust Manager. The DSM domain must already be added in CCKM. Refer to [Prerequisites](#).

Azure Key Backup Storage Requirements

Number of versions in the Azure key	Cloud key backup size for RSA 2048 key	Cloud key backup size for RSA 3072 key	Cloud key backup size for RSA 4096 key
1 version	24 kB	24 kB	24 kB
10 version	80 kB	88 kB	144 kB
20 version	144 kB	192 kB	232kB

Note

The above sample results are from one of Azure production accounts. The numbers might be different in customer environments.

Prerequisites for Azure Cloud

Before adding an Azure cloud in CCKM:

1. Register CCKM App on Azure Portal
2. Subscribe CCKM App on Azure Portal
3. Set Required Permissions on CCKM App on Azure Portal
4. Assign CCKM App Permissions
5. Add Azure Connection on CipherTrust Manager

Register CCKM App on Azure Portal

Before adding an Azure cloud to CCKM, you must register the CCKM app and assign required permissions on the Azure portal. Then, depending on the type of app credential you plan to employ, either create a key (client secret) on the Azure portal or generate a certificate from CCKM, download it, and then upload it to Azure. This entire process generates the connection data needed to configure the Azure cloud.

To register the app:

1. Create an Azure Active Directory application on the Azure Portal:

On **Azure Active Directory > App registrations > New registration**, provide the following parameters:

- a. Choose a **Name** for the app that CCKM will use to access Azure.
- b. Select the account type under **Supported account types**. Select either of the following:
 - Accounts in this organizational directory only (<account-name> (Default Directory) only - Single tenant)

- Accounts in any organizational directory (Any Azure AD directory - Multitenant)

c. Click **Register**. The CCKM app creation starts. The app creation might take some time.

2. Access the new app under **App registrations**.
3. From **App registrations** > **{App Name}** > **Overview**, copy **Application (client) ID** and **Directory (tenant) ID**.
4. Navigate to **App registrations** > **{App Name}** > **Manage** > **Certificates & secrets**.
5. Create a new client secret or upload the certificate:
 - Secret (password) — The user will generate a secret key in Azure when registering the app, and then copy the secret key and provide it to the app.
 - Certificate (public key) — The user will create a private key and public key pair locally, create a certificate for the public key, and then provide the certificate to Azure when registering the app. For the private key, the app will create a client assertion and send it to Azure when making OAuth authentication calls.

Depending on your requirements, you can use a self-signed or CA-signed certificate. Refer to [Certificate creation methods](#) in the Azure documentation for details.

Subscribe CCKM App on Azure Portal

1. Access **Subscriptions** > **{Subscription name}** > **Access control (IAM)**.
2. Click **Add** > **Add role assignment**. The **Add role assignment** pane is displayed.
3. Select **Reader** from the **Role** drop-down list.
4. Select **User, group, or service principal** from the **Assign access to** drop-down list.
5. Under **Select**, enter the name of your CCKM app.
6. Click **Save**.

Set Required Permissions on CCKM App on Azure Portal

1. Access **App Registrations** > **{App Name}** > **Manage** > **API permissions**.
2. Click **+ Add a permission** and add the following APIs and their associated permissions:
 - **Azure Key Vault** (Delegated permissions): Have full access to the Azure Key Vault service.
 - **Azure Service Management** (Delegated permissions): Access Azure Service Management as organization users (preview).

Assign CCKM App Permissions

Assign CCKM App Permissions to Required Key Vault in the Azure Portal

Azure supports two types of permission models, vault access policy and azure role-based access control. Steps to assign CCKM app permissions to your key vaults on Azure portal vary based on the permission model of the vault, as described below.

Vault Access Policy

1. Access **Key vaults** > **{Key Vault Name}** > **Settings** > **Access policies**.
2. On the right, click **+ Add Access Policy**.
3. Add the following details:
 - From the **Key permissions** list, select **Key Management Operations** and **Privileged Key Operations**.
 - From the **Secret permissions** list, select **Secret Management Operations** and **Privileged Secret Operations**.
 - From the **Certificate permissions** list, select **Certificate Management Operations** and **Privileged Certificate Operations**.
 - Next to the **Select principal** label, click **None selected**, browse the CCKM app, and select it.
4. Click **Add**.

Azure Role-based Access Control

1. Access **Key vaults** > **{Key Vault Name}** > **Access control (IAM)**. The right pane provides options to view your existing level of access, assign a role, and view access to the vault.
2. Under **Grant access to this resource**, click **Add role assignment**.
3. On the **Roles** tab of the **Add role assignment** page, select **Key Vault Administrator**. Use the search field to search for the role.
4. Click **Next**.
5. On the **Members** tab, make sure that **Assign access to** is set to **User, group, or service principal**.
6. Click **+ Select members**.
7. On the right, in the **Select members** pane, select the desired CCKM app. Use the search field to search for your CCKM app. The **Selected members** section shows the selected app.
8. In the right pane, click **Select**.

-
9. (Optional) Provide a basic **Description** of the member.
 10. Click **Next**.
 11. Review the details. If the details are incorrect or you want to modify them, click **Previous** and update the details.
 12. Click **Review + assign**. The selected role with desired permissions is assigned to the CCKM app.

Assign CCKM App Permissions to the Required Managed HSM Vault in the Azure Portal

1. Sign in to the Microsoft Azure portal.
2. Click **Azure Key Vault Managed HSMs**.
3. Select the desired Azure Key Vault Managed HSM from the list. The selected Azure Key Vault Managed HSM opens.
4. Click **Settings > Local RBAC**.
5. Add a Crypto Officer.
 - a. Click **Add**. The Role Assignment window appears.
 - b. Under Role, select **Managed HSM Crypto Officer**.
 - c. Under Scope, select **All keys (/ or /keys)**.
 - d. Click **Select security principal**. The Add principal wizard appears.
 - e. Search for and select the desired **principal**.

Note

Ensure that the type of the selected principal is an enterprise application.

- f. Click **Select**.
 - g. Click **Create**.
6. Add a Crypto User.
 - a. Click **Add**. The Role Assignment window appears.
 - b. Under Role, select **Managed HSM Crypto User**.

- c. Under Scope, select **All keys (/ or /keys)**.
- d. Click **Select security principal**. The Add principal wizard appears.
- e. Search for and select the desired **principal**.

Note

Ensure that the type of the selected principal is an enterprise application.

- f. Click **Select**.
- g. Click **Create**.

Add Azure Connection on CipherTrust Manager

Before you can add an Azure vault to the CCKM, an Azure connection must already exist on the CipherTrust Manager. A CipherTrust Manager administrator manages connections to external resources on the **Access Management > Connections Management** page of the CipherTrust Manager GUI. Refer to [Connections Management](#) for details.

Now, [Azure vaults](#) and [Azure keys](#) can be managed on the CipherTrust Manager.

Prerequisites for Azure Stack Cloud with Azure AD

Before adding an Azure Stack cloud with Azure AD in CCKM:

1. [Register CCKM App on Azure Portal](#)
2. [Set Required Permissions on CCKM App on Azure Portal](#)
3. [Connect with Azure AD](#)
4. [Subscribe CCKM App on Azure Stack Portal](#)
5. [Assign CCKM App Permissions to Required Key Vault on Azure Stack Portal](#)
6. [Add Azure Connection on CipherTrust Manager](#)

Register CCKM App on Azure Portal

Before adding an Azure cloud to CCKM, you must register the CCKM app and assign required permissions on the Azure portal. Then, depending on the type of app credential you plan to employ, either create a key (client secret) on the Azure portal or generate a certificate from CCKM, download it, and then upload it to Azure. This entire process generates the connection data needed to configure the Azure cloud.

To register the app:

1. Create an Azure Active Directory application on the Azure Portal:

On **Azure Active Directory > App registrations > New registration**, provide the following parameters:

- a. Choose a **Name** for the app that CCKM will use to access Azure.
 - b. Select the account type under **Supported account types**. Select either of the following:
 - Accounts in this organizational directory only (<account-name> (Default Directory) only - Single tenant)
 - Accounts in any organizational directory (Any Azure AD directory - Multitenant)
 - c. Click **Register**. The CCKM app creation starts. The app creation might take some time.
2. Access the new app under **App registrations**.
 3. From **App registrations > {App Name} > Overview**, copy **Application (client) ID** and **Directory (tenant) ID**.
 4. Navigate to **App registrations > {App Name} > Manage > Certificates & secrets**.
 5. Create a new client secret or upload the certificate:
 - Secret (password) — The user will generate a secret key in Azure when registering the app, and then copy the secret key and provide it to the app.
 - Certificate (public key) — The user will create a private key and public key pair locally, create a certificate for the public key, and then provide the certificate to Azure when registering the app. For the private key, the app will create a client assertion and send it to Azure when making OAuth authentication calls.

Depending on your requirements, you can use a self-signed or CA-signed certificate. Refer to [Certificate creation methods](#) in the Azure documentation for details.

Set Required Permissions on CCKM App on Azure Portal

1. Access **App Registrations > {App Name} > Manage > API permissions**.
2. Click **+ Add a permission** and add the following APIs and their associated permissions:
 - **Azure Key Vault** (Delegated permissions): Have full access to the Azure Key Vault service.
 - **Azure Service Management** (Delegated permissions): Access Azure Service Management as organization users (preview).

Connect with Azure AD

1. Log on to your Azure AD VM.
2. Connect to Azure Stack Hub with PowerShell as a user. Refer to [Connect to Azure Stack Hub with PowerShell as a user - Azure Stack Hub](#) for details.

- Run the following command in Windows PowerShell:

```
Get-AzureRmEnvironment -name AzureStackUser | Format-List
```

A sample output is shown below:

```
Name : AzureStackUser
EnableAdfsAuthentication : False
OnPremise : False
ActiveDirectoryServiceEndpointResourceId : https://management.azurecc
km.onmicrosoft.com/7f683eac-8000-2a43-3f9e-86d9117cc571
AdTenant :
GalleryUrl : https://providers.azuresta
ck.local:30016/
ManagementPortalUrl :
ServiceManagementUrl :
PublishSettingsFileUrl :
ResourceManagerUrl : https://management.local.a
zurestack.external
SqlDatabaseDnsSuffix :
StorageEndpointSuffix : local.azurestack.external
ActiveDirectoryAuthority : https://login.microsoftonl
ine.com/
GraphUrl : https://graph.windows.net/
GraphEndpointResourceId : https://graph.windows.net/
TrafficManagerDnsSuffix :
AzureKeyVaultDnsSuffix : vault.local.azurestack.ext
ernal
DataLakeEndpointResourceId :
AzureDataLakeStoreFileSystemEndpointSuffix :
AzureDataLakeAnalyticsCatalogAndJobEndpointSuffix :
AzureKeyVaultServiceEndpointResourceId : https://vault.local.azures
tack.external
AzureOperationalInsightsEndpointResourceId :
AzureOperationalInsightsEndpoint :
AzureAnalysisServicesEndpointSuffix :
VersionProfiles : {}
ExtendedProperties : {}
BatchEndpointResourceId
```

The sample output above lists a number of links that are required while creating an Azure connection on the CipherTrust Manager. Use the *actual* links that are returned by the command on your setup. Refer to [Add](#)

Azure Connection on CipherTrust Manager for details.

The fields in the sample output and on the **Configure Azure Stack** page of the Add Connection dialog box on the CipherTrust Manager differ slightly. Here is the mapping to help you configure the connection appropriately.

Sample Output	Azure Connection Manager
ActiveDirectoryServiceEndpointResourceId	Management URL
ResourceManagerUrl	Resource Manager URL
ActiveDirectoryAuthority	Active Directory Endpoint
AzureKeyVaultDnsSuffix	Key Vault DNS Suffix
AzureKeyVaultServiceEndpointResourceId	Vault Resource URL

- Download the SSL certificate of the Azure Stack portal. Refer to online resources for details.

Subscribe CCKM App on Azure Stack Portal

- Access **Subscriptions** > {Subscription name} > **Access control (IAM)**.
- Click **Add** > **Add role assignment**. The **Add role assignment** pane is displayed.
- Select **Reader** from the **Role** drop-down list.
- Select **User, group, or service principal** from the **Assign access to** drop-down list.
- Under **Select**, enter the name of your CCKM app.
- Click **Save**.

Assign CCKM App Permissions to Required Key Vault on Azure Stack Portal

Azure supports two types of permission models, vault access policy and azure role-based access control. Steps to assign CCKM app permissions to your key vaults on Azure portal vary based on the permission model of the vault, as described below.

Vault Access Policy

- Access **Key vaults** > {Key Vault Name} > **Settings** > **Access policies**.
- On the right, click **+ Add Access Policy**.
- Add the following details:
 - From the **Key permissions** list, select **Key Management Operations** and **Privileged Key Operations**.

- From the **Secret permissions** list, select **Secret Management Operations** and **Privileged Secret Operations**.
- From the **Certificate permissions** list, select **Certificate Management Operations** and **Privileged Certificate Operations**.
- Next to the **Select principal** label, click **None selected**, browse the CCKM app, and select it.

4. Click **Add**.

Azure Role-based Access Control

1. Access **Key vaults** > **{Key Vault Name}** > **Access control (IAM)**. The right pane provides options to view your existing level of access, assign a role, and view access to the vault.
2. Under **Grant access to this resource**, click **Add role assignment**.
3. On the **Roles** tab of the **Add role assignment** page, select **Key Vault Administrator**. Use the search field to search for the role.
4. Click **Next**.
5. On the **Members** tab, make sure that **Assign access to** is set to **User, group, or service principal**.
6. Click **+ Select members**.
7. On the right, in the **Select members** pane, select the desired CCKM app. Use the search field to search for your CCKM app. The **Selected members** section shows the selected app.
8. In the right pane, click **Select**.
9. (Optional) Provide a basic **Description** of the member.
10. Click **Next**.
11. Review the details. If the details are incorrect or you want to modify them, click **Previous** and update the details.
12. Click **Review + assign**. The selected role with desired permissions is assigned to the CCKM app.

Add Azure Connection on CipherTrust Manager

Before you can add an Azure vault to the CCKM, an Azure connection must already exist on the CipherTrust Manager. A CipherTrust Manager administrator manages connections to external resources on the **Access Management** > **Connections Management** page of the CipherTrust Manager GUI. Refer to [Connections Management](#) for details.

Now, [Azure vaults](#) and [Azure keys](#) can be managed on the CipherTrust Manager.

Prerequisites for Azure Stack Cloud with Azure ADFS

Before adding an Azure Stack cloud with Azure ADFS in CCKM:

1. [Create Service Principal for Azure Stack Cloud with ADFS](#)
2. [Connect with Azure AD](#)
3. [Assign CCKM App Permissions to Required Key Vault on Azure Stack Portal](#)
4. [Add Azure Connection on CipherTrust Manager](#)

Create Service Principal for Azure Stack Cloud with ADFS

A service principal for CCKM in ADFS is needed if you are using ADFS as the identity provider. Refer to the [Azure Stack documentation](#) for details.

Connect with Azure AD

1. Log on to your Azure AD VM.
2. Connect to Azure Stack Hub with PowerShell as a user. Refer to [Connect to Azure Stack Hub with PowerShell as a user - Azure Stack Hub](#) for details.
3. Run the following command in Windows PowerShell:

```
Get-AzureRmEnvironment -name AzureStackUser | Format-List
```

A sample output is shown below:

```
Name : AzureStackUser
EnableAdfsAuthentication : False
OnPremise : False
ActiveDirectoryServiceEndpointResourceId : https://management.azure.com/7f683eac-8000-2a43-3f9e-86d9117cc571
AdTenant :
GalleryUrl : https://providers.azurestack.local:30016/
ManagementPortalUrl :
ServiceManagementUrl :
PublishSettingsFileUrl :
ResourceManagerUrl : https://management.local.azurestack.external
```

```

SqlDatabaseDnsSuffix           :
StorageEndpointSuffix         : local.azurestack.external
ActiveDirectoryAuthority       : https://login.microsoftonline.com/
GraphUrl                       : https://graph.windows.net/
GraphEndpointResourceId        : https://graph.windows.net/
TrafficManagerDnsSuffix       :
AzureKeyVaultDnsSuffix        : vault.local.azurestack.external
DataLakeEndpointResourceId     :
AzureDataLakeStoreFilesystemEndpointSuffix :
AzureDataLakeAnalyticsCatalogAndJobEndpointSuffix :
AzureKeyVaultServiceEndpointResourceId : https://vault.local.azurestack.external
AzureOperationalInsightsEndpointResourceId :
AzureOperationalInsightsEndpoint :
AzureAnalysisServicesEndpointSuffix :
VersionProfiles               : {}
ExtendedProperties             : {}
BatchEndpointResourceId

```

The sample output above lists a number of links that are required while creating an Azure connection on the CipherTrust Manager. Use the *actual* links that are returned by the command on your setup. Refer to [Add Azure Connection on CipherTrust Manager](#) for details.

The fields in the sample output and on the **Configure Azure Stack** page of the Add Connection dialog box on the CipherTrust Manager differ slightly. Here is the mapping to help you configure the connection appropriately.

Sample Output	Azure Connection Manager
ActiveDirectoryServiceEndpointResourceId	Management URL
ResourceManagerUrl	Resource Manager URL
ActiveDirectoryAuthority	Active Directory Endpoint
AzureKeyVaultDnsSuffix	Key Vault DNS Suffix
AzureKeyVaultServiceEndpointResourceId	Vault Resource URL

- Download the SSL certificate of the Azure Stack portal. Refer to online resources for details.

Assign CCKM App Permissions to Required Key Vault on Azure Stack Portal

Azure supports two types of permission models, vault access policy and azure role-based access control. Steps

to assign CCKM app permissions to your key vaults on Azure portal vary based on the permission model of the vault, as described below.

Vault Access Policy

1. Access **Key vaults** > **{Key Vault Name}** > **Settings** > **Access policies**.
2. On the right, click **+ Add Access Policy**.
3. Add the following details:
 - From the **Key permissions** list, select **Key Management Operations** and **Privileged Key Operations**.
 - From the **Secret permissions** list, select **Secret Management Operations** and **Privileged Secret Operations**.
 - From the **Certificate permissions** list, select **Certificate Management Operations** and **Privileged Certificate Operations**.
 - Next to the **Select principal** label, click **None selected**, browse the CCKM app, and select it.
4. Click **Add**.

Azure Role-based Access Control

1. Access **Key vaults** > **{Key Vault Name}** > **Access control (IAM)**. The right pane provides options to view your existing level of access, assign a role, and view access to the vault.
2. Under **Grant access to this resource**, click **Add role assignment**.
3. On the **Roles** tab of the **Add role assignment** page, select **Key Vault Administrator**. Use the search field to search for the role.
4. Click **Next**.
5. On the **Members** tab, make sure that **Assign access to** is set to **User, group, or service principal**.
6. Click **+ Select members**.
7. On the right, in the **Select members** pane, select the desired CCKM app. Use the search field to search for your CCKM app. The **Selected members** section shows the selected app.
8. In the right pane, click **Select**.
9. (Optional) Provide a basic **Description** of the member.
10. Click **Next**.
11. Review the details. If the details are incorrect or you want to modify them, click **Previous** and update the details.

12. Click **Review + assign**. The selected role with desired permissions is assigned to the CCKM app.

Add Azure Connection on CipherTrust Manager

Before you can add an Azure vault to the CCKM, an Azure connection must already exist on the CipherTrust Manager. A CipherTrust Manager administrator manages connections to external resources on the **Access Management > Connections Management** page of the CipherTrust Manager GUI. Refer to [Connections Management](#) for details.

Now, [Azure vaults](#) and [Azure keys](#) can be managed on the CipherTrust Manager.

Managing Azure Vaults

This section describes how to manage Azure vaults on CCKM.

Before proceeding, make sure to fulfill [prerequisites](#).

Adding Existing Vaults

To add an existing Azure vault to the CCKM:

1. Log on to the **CipherTrust Manager** GUI as administrator.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab of the Azure Key Vaults page is displayed.
4. Click **Add Existing Vault**. The Add Existing Key Vaults dialog box is displayed.
5. Select/enter the following details:
 - a. From the **Azure Connection** drop-down list, select the desired connection. Start typing to search for connections with a matching string in their names. The linked subscriptions are auto-populated in the **Subscription** drop-down list.
 - b. From the **Subscription** drop-down list, search and select the desired subscription. Start typing to search for subscriptions with a matching string in their names.
 - c. Select the vaults that you want to add to the CCKM.

To select a vault from an Azure Managed HSM pool:

- a. Select **Managed HSM**.
- b. Use the **Vault Name** check boxes to select vaults from an Azure Managed HSM pool.

To select an Azure vault:

- a. Make sure that **Managed HSM** is clear.

-
- b. Use the **Vault Name** check boxes to select Azure vaults.

6. Click **Save**. The Azure vault is added to the CCKM.

A message **vault added successfully...** is displayed on the screen.

Refreshing Azure Keys

Refreshing is the process of downloading keys created on the Azure vaults to CCKM. You can refresh keys from individual vaults or all vaults at once.

- [Refreshing Specific Vaults](#)
- [Refreshing All Vaults](#)

Refreshing Specific Vaults

To refresh keys of a specific vault:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab of the Azure Key Vaults page is displayed. This tab displays the list of Azure key vaults.
3. Click the overflow icon () corresponding to the desired Azure vault and click **Refresh Now**.

A message **Refresh started...** is displayed on the screen.

After successful refresh, the refreshed keys are listed on the **Cloud Keys > Azure > Azure Keys** page. Refer to [Viewing Azure Keys](#) for details.

Refreshing All Vaults

To refresh keys of all vaults:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab of the Azure Key Vaults page is displayed. This tab displays the list of Azure key vaults.
3. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all Azure Key Vaults is a time intensive operation that could take several hours or days to complete. It will continue running in the background. Do you want to continue?

4. Click **Refresh All** to continue.
-

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

After successful refresh, the refreshed keys are listed on the **Cloud Keys > Azure > Azure Keys** page. Refer to [Viewing Azure Keys](#) for details.

Viewing/Editing Details of Azure Vaults

Viewing Azure Vaults Details

The Vaults tab of the Azure Key Vaults page shows the list of existing Azure key vaults. Search for the key vaults by Vault Name, Subscription ID, or Subscription Name.

To view the details of Azure vaults:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab of the Azure Key Vaults page displays the following details.

Column	Description
Name	Name of the Azure vault.
Last Refreshed	Time when the vault was last refreshed.
Connection	Name of the connection.
Cloud	Cloud name.
Managed HSM	Whether the vault is inside an Azure Managed HSM pool.
Location	Location in which the vault is added.
SKU	Pricing Tier information of the vault.
Subscription Name	Name of the subscription.
Subscription ID	ID of the subscription.
Vault Type	Type of the Azure key vault. The type can be Standard, Premium, or ManagedHSM.
Schedule Name	Name of the key rotation schedule.
Frequency	Frequency of the key rotation.
Next Rotation Date	Date when the next key rotation will happen.
Vault URI	URI of the Azure vault. By default, this column is not visible. Click the Customize View (⚙️) icon, select Vault URI , and click OK to display the column.

Modifying Azure Vault Details

To edit the details of an Azure vault:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab of the Azure Key Vaults page displays the list of added Azure vaults.
3. Click the overflow icon () corresponding to the desired Azure vault and click **View/Edit Details**.

You can modify the user permission and the Azure connection linked with the Azure vaults. Refer to the following sections:

- [Managing User Permissions on Azure Vaults](#)
- [Changing the Azure Connection](#)

Adding/Editing Rotation Schedules

Adding a Rotation Schedule to an Azure Vault

To add a rotation schedule to an Azure vault.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab of the Azure Key Vaults page displays the list of added Azure vaults.
3. Click the overflow icon () corresponding to the desired Azure vault and click **Add/Edit Schedule**. The Add Schedule to Vault wizard is displayed.

Note

If a rotation schedule is not already added to the vault, then the Add Schedule to Vault wizard is displayed.

4. Select the desired **Rotation Schedule** from the drop-down list.
5. Select **Schedule Setting**. You can select either **Ignore Keys With Schedules** or **Override Existing Key Schedules**.
6. Click **Save**.

Alternatively, you can do the following.

1. Click the desired Azure vault from the list of Azure vaults. The Azure Vault Detail screen is displayed.

2. Go to **KEY SCHEDULE** section and select the desired **Rotation Schedule** from the drop-down list.
3. Select **Schedule Setting**. You can select either **Ignore Keys With Schedules** or **Override Existing Key Schedules**.
4. Click **Update**.

Refer to [Scheduling Operations](#) for more details.

Editing an Existing Rotation Schedule of an Azure Vault

To edit an existing rotation schedule of an Azure vault.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab of the Azure Key Vaults page displays the list of added Azure vaults.
3. Click the overflow icon () corresponding to the desired Azure vault and click **Add/Edit Schedule**. The Edit Schedule to Vault wizard is displayed.

Note

If a rotation schedule is already added to the vault, then the Edit Schedule to Vault wizard is displayed.

4. Change **Rotation Schedule** from the drop-down list.
5. Select **Schedule Setting**. You can select either **Ignore Keys With Schedules** or **Override Existing Key Schedules**.
6. Click **Save**.

Alternatively, you can do the following.

1. Click the desired Azure Vault from the list of the added Azure vaults. The Azure Vault Detail screen is displayed.
2. Go to **KEY SCHEDULE** section and change **Rotation Schedule** from the drop-down list.
3. Select **Schedule Setting**. You can select either **Ignore Keys With Schedules** or **Override Existing Key Schedules**.
4. Click **Update**.

Refer to [Scheduling Operations](#) for more details.

Managing User Permissions on Azure Vaults

To work with Azure, users/group must have the minimum set of permissions that allow them to use the Azure resources such as Azure keys, vaults, secrets, certificates, and reports. Initially, the user only has permission to view the keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform operations on the Azure vault. Refer to [User Roles](#) for details.

User permissions for Azure resources can be configured on the **Vault Access Control** section of the Azure vault details page.

To add permission for user/group:

1. In the **Vault Access Control** section, click **Assign User/Group**.
2. On the **Assign User/Group** screen, select the user or group to be assigned permissions from the **User/Group** drop-down list.
3. Click **Save**.

The newly added user/group is displayed under **Name** in the **Vault Access Control** section.

Note

Use the Keys, Certificates, Secrets, and Reports tabs to specify permissions for keys, certificates, secrets, and reports. These tabs display the operations allowed on the Azure vaults.

Note

Depending on the type of the Azure vault, a key is deleted differently.

- **Vault with Soft-Delete Enabled:** To delete the key permanently, perform the **Soft Delete** and the **Purge** operations on the key. However, for this operation to be successful, the user must have the **Soft Delete Key** and the **Purge Key** access.
- **Vault with Soft-Delete Disabled:** To delete the key permanently, use the Delete option. However, for this operation to be successful, the user must have the **Soft Delete Key** access.

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. Select the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

Removing a Permission

To remove a permission assigned to a user or group:

1. Clear the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. Under **Unassign**, click the **X** button corresponding to the desired user/group.
2. On the **Unassign User / Unassign Group** screen, click **Unassign**.

Note

Unassigning this user/group will remove all permissions currently assigned to the user/group. Are you sure you want to continue?

3. Click **Unassign**.

A message **Updated access control for this key vault** is displayed on the screen.

Changing the Azure Connection

To change the connection linked with the Azure vault:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab of the Azure Key Vaults page displays the list of added Azure vaults.
3. Click the overflow icon () corresponding to the desired Azure vault and click **View/Edit Details**.

4. Expand the **GENERAL INFO** section.
5. From the **Azure Connection** drop-down list, select the desired connection.
6. Click **Update**.

Deleting Azure Key Vaults

To delete an Azure key vault:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab of the Azure Key Vaults page displays the list of added Azure vaults.
3. Click the overflow icon () corresponding to the desired Azure vault and click **Delete**.
4. On the **Delete Azure Key Vault** screen, select **Delete Azure Key Vault**.

A message **Azure Vault deleted** is displayed on the screen.

Warning

Be extremely careful when deleting an Azure vault. Once the Azure vault is deleted, it will no longer be available for use.

Managing Azure Keys

This section describes how to manage Azure keys on CCKM. Before proceeding, you must have an Azure key vault added to the CCKM. Refer to [Managing Azure Vaults](#) for details.

Adding Azure Keys

CCKM provides two methods to add Azure keys:

- **Creating/Uploading New Key Material:** Add key material by creating and uploading new source key or creating new native key.
- **Cloning Existing Key Material:** Clone key material from an existing key to create a new key.

Creating/Uploading New Key Material

To add an Azure key by creating/uploading new key material:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.

-
3. Click **Add Key**. The Select Material Origin screen of the Add Azure Key wizard is displayed.
 4. Under **Select Method**, select **Create/Upload New Key Material**. The **Select Source** section appears. Depending on your requirements, select from the following:
 - CipherTrust (External)
 - CipherTrust (Local)
 - Microsoft Azure (Native)
 - Luna HSM
 - Vormetric DSM

Uploading CipherTrust (External) Key Material

Upload the key material of an external CipherTrust Manager key using the CipherTrust Manager to configure the source key.

Select Material Origin

1. Select **CipherTrust (External)**.
2. Click **Next**. The Configure Source Key screen is displayed.

Configure CipherTrust (External) Key

1. **Select Domain** from the drop-down list.
 2. Enter a **Key Name**.
 3. Select the **Key Size** from the available options. The supported sizes are 2048, 3072, and 4096.
1. Click **Next**.

Configure Destination (Azure) Key

1. Enter a user-friendly alias as the **Azure Key Name**. This helps uniquely identify a key. By default, the Key Name specified on the previous screen is populated.
2. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types are Standard, Premium, and ManagedHSM.
3. Select the **Key Type**.
 - For a standard vault, the key type is **RSA**.
 - For vaults inside an Azure Managed HSM pool, the key type is **RSA HSM**.
 - For a premium vault, the key type can be **RSA** or **RSA HSM**.

4. (Optional) Set the key activation and expiration dates.

- Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the key.
- Select the **Set Expiration Date** check box and from the on-screen calendar, select the key expiration date and time.

5. (Optional) Select the **Enable Key** check box.6. (Optional, applicable to premium Azure vaults and vaults stored in an Azure Managed HSM pool) Select the **Exportable** check box to enable Confidential Computing.

When you select **Exportable**, the Release Policy text box is displayed. In this text box, specify or paste the release policy in the JSON format. A release policy specifies the policy rules under which the key can be exported. The release policy must be specified when creating the first version of an exportable key.

A sample release policy is displayed below.

```
{
  "anyOf": [
    {
      "allof": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedeus.eus.attest.azure.net/"
    },
    {
      "allof": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedwus.wus.attest.azure.net/"
    }
  ]
}
```



```

    },
    {
      "allof": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedneu.neu.attest.azure.net/"
    },
    {
      "allof": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedweu.weu.attest.azure.net/"
    },
    {
      "allof": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedeus2.eus2.attest.azure.net/"
    }
  ],
  "version": "1.0.0"
}

```

7. Select the **Key Attributes**. The supported attributes are:

- Encrypt, Decrypt, Sign
- Verify, Wrap Key, Unwrap Key

8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; . ' " _ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. Select **Rotation Schedule**.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.
 - **Native (Azure)**: Native Microsoft Azure application. Also select the Key Type - **RSA** or **EC**.
 - **Luna**: HSM Luna. Also select a **Partition**.
 - **DSM**: Vormetric Data Security Manager. Also select a **Domain**.
3. Select a **Key Size**.

Note

When the key origin is native Azure, select a curve for Elliptic Curve or Elliptic Curve HSM key types.

4. (Optional) Select the **Enabled** check box to enable the key.

-
5. Click **Next**.

The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Azure Key wizard is closed.

The newly created key is displayed in the list of Azure keys. The origin of the key is **CCKM**.

Uploading CipherTrust (Local) Key Material

Upload the local key material using CipherTrust to configure source key.

Select Material Origin

1. Select **CipherTrust (Local)**.
2. Click **Next**. The Configure Source Key screen is displayed.

Configure CipherTrust (Local) Key

1. Enter a **Key Name**.
2. Select the **Key Size** from the available options. The supported sizes are 2048, 3072, and 4096.
3. Click **Next**.

Configure Destination (Azure) Key

1. Enter a user-friendly alias as the **Azure Key Name**. This helps uniquely identify a key. By default, the Key Name specified on the previous screen is populated.
 2. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types
-

specified in parenthesis. The Azure key vault types are Standard, Premium, and ManagedHSM.

3. Select the **Key Type**.

- For a standard vault, the key type is **RSA**.
- For vaults inside an Azure Managed HSM pool, the key type is **RSA HSM**.
- For a premium vault, the key type can be **RSA** or **RSA HSM**.

4. (Optional) Set the key activation and expiration dates.

- Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the key.
- Select the **Set Expiration Date** check box and from the on-screen calendar, select the key expiration date and time.

5. (Optional) Select the **Enable Key** check box.

6. (Optional, applicable to premium Azure vaults and vaults stored in an Azure Managed HSM pool) Select the **Exportable** check box to enable Confidential Computing.

When you select **Exportable**, the Release Policy text box is displayed. In this text box, specify or paste the release policy in the JSON format. A release policy specifies the policy rules under which the key can be exported. The release policy must be specified when creating the first version of an exportable key.

A sample release policy is displayed below.

```
{
  "anyOf": [
    {
      "allOf": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedeus.eus.attest.azure.net/"
    },
    {
      "allOf": [
        {
```

```

        "claim": "x-ms-attestation-type",
        "equals": "sevsnpvm"
      },
      {
        "claim": "x-ms-compliance-status",
        "equals": "azure-compliant-cvm"
      }
    ],
    "authority": "https://sharedwus.wus.attest.azure.net/"
  },
  {
    "allof": [
      {
        "claim": "x-ms-attestation-type",
        "equals": "sevsnpvm"
      },
      {
        "claim": "x-ms-compliance-status",
        "equals": "azure-compliant-cvm"
      }
    ],
    "authority": "https://sharedneu.neu.attest.azure.net/"
  },
  {
    "allof": [
      {
        "claim": "x-ms-attestation-type",
        "equals": "sevsnpvm"
      },
      {
        "claim": "x-ms-compliance-status",
        "equals": "azure-compliant-cvm"
      }
    ],
    "authority": "https://sharedweu.weu.attest.azure.net/"
  },
  {
    "allof": [
      {
        "claim": "x-ms-attestation-type",
        "equals": "sevsnpvm"
      },
      {
        "claim": "x-ms-compliance-status",

```

```

        "equals": "azure-compliant-cvm"
      }
    ],
    "authority": "https://sharedeus2.eus2.attest.azure.net/"
  }
],
"version": "1.0.0"
}

```

7. Select the **Key Attributes**. The supported attributes are:

- Encrypt, Decrypt, Sign
- Verify, Wrap Key, Unwrap Key

8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; : ' " _ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**.

The Add to Schedule screen is displayed.

Add to Schedule

1. Select **Rotation Schedule**.

2. Select the **Key Origin** from the available options. The key origin can be:

- **CipherTrust**: CipherTrust Manager.
- **Native (Azure)**: Native Microsoft Azure application. Also select the Key Type - **RSA** or **EC**.
- **Luna**: HSM Luna. Also select a **Partition**.
- **DSM**: Vormetric Data Security Manager. Also select a **Domain**.

3. Select a **Key Size**.

Note

When the key origin is native Azure, select a curve for Elliptic Curve or Elliptic Curve HSM key types.

4. (Optional) Select the **Enabled** check box to enable the key.

5. Click **Next**.

The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY, DESTINATION KEY and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Azure Key wizard is closed.

The newly created key is displayed in the list of Azure keys. The origin of the key is **CCKM**.

Creating Microsoft Azure (Native) Key Material

Create the Azure key material directly using the native Microsoft Azure application.

Select Material Origin

1. Select **Microsoft Azure (Native)**.

2. Click **Next**. The Configure Source Key screen is displayed.

Configure Source Key

-
1. Enter a user-friendly alias as the **Azure Key Name**. This helps uniquely identify a key.
 2. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types are Standard, Premium, and ManagedHSM.
 3. Select the **Key Type**.
 - For a standard vault, the key type can be **RSA** or **Elliptic Curve**.
 - For vaults inside an Azure Managed HSM pool, the key type can be **RSA HSM** or **Elliptic Curve HSM**.
 - For a premium vault, the key type can be **Elliptic Curve**, **Elliptic Curve HSM**, **RSA**, or **RSA HSM**.
 4. (Applicable to **RSA** or **RSA HSM** key types) Select **Size** from the following options: 2048, 3072, and 4096.
 5. (Applicable to **Elliptic Curve** or **Elliptic Curve HSM** key types) Select **Curve** from the following options: P-256, P-384, P-521, and SECP256K1.
 6. (Optional) Set the key activation and expiration dates.
 - Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the key.
 - Select the **Set Expiration Date** check box and from the on-screen calendar, select the key expiration date and time.
 7. (Optional) Select the **Enable Key** check box.
 8. Select the **Key Attributes**.

If the Key Type is **RSA** or **RSA HSM**, the supported attributes are:

- Encrypt, Decrypt, Sign
- Verify, Wrap Key, Unwrap Key

If the Key Type is **Elliptic Curve** or **Elliptic Curve HSM**, the supported attributes are:

- Sign
- Verify

9. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:

- Alphanumeric characters
- Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; : ' " _ **`

c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

10. Click **Next**.

The Add to Schedule screen is displayed.

Add to Schedule

1. Select **Rotation Schedule**.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.
 - **Native (Azure)**: Native Microsoft Azure application. Also select the Key Type - **RSA** or **EC**.
 - **Luna**: HSM Luna. Also select a **Partition**.
 - **DSM**: Vormetric Data Security Manager. Also select a **Domain**.
3. Select a **Key Size**.

Note

When the key origin is native Azure, select a curve for Elliptic Curve or Elliptic Curve HSM key types.

4. (Optional) Select the **Enabled** check box to enable the key.
5. Click **Next**.

The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, NATIVE KEY and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **NATIVE KEY** and **KEY SCHEDULES** section and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the **NATIVE KEY** and **KEY SCHEDULES** sections becomes **Complete** and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Azure Key wizard is closed.

The newly created key is displayed in the list of Azure keys. The origin of the key is **Native**.

Uploading Luna HSM Key Material

Upload the key material using Luna HSM to configure source key.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Select Material Origin

1. Select **Luna HSM**.
2. Click **Next**. The Configure Source Key screen is displayed.

Configure Source Key

1. Enter a **Luna HSM Key Label**.
2. Select the **Partition ID** of the desired Luna HSM.
3. Select the key **Mechanism**. The supported key mechanisms are:
 - CKM_RSA_FIPS_186_3_AUX_PRIME_KEY_PAIR_GEN
 - CKM_RSA_X9_31_KEY_PAIR_GEN
 - CKM_RSA_FIPS_186_3_PRIME_KEY_PAIR_GEN
 - CKM_RSA_PKCS_KEY_PAIR_GEN
4. Select the **Key Size** from the available options. The supported sizes are 2048, 3072, and 4096.
5. Select the **Key Attributes**. The options are:
 - Encrypt, Decrypt, Wrap, Unwrap

- Sign, Verify, Derive

6. Click **Next**.

Configure Destination (Azure) Key

1. Enter a user-friendly alias as the **Azure Key Name**. This helps uniquely identify a key. By default, the Key Name specified on the previous screen is populated.
2. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types are Standard, Premium, and ManagedHSM.
3. Select the **Key Type**.
 - For a standard vault, the key type is **RSA**.
 - For a premium vault, the key type can be **RSA** or **RSA HSM**.
4. (Optional) Set the key activation and expiration dates.
 - Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the key.
 - Select the **Set Expiration Date** check box and from the on-screen calendar, select the key expiration date and time.
5. (Optional) Select the **Enable Key** check box.
6. (Optional, applicable to premium Azure vaults and vaults stored in an Azure Managed HSM pool) Select the **Exportable** check box to enable Confidential Computing.

When you select **Exportable**, the Release Policy text box is displayed. In this text box, specify or paste the release policy in the JSON format. A release policy specifies the policy rules under which the key can be exported. The release policy must be specified when creating the first version of an exportable key.

A sample release policy is displayed below.

```
{
  "anyOf": [
    {
      "allof": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ]
    }
  ]
}
```

```

        }
    ],
    "authority": "https://sharedeus.eus.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedwus.wus.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedneu.neu.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedweu.weu.attest.azure.net/"
},
{

```

```

        "allOf": [
            {
                "claim": "x-ms-attestation-type",
                "equals": "sevsnpvm"
            },
            {
                "claim": "x-ms-compliance-status",
                "equals": "azure-compliant-cvm"
            }
        ],
        "authority": "https://sharedeus2.eus2.attest.azure.net/"
    },
    "version": "1.0.0"
}

```

7. Select the **Key Attributes**. The supported attributes are:

- Encrypt, Decrypt, Sign
- Verify, Wrap Key, Unwrap Key

8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `**!@#$%)({}><?+ - / \ [ ] ^ & + = | ~ ` , ; : ' " _ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. Select **Rotation Schedule**.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.

- **Native (Azure):** Native Microsoft Azure application. Also select the Key Type - **RSA** or **EC**.
- **Luna:** HSM Luna. Also select a **Partition**.
- **DSM:** Vormetric Data Security Manager. Also select a **Domain**.

3. Select a **Key Size**.

Note

When the key origin is native Azure, select a curve for Elliptic Curve or Elliptic Curve HSM key types.

4. (Optional) Select the **Enabled** check box to enable the key.

5. Click **Next**.

The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY** and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Azure Key wizard is closed.

The newly created key is displayed in the list of Azure keys. The origin of the key is **CCKM**.

Uploading Vormetric DSM Key Material

Upload the key material using Vormetric DSM to configure source key.

Select Material Origin

-
1. Select **Vormetric DSM**.
 2. Click **Next**. The Configure Source Key screen is displayed.

Configure Source Key

1. Enter a **DSM Key Name**.
2. (Optional) Provide a basic **Description** of the key.
3. Select the desired **DSM Domain**.
4. (Optional) Set the key expiration date. Select the **Expiration Date** check box and from the on-screen calendar, select the key expiration date and time.
5. Select the key **Algorithm**. The supported algorithms are:
 - RSA-4048
 - RSA-3072
 - RSA-4096
6. Click **Next**.

Configure Destination (Azure) Key

1. Enter a user-friendly alias as the **Azure Key Name**. This helps uniquely identify a key. By default, the Key Name specified on the previous screen is populated.
 2. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types are Standard, Premium, and ManagedHSM.
 3. Select the **Key Type**.
 - For a standard vault, the key type is **RSA**.
 - For a premium vault, the key type can be **RSA** or **RSA HSM**.
 4. (Optional) Set the key activation and expiration dates.
 - Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the key.
 - Select the **Set Expiration Date** check box and from the on-screen calendar, select the key expiration date and time.
 5. (Optional) Select the **Enable Key** check box.
 6. (Optional, applicable to premium Azure vaults and vaults stored in an Azure Managed HSM pool) Select the **Exportable** check box to enable Confidential Computing.
-

When you select **Exportable**, the Release Policy text box is displayed. In this text box, specify or paste the release policy in the JSON format. A release policy specifies the policy rules under which the key can be exported. The release policy must be specified when creating the first version of an exportable key.

A sample release policy is displayed below.

```
{
  "anyOf": [
    {
      "allOf": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedeus.eus.attest.azure.net/"
    },
    {
      "allOf": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedwus.wus.attest.azure.net/"
    },
    {
      "allOf": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
    }
  ]
}
```



```

        "authority": "https://sharedneu.neu.attest.azure.net/"
    },
    {
        "allof": [
            {
                "claim": "x-ms-attestation-type",
                "equals": "sevsnpvm"
            },
            {
                "claim": "x-ms-compliance-status",
                "equals": "azure-compliant-cvm"
            }
        ],
        "authority": "https://sharedweu.weu.attest.azure.net/"
    },
    {
        "allof": [
            {
                "claim": "x-ms-attestation-type",
                "equals": "sevsnpvm"
            },
            {
                "claim": "x-ms-compliance-status",
                "equals": "azure-compliant-cvm"
            }
        ],
        "authority": "https://sharedeus2.eus2.attest.azure.net/"
    }
],
"version": "1.0.0"
}

```

7. Select the **Key Attributes**. The supported attributes are:

- Encrypt, Decrypt, Sign
- Verify, Wrap Key, Unwrap Key

8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.

- CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; : ' " _ **`

c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. Select **Rotation Schedule**.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.
 - **Native (Azure)**: Native Microsoft Azure application. Also select the Key Type - **RSA** or **EC**.
 - **Luna**: HSM Luna. Also select a **Partition**.
 - **DSM**: Vormetric Data Security Manager. Also select a **Domain**.
3. Select a **Key Size**.

Note

When the key origin is native Azure, select a curve for Elliptic Curve or Elliptic Curve HSM key types.

4. (Optional) Select the **Enabled** check box to enable the key.

5. Click **Next**.

The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as

appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Azure Key wizard is closed.

The newly created key is displayed in the list of Azure keys. The origin of the key is **CCKM**.

Cloning Existing Key Material

To add a new Azure key by cloning existing key material:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click **Add Key**. The Select Material Origin screen of the Add Azure Key wizard is displayed.
4. Under **Select Method**, select **Clone Existing Key Material**. The **Select Source** section appears. Depending on your requirements, select from the following:
 - CipherTrust (External)
 - CipherTrust (Local)
 - Luna HSM
 - Vormetric DSM

Cloning CipherTrust (External) Key Material

Upload the key material of an external CipherTrust Manager key using CipherTrust to configure source key.

Select Material Origin

1. Select **CipherTrust (External)**.
2. Click **Next**. The Select CipherTrust Key screen is displayed.

Select CipherTrust (External) Key

1. Select a **Domain** from the drop-down list. This field displays the domains of the external CipherTrust Manager.
2. Select the desired **Key Size**. The options are 2048, 3072, and 4096.

3. Select a **CipherTrust (External) Key** from the list. The list displays the keys of the selected domain of the external CipherTrust Manager.
4. Click **Next**. The Configure Destination (Azure) Key screen is displayed.

Configure Destination (Azure) Key

1. Enter a user-friendly alias as the **Azure Key Name**. This helps uniquely identify a key.
2. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types are Standard, Premium, and ManagedHSM.
3. Select the **Key Type**.
 - For a standard vault, the key type is **RSA**.
 - For a premium vault, the key type can be **RSA** or **RSA HSM**.
4. (Optional) Set the key activation and expiration dates.
 - Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the key.
 - Select the **Set Expiration Date** check box and from the on-screen calendar, select the key expiration date and time.
5. (Optional) Select the **Enable Key** check box.
6. (Optional, applicable to premium Azure vaults and vaults stored in an Azure Managed HSM pool) Select the **Exportable** check box to enable Confidential Computing.

When you select **Exportable**, the Release Policy text box is displayed. In this text box, specify or paste the release policy in the JSON format. A release policy specifies the policy rules under which the key can be exported. The release policy must be specified when creating the first version of an exportable key.

A sample release policy is displayed below.

```
{
  "anyOf": [
    {
      "allOf": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ]
    }
  ]
}
```

```

        }
    ],
    "authority": "https://sharedeus.eus.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedwus.wus.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedneu.neu.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedweu.weu.attest.azure.net/"
},
{

```

```

        "allOf": [
            {
                "claim": "x-ms-attestation-type",
                "equals": "sevsnpvm"
            },
            {
                "claim": "x-ms-compliance-status",
                "equals": "azure-compliant-cvm"
            }
        ],
        "authority": "https://sharedeus2.eus2.attest.azure.net/"
    },
    "version": "1.0.0"
}

```

7. Select the **Key Attributes**. The supported attributes are:

- Encrypt, Decrypt, Sign
- Verify, Wrap Key, Unwrap Key

8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `**!@#$%)( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; : ' " _ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. Select **Rotation Schedule**.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.

- **Native (Azure):** Native Microsoft Azure application. Also select the Key Type - **RSA** or **EC**.
- **Luna:** HSM Luna. Also select a **Partition**.
- **DSM:** Vormetric Data Security Manager. Also select a **Domain**.

3. Select a **Key Size**.

Note

When the key origin is native Azure, select a curve for Elliptic Curve or Elliptic Curve HSM key types.

4. (Optional) Select the **Enabled** check box to enable the key.

5. Click **Next**.

The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY and KEY SCHEDULES section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Azure Key wizard is closed.

The newly created key is displayed in the list of Azure keys. The origin of the key is CCKM.

Cloning CipherTrust (Local) Key Material

Upload the local key material using CipherTrust to configure source key.

Select Material Origin

1. Select **CipherTrust (Local)**.
2. Click **Next**. The Select CipherTrust Key screen is displayed.

Select CipherTrust (Local) Key

1. Select the desired **Key Size**. The options are 2048, 3072, and 4096.
2. Select a **CipherTrust (Local) Key** from the list. The list displays the keys of the selected domain of the local CipherTrust Manager.
3. Click **Next**. The Configure Destination (Azure) Key screen is displayed.

Configure Destination (Azure) Key

1. Enter a user-friendly alias as the **Azure Key Name**. This helps uniquely identify a key.
2. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types are Standard, Premium, and ManagedHSM.
3. Select the **Key Type**.
 - For a standard vault, the key type is **RSA**.
 - For a premium vault, the key type can be **RSA** or **RSA HSM**.
4. (Optional) Set the key activation and expiration dates.
 - Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the key.
 - Select the **Set Expiration Date** check box and from the on-screen calendar, select the key expiration date and time.
5. (Optional) Select the **Enable Key** check box.
6. (Optional, applicable to premium Azure vaults and vaults stored in an Azure Managed HSM pool) Select the **Exportable** check box to enable Confidential Computing.

When you select **Exportable**, the Release Policy text box is displayed. In this text box, specify or paste the release policy in the JSON format. A release policy specifies the policy rules under which the key can be exported. The release policy must be specified when creating the first version of an exportable key.

A sample release policy is displayed below.

```
{
  "anyOf": [
    {
      "allOf": [
```



```

        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedeus.eus.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedwus.wus.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedneu.neu.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {

```

```

        "claim": "x-ms-compliance-status",
        "equals": "azure-compliant-cvm"
      },
    ],
    "authority": "https://sharedweu.weu.attest.azure.net/"
  },
  {
    "allof": [
      {
        "claim": "x-ms-attestation-type",
        "equals": "sevsnpvm"
      },
      {
        "claim": "x-ms-compliance-status",
        "equals": "azure-compliant-cvm"
      }
    ],
    "authority": "https://sharedeus2.eus2.attest.azure.net/"
  }
],
"version": "1.0.0"
}

```

7. Select the **Key Attributes**. The supported attributes are:

- Encrypt, Decrypt, Sign
- Verify, Wrap Key, Unwrap Key

8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; : . ' " _ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. Select **Rotation Schedule**.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.
 - **Native (Azure)**: Native Microsoft Azure application. Also select the Key Type - **RSA** or **EC**.
 - **Luna**: HSM Luna. Also select a **Partition**.
 - **DSM**: Vormetric Data Security Manager. Also select a **Domain**.
3. Select a **Key Size**.

Note

When the key origin is native Azure, select a curve for Elliptic Curve or Elliptic Curve HSM key types.

4. (Optional) Select the **Enabled** check box to enable the key.
5. Click **Next**.

The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY and KEY SCHEDULES section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Azure Key wizard is closed.

The newly created key is displayed in the list of Azure keys. The origin of the key is **CCKM**.

Cloning Luna HSM Key Material

Upload the key material using Luna HSM to configure source key.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Select Material Origin

1. Select **Luna HSM**.
2. Click **Next**. The Select Luna HSM Key screen is displayed.

Select Luna HSM Key

1. Select the desired **Key Size**. The options are 2048, 3072, and 4096.
2. Select an **HSM Key** from the list. The list displays the keys of the selected domain of the external CipherTrust Manager.
3. Click **Next**. The Configure Destination (Azure) Key screen is displayed.

Configure Destination (Azure) Key

1. Enter a user-friendly alias as the **Azure Key Name**. This helps uniquely identify a key.
2. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types are Standard, Premium, and ManagedHSM.
3. Select the **Key Type**.
 - For a standard vault, the key type is **RSA**.
 - For a premium vault, the key type can be **RSA** or **RSA HSM**.
4. (Optional) Set the key activation and expiration dates.
 - Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the key.
 - Select the **Set Expiration Date** check box and from the on-screen calendar, select the key expiration date and time.

5. (Optional) Select the **Enable Key** check box.
6. (Optional, applicable to premium Azure vaults and vaults stored in an Azure Managed HSM pool) Select the **Exportable** check box to enable Confidential Computing.

When you select **Exportable**, the Release Policy text box is displayed. In this text box, specify or paste the release policy in the JSON format. A release policy specifies the policy rules under which the key can be exported. The release policy must be specified when creating the first version of an exportable key.

A sample release policy is displayed below.

```
{
  "anyOf": [
    {
      "allOf": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedeus.eus.attest.azure.net/"
    },
    {
      "allOf": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedwus.wus.attest.azure.net/"
    },
    {
      "allOf": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        }
      ],

```

```

        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedneu.neu.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedweu.weu.attest.azure.net/"
},
{
    "allof": [
        {
            "claim": "x-ms-attestation-type",
            "equals": "sevsnpvm"
        },
        {
            "claim": "x-ms-compliance-status",
            "equals": "azure-compliant-cvm"
        }
    ],
    "authority": "https://sharedeus2.eus2.attest.azure.net/"
}
],
"version": "1.0.0"
}

```

7. Select the **Key Attributes**. The supported attributes are:

- Encrypt, Decrypt, Sign
- Verify, Wrap Key, Unwrap Key

8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; : ' " _ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**.

The Add to Schedule screen is displayed.

Add to Schedule

1. Select **Rotation Schedule**.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.
 - **Native (Azure)**: Native Microsoft Azure application. Also select the Key Type - **RSA** or **EC**.
 - **Luna**: HSM Luna. Also select a **Partition**.
 - **DSM**: Vormetric Data Security Manager. Also select a **Domain**.
3. Select a **Key Size**.

Note

When the key origin is native Azure, select a curve for Elliptic Curve or Elliptic Curve HSM key types.

4. (Optional) Select the **Enabled** check box to enable the key.
5. Click **Next**.

The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN,

SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY and KEY SCHEDULES sections becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Azure Key wizard is closed.

The newly created key is displayed in the list of Azure keys. The origin of the key is **CCKM**.

Cloning Vormetric DSM Key Material

Upload the key material using Vormetric DSM to configure source key.

Select Material Origin

1. Select **Vormetric DSM**.
2. Click **Next**. The Select Vormetric DSM Key screen is displayed.

Select Vormetric DSM Key

1. Select the desired **Key Size**. The options are 2048, 3072, and 4096.
2. Select an **DSM Key** from the list. The list displays the keys of the selected domain of the external CipherTrust Manager.
3. Click **Next**. The Configure Destination (Azure) Key screen is displayed.

Configure Destination (Azure) Key

1. Enter a user-friendly alias as the **Azure Key Name**. This helps uniquely identify a key.
2. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types are Standard, Premium, and ManagedHSM.
3. Select the **Key Type**.
 - For a standard vault, the key type is **RSA**.

- For a premium vault, the key type can be **RSA** or **RSA HSM**.
4. (Optional) Set the key activation and expiration dates.
 - Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the key.
 - Select the **Set Expiration Date** check box and from the on-screen calendar, select the key expiration date and time.
 5. (Optional) Select the **Enable Key** check box.
 6. (Optional, applicable to premium Azure vaults and vaults stored in an Azure Managed HSM pool) Select the **Exportable** check box to enable Confidential Computing.

When you select **Exportable**, the Release Policy text box is displayed. In this text box, specify or paste the release policy in the JSON format. A release policy specifies the policy rules under which the key can be exported. The release policy must be specified when creating the first version of an exportable key.

A sample release policy is displayed below.

```
{
  "anyOf": [
    {
      "allof": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ],
      "authority": "https://sharedeus.eus.attest.azure.net/"
    },
    {
      "allof": [
        {
          "claim": "x-ms-attestation-type",
          "equals": "sevsnpvm"
        },
        {
          "claim": "x-ms-compliance-status",
          "equals": "azure-compliant-cvm"
        }
      ]
    }
  ]
}
```

```

    ],
    "authority": "https://sharedwus.wus.attest.azure.net/"
  },
  {
    "allof": [
      {
        "claim": "x-ms-attestation-type",
        "equals": "sevsnpvm"
      },
      {
        "claim": "x-ms-compliance-status",
        "equals": "azure-compliant-cvm"
      }
    ],
    "authority": "https://sharedneu.neu.attest.azure.net/"
  },
  {
    "allof": [
      {
        "claim": "x-ms-attestation-type",
        "equals": "sevsnpvm"
      },
      {
        "claim": "x-ms-compliance-status",
        "equals": "azure-compliant-cvm"
      }
    ],
    "authority": "https://sharedweu.weu.attest.azure.net/"
  },
  {
    "allof": [
      {
        "claim": "x-ms-attestation-type",
        "equals": "sevsnpvm"
      },
      {
        "claim": "x-ms-compliance-status",
        "equals": "azure-compliant-cvm"
      }
    ],
    "authority": "https://sharedeus2.eus2.attest.azure.net/"
  }
],
"version": "1.0.0"

```

}

7. Select the **Key Attributes**. The supported attributes are:

- Encrypt, Decrypt, Sign
- Verify, Wrap Key, Unwrap Key

8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; : ' " _ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. Select **Rotation Schedule**.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust**: CipherTrust Manager.
 - **Native (Azure)**: Native Microsoft Azure application. Also select the Key Type - **RSA** or **EC**.
 - **Luna**: HSM Luna. Also select a **Partition**.
 - **DSM**: Vormetric Data Security Manager. Also select a **Domain**.
3. Select a **Key Size**.

Note

When the key origin is native Azure, select a curve for Elliptic Curve or Elliptic Curve HSM key types.

4. (Optional) Select the **Enabled** check box to enable the key.
5. Click **Next**.

The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Azure Key wizard is closed.

The newly created key is displayed in the list of Azure keys. The origin of the key is **CCKM**.

Viewing Azure Keys

Search for Azure keys by Key Name, Key Vault, or Tags.

CCKM does not allow searching for keys:

- By tag values using colon (:)
 - By "key:value" pair using these characters:

\ , : " %

To view an Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed. The Azure Keys page displays the following details:

Field	Description
Key Name	Unique, user-friendly alias of the key. This is useful

Field	Description
	in searching for specific keys.
Current Version	Current version of the key. Click the expand icon () corresponding to a key to view its versions.
Key Vault	Name of the Azure key vault.
Key State	State of the key. The status can be Enabled or Disabled. Click the filter icon () to view the list of Enabled or Disabled keys.
Region	Azure region where the key is created. Click the filter icon () to view the list of supported Azure regions.
Status	State of the key. The status can be: • Available • Soft Deleted • Deleted
Algorithm	Name of the algorithm. Supported algorithms are: • EC • RSA • EC-HSM • RSA HSM
Origin	Source of the key material. The origin of the key can be: • CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud. • External (Unknown): Source of the key material is unknown. It is different than CCKM and the native cloud.
Cloud	Name of the cloud. Supported clouds are: • Azure Cloud • Azure China Cloud • Azure German Cloud • Azure US Government • Azure Stack
Backup	State of the backup of the key. If the backup of the key exists, the backup icon becomes visible.
Creation Date	Time when the key is created.

Sometimes, you might notice certain keys are displayed as grayed out. This happens when the keys are no longer accessible. For example, when:

- Any cloud permissions on the keys are changed. The keys are no longer accessible from the Azure connection.
- Connection is changed in KMS. The new connection does not have permissions to access the keys.

Editing Azure Keys

To view or edit an Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **View/Edit**.
4. Configure the **KEY SCHEDULES** and manage the **BACKUPS**. Refer to [KEY SCHEDULES](#) and [BACKUPS](#) for details.

KEY SCHEDULES

To configure the key schedules, go to the KEY SCHEDULES section and select/enter the following details:

1. From the **Select Rotation Schedule** drop-down list, select a rotation schedule.
2. Select the key origin. The available options are CipherTrust (External), CipherTrust (Local), Native (Azure), Luna, and DSM.

Click the desired tab to view the instructions.

CipherTrust (Local)

- a. Select the **Key Type**. The available options are RSA and RSA-HSM.
- b. Select the **Key Size**. The available options are 2048, 3072, and 4096.

Native (Azure)

- a. Select the **Key Type**. The available options are RSA, EC, RSA-HSM, and EC-HSM.
- b. select the **Elliptical Curve Name**. The available options are P-56, P-384, P-521, and SECP256K1.

CipherTrust (External)

- a. Select the **Key Type**. The available options are RSA and RSA-HSM.
- b. Select the **Key Size**. The available options are 2048, 3072, and 4096.

Luna

- a. Select the Partition.

-
- b. Select the **Key Type**. The available options are RSA and RSA-HSM.
 - c. Select the **Key Size**. The available options are 2048, 3072, and 4096.

DSM

- a. Select the Domain.
 - b. Select the **Key Type**. The available options are RSA and RSA-HSM.
 - c. Select the **Key Size**. The available options are 2048, 3072, and 4096.
3. Select **Enabled** if you want to enable the rotated key.
 4. Click **Update**.

A message **Key schedule updated successfully** is displayed on the screen.

To select the Key backup schedule.

1. From the **Select Backup Schedule** drop-down list, select a backup schedule.
2. Click **Update**.

BACKUPS

This section allows you to perform the following operations.

- View all the backups of the Azure key.
- Restore a service backup.
- Retrieve the latest service backup date
- Backup the Azure key.
- Restore the backups of the Azure key.
- Delete the backups of the Azure key.
- Delete multiple backups of the Azure key.
- Delete multiple backups of the Azure key by date.

To restore the service backup.

1. Click the overflow icon () corresponding to the Service Backup and click **Restore Backup**. The Restore Backup dialog box is displayed.
2. Click **Restore Backup**.

To view the latest backup date of the Service, click the **Refresh** icon to the left the Service Backup.

To backup the Azure key.

1. Click **Backup Now**. The Backup Now dialog box is displayed.
2. (Optional) Enter the **Backup Name**.
3. (Optional) Enter the **Description**.

Note

- The CCKM Users should have Add Key or Upload Key permissions. To add the permission in the vault, refer to [Managing User Permissions on Azure Vaults](#).
- Backup Now is the manual creation of a cloud key's backup.
- **Backup Limit:**
 - The default backup limit is 30.
 - You can change the limit at the domain level using the [Updating CCKM Settings API](#). The limit will be applicable to all the vaults in the domain.
 - You can change the limit at the vault level using the [Adding Azure Vaults](#) or [Updating Azure Vaults](#) APIs. The limit will apply to that particular vault.
 - If you have changed the limit at both levels, the changes made at the vault level will be applicable.
 - If the total Scheduled + Manual Backups exceed the maximum allowance, older backups will automatically be deleted.

To restore the backup of the Azure key.

4. Click the overflow icon () corresponding to the desired backup and click **Restore Backup**. The Restore Backup dialog box is displayed.
5. Click **Restore Backup**.

To delete the backup of the Azure key.

1. Click the overflow icon () corresponding to the desired backup and click **Delete Backup**. The Delete Backup dialog box is displayed.
2. Click **Delete Backup**.

To delete multiple backups of the Azure key.

1. Select the desired backups from the list to delete.

Note

If you want to delete all the backups, select **All**.

2. Click **Delete Selected**. The Delete Backups dialog box is displayed.
3. Click **Delete Backup**.

To delete multiple backups of the Azure key by date.

1. Click **Delete Backup by Date**. The Delete Backup by Date dialog box is displayed.
2. Select the date range, **From Date** and **To Date**.
3. Click **Delete Backups**.

Refreshing Azure Keys

Refreshing is the process of downloading keys created on the Azure key vault to CCKM.

Note

If the key vault of the key has been deleted from the Azure Portal and refreshed on CCKM, the key cannot be refreshed and restored. To refresh and restore the key, create a new key vault on the Azure Portal, in the same region with the same name, and refresh it on CCKM.

Refreshing All Keys

Keys from all key vaults can be refreshed at once. To refresh all the Azure keys:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Keys** tab. The list of available Azure keys is displayed.
4. Click **Refresh**. The **This may take a while...** message is displayed.

Note

Refresh is a time intensive operation that could take several hours or days to complete. It will continue

running in the background. Do you want to continue?

5. Click **Refresh** to continue.

A message **Refresh started...** is displayed on the screen. The refreshed keys are listed on the **Cloud Keys > Azure > Keys** tab.

To cancel the refresh:

1. Click **Cancel Refresh**. The **Cancel Refresh?** message is displayed. The action will terminate all currently active refresh operations. All progress will be lost and this action cannot be undone. Do you want to proceed to cancel refresh or cancel this action?
2. Click **Cancel Refresh**.

A message **Canceled Refresh** is displayed on the screen.

Refreshing a Desired Key

You can refresh a desired key from the list. To refresh an Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the overflow icon () corresponding to the desired alias and click **Refresh**.

A message **Key refreshed successfully.** is displayed on the screen.

Disabling Keys and Versions

You can disable the Azure keys and key versions.

Disabling Keys

If required, you can disable an enabled key. A disabled key cannot operate on data. To disable an Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the overflow icon () corresponding to the desired key and click **Disable**. The Confirm Disable Key dialog box is displayed.
4. Click **Disable**.

A confirmation message is displayed on the screen. The Key State changes to **Disabled**.

Disabling Key Versions

If required, you can disable an enabled version of a key. A disabled version cannot operate on data. To disable an Azure key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the expand icon () to the left of the desired key.
4. Click the overflow icon () corresponding to the desired version.
5. Click **Disable**. The Confirm Disable Version dialog box is displayed.
6. Click **Disable**.

A confirmation message is displayed on the screen. The Key State changes to **Disabled**.

Enabling Keys and Versions

You can enable the Azure keys and key versions.

Enabling Keys

If required, you can enable a disabled key. To enable an Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the overflow icon () corresponding to the desired key and click **Enable**. The Confirm Enable Key dialog box is displayed.
4. Click **Enable**.

A confirmation message is displayed on the screen. The Key State changes to **Enabled**.

Enabling Key Versions

If required, you can enable a disabled version of a key. To enable an Azure key version:

1. Open the **Cloud Key Manager** application.
 2. In the left pane, click **Cloud Keys > Azure**.
 3. Click the expand icon () to the left of the desired key.
 4. Click the overflow icon () corresponding to the desired version.
 5. Click **Enable**. The Confirm Enable Version dialog box is displayed.
-

6. Click **Enable**.

A confirmation message is displayed on the screen. The Key State changes to **Enabled**.

Rotating Keys (Add Version)

Note

To rotate Azure keys, **CCKM Users** require **Add Key** and **Upload Key** permissions.

To rotate a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Rotate Now (Add Version)**. The Add New Version screen is displayed.
4. Select Key Material Origin. Depending on your requirements, select an appropriate option. Refer to the following sections for details:
 - [Creating/Uploading New Key Material](#)
 - [Cloning Existing Key Material](#)

A message **Azure Key successfully rotated** is displayed on the screen. Navigate to **Cloud Keys > Azure > View/Edit > Versions** to view the versions of the rotated Azure key.

Deleting Azure Keys

Non-soft-delete keys can be deleted directly from the Azure vaults using CCKM.

To delete an Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Delete**. The Confirm Delete dialog box is displayed.
4. Click **Delete**.

A confirmation message is displayed on the screen. The key status changes to **DELETED**.

Soft-Deleting Azure Keys

Soft deleting is the process of deleting Azure keys from the Azure vaults and CCKM. These keys still exist on CCKM and in the Azure vaults. The soft-deleted keys can be recovered.

Note

This operation can be performed only on the Azure keys residing in the soft-enabled key vaults.

To soft-delete an Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Soft Delete**. The Confirm Soft Delete dialog box is displayed.
4. Click **Soft Delete**.

A message **key <key name> soft-deleted** is displayed on the screen. The status of the key changes to **SOFT-DELETED**.

Recovering Soft-Deleted Azure Keys

If needed, you can recover a soft-deleted key.

To recover a soft-deleted Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Recover Soft Delete**. The Confirm Recover Key dialog box is displayed.
4. Click **Recover Key**.

A message **Key <key name> recovered from soft-delete.** is displayed on the screen. The status of the key changes to **AVAILABLE**.

Purging Azure Keys

Purging is the process of permanently deleting soft-deleted Azure keys from the Azure vaults. However, backup of the purged key can be restored on CCKM. If you wish to restore backup of the purged key, follow the steps mentioned in the [Restoring Backup](#) section.

Note

This operation can be performed only on the soft-deleted Azure keys residing in the soft-enabled key vaults.

To purge an Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Purge**. The Purge Azure Key dialog box is displayed.
4. Select the **I wish to purge this key.** check box.
5. Click **Purge Key**.

Purging a key might take some time. After successful deletion, a message **key <key name> hard deleted** is displayed on the screen. The status of the key changes to **DELETED**.

If needed, you can restore a purged key from its backup. Refer to [Restoring Backup](#) for details.

Backing Up Azure Keys

Warning

- The Physical and Virtual storage appliances have storage limits. Ensure that you have sufficient storage before enabling scheduled backup.
- If you do have sufficient storage, you can extend the storage, refer to [Resizing the Virtual Hard Disk After Deployment](#) for details.

To backup an Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Backup Now**. The Backup Now dialog box is displayed.

Note

Backup Now is the manual creation of a cloud key's backup.

4. (Optional) Enter the **Backup Name**.
5. (Optional) Enter the **Description**.
6. Click **Backup**.

A message **Backup created successfully for key <key name>** is displayed on the screen.

Restoring Service Backup

To restore a purged Azure key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Restore Service Backup**. The Confirm Restore Key dialog box is displayed.
4. Select the desired key vault from the **Select Vault** drop-down list.

Note

- Restoration of keys among cross-region vaults is not allowed.

5. Click **Restore**.

A message **Key <key name> restored** is displayed on the screen. The key is restored to the selected key vault. The key status changes to **AVAILABLE**.

Deleting Backup

Deleting a backup permanently removes the backup of a deleted Azure key from CCKM.

Note

This operation can be performed only on keys with a **DELETED** status. You can [delete](#) or [purge](#) keys to change their status to **DELETED**.

To delete the backup of an Azure key from CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Delete Backup**. The Hard Delete Azure Key dialog box is displayed.
4. Select the **I wish to delete the backup of this key.** check box.
5. Click **Delete Key Backup**.

A message **Key <key name> backup deleted** is displayed on the screen. The backup of the key is permanently deleted from CCKM.

Downloading Keys and Versions

Asymmetric RSA/RSA-HSM keys and their older versions with the status AVAILABLE can be downloaded to your local machines. Other keys cannot be downloaded.

Downloading Keys

To download an AVAILABLE asymmetric RSA/RSA-HSM key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the overflow icon () corresponding to the desired key and click **Download Key**. The key is downloaded.

Downloading Key Versions

To download an older version of an AVAILABLE asymmetric RSA/RSA-HSM key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The list of available Azure keys is displayed.
3. Click the expand icon () corresponding to the desired key. The Older Versions are displayed.
4. Click the overflow icon () corresponding to the desired key version and click **Download Key**. The key version is downloaded.

Alternatively, click the key link under the Key Name field to view the key details. Scroll down to the VERSIONS section.

Note

If an exportable Luna HSM key is created on Azure cloud, and the synchronization operation is performed on CCKM, the Exportable field under VERSIONS does not show any value for the key.

Managing Azure Certificates

This section describes how to manage Azure certificates on CCKM. Before proceeding, you must have an Azure vault added to the CCKM. Refer to [Managing Azure Vaults](#) for details.

A CCKM User must have the following permissions on the CipherTrust Manager to perform operations on certificates:

- ReadLocalCA
- ReadCertificate
- UpdateSoftDeleteAzureCertificateCCKM
- UpdateHardDeleteAzureCertificateCCKM
- UpdateRecoverAzureCertificateCCKM
- RestoreAzureCertificateCCKM
- CreatAzureCertificateCCKM
- UploadAzureCertificateCCKM

Adding Azure Certificates

CCKM provides two methods to add Azure certificates:

- [Generating Certificate](#)
- [Importing Certificate](#)

Generating Certificate

To add an Azure Certificate by generating a certificate:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab.
4. Click **Add Certificate**. The Select Material Origin screen of the Add Certificate wizard is displayed.

Select Material Origin

1. Under **Select Method**, select **Generate Certificate**.
2. Click **Next**. The Configure Certificate screen is displayed.

Configure Certificate

1. Enter **Certificate Name**.
2. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types can be Standard and Premium. Vaults stored in Azure Managed HSM pools are not supported.
3. Enter **Issuer Name**.
4. Enter **Subject**.
5. (Optional) Enter **DNS Name**.
6. (Optional) Enter **Validity Period (in months)**.
7. Select **Content Type** from the available options. The supported types are PKCSW12 and PEM.
8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; . ' " _ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Enable or disable **Advanced Configuration**.
 - Click **Yes** to enable.
 - Click **No** to disable.
10. Click **Next**. The Add Schedule screen is displayed.

Add Schedule

1. Under **Lifetime Action Type**, select either **Automatically Renew** or **Email Contacts**.
 - **Automatically Renew**: Azure generates a new key pair and updates the certificate automatically.
 - **Email Contacts**: Renewal notifications are sent to contacts configured in Azure (not managed in CCKM).
2. Under **Select Action**, select either **Percentage Lifetime** or **Number of Days Before Expiry**.
3. As per the selection above, enter the **Percentage** of Lifetime or **Days** Before Expiry.
4. Click **Next**. The Advanced Configuration screen is displayed.

Advanced Configuration

Note

To provide the Advance Configuration details, you need to enable **Advance Configuration** when providing the Certificate Configuration details.

1. (Optional) Enter **Extended Key Usages (EKUs)** separated by comma.
2. (Optional) Select **X509 Key Usage Flags** from the drop-down list.
3. Select **Key Type**.
 - For a standard vault, the key type is **RSA**.
 - For a premium vault, the key type is **EC (Elliptic Curve)**.
4. (Applicable to **RSA** key type) Select **Key Size** from the following options: 2048, 3072, and 4096.
5. (Applicable to **Elliptic Curve** key type) Select **Elliptic Curve Name** from the following options: P-256, P-384, P-521, and P-256K.
6. (Optional) Select the **Reuse Key on Renewal** check box.
7. (Optional) Select the **Exportable Private Keys** check box.
8. (Optional) Select the **Enable Certificate Transparency** check box.
9. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the certificate details that you have provided. These details are divided into MATERIAL ORIGIN, CONFIGURE CERTIFICATE, ADD SCHEDULE, and ADVANCED CONFIGURATION sections.

Before adding the certificate, review all details. After the certificate is added, certain features will no longer be editable.

1. Review the certificate details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **MATERIAL ORIGIN**, **CONFIGURE CERTIFICATE**, **ADD SCHEDULE**, and **ADVANCED CONFIGURATION** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Certificate**.

The certificate creation starts. A Create Certificate In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the MATERIAL ORIGIN, CONFIGURE CERTIFICATE, ADD SCHEDULE, and ADVANCED CONFIGURATION sections becomes Complete, the certificate is created successfully.

3. Click **OK**. The Add Certificate wizard is closed.

Importing Certificate

Note

A CCKM User must have the **ReadLocalCA** and **ReadCertificate** permissions on the CipherTrust Manager to import certificates.

To add an Azure Certificate by importing a certificate:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab.
4. Click **Add Certificate**. The Select Material Origin screen of the Add Certificate wizard is displayed.

Select Material Origin

1. Under **Select Method**, select **Import Certificate**.
2. Click **Next**. The Configure Certificate screen is displayed. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types can be Standard and Premium. Vaults stored in Azure Managed HSM pools are not supported.

Import Certificate

1. Enter **Certificate Name**.
2. Select the desired **Vault** from the drop-down list. Vaults stored in Azure Managed HSM pools are not supported.

3. Select **CA ID**.
4. Select **Certificate Identifier**.
5. Upload **Private Key (pem file)**.
6. Enable or disable **Is Certificate File Password Protected?**.
 - Click **Yes** to enable.
 - Click **No** to disable.
7. Enter **Password**.

Note

If the **Is Certificate File Password Protected?** option is disabled, then you do not need to provide the password.

8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; : ' " _ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the certificate details that you have provided. These details are divided into MATERIAL ORIGIN and CONFIGURE CERTIFICATE sections.

Before adding the certificate, review all details. After the certificate is added, certain features will no longer be editable.

1. Review the certificate details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the MATERIAL ORIGIN and CONFIGURE CERTIFICATE sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Certificate**.

The certificate creation starts. A Create Certificate In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the MATERIAL ORIGIN and CONFIGURE CERTIFICATE sections becomes Complete, the certificate is created successfully.

3. Click **OK**. The Add Certificate wizard is closed.

Viewing Azure Certificates

Search for Azure Certificates by Certificate Name, Certificate Vault, or Tags.

CCKM does not allow searching for certificates:

- By tag values using colon (:)
- By "certificate:value" pair using these characters:

\, : " %

To view an Azure certificate:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab. The list of available Azure certificates is displayed. The Azure Certificates tab displays the following details:

Field	Description
Certificate Name	Unique, user-friendly alias of the certificate. This is useful in searching for specific certificates.
Version ID	Current version of the certificate. Click the expand icon (▸) corresponding to a certificate to view its versions.
Status	State of the certificate. The status can be: • Available • Soft Deleted • Deleted
Origin	Source of the certificate material. The origin of the

Field	Description
	certificate can be: • CCKM: Certificate material is created on CCKM. • Native: Certificate material is created on the cloud. • External (Unknown): Source of the certificate material is unknown. It is different than CCKM and the native cloud.
Source	The source of the certificate.
Creation Date	Time when the certificate is created.
Expiry Date	Time when the certificate is created.
Key Vault	Name of the Azure key vault.
Region	Azure region where the certificate is created. Click the filter icon () to view the list of supported Azure regions.
Issuer Name	Name for the referenced issuer object or reserved names. Set to Self for a self-signed certificate.

Sometimes, you might notice certain certificates are displayed as grayed out. This happens when the certificates are no longer accessible. For example, when:

- Any cloud permissions on the certificates are changed. The certificates are no longer accessible from the Azure connection.
- Connection is changed in KMS. The new connection does not have permissions to access the certificates.

Editing Azure Certificates

To view or edit an Azure certificate:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab. The list of available Azure certificates is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **View/Edit**.
5. Configure the **CERTIFICATE SCHEDULES**. Refer to [CERTIFICATE SCHEDULES](#) for details.

CERTIFICATE SCHEDULES

To configure certificate schedules:

1. In the **KEY SCHEDULES** section, select/enter the following details:
 - From the **Select Rotation Schedule** drop-down list, select a rotation schedule.
 - In the **Rotation Settings** section, select/enter the following details:
 - a. **Key Origin:** Select the key origin from the available options. The key origin can be CipherTrust, Native (Azure), Luna, or DSM.
 - b. (Applicable to Luna key origin) **Select Partition:** Select the Luna HSM partition.
 - c. **Key Type:** Select the key type. Key types differ based on the key origin.
 - For CipherTrust, Luna, and DSM, the supported key types are RSA and RSA-HSM.
 - For Native (Azure), the supported key types are RSA, EC, RSA-HSM, and EC-HSM.
 - d. Select the **Key Size** or **Elliptical Curve Name** depending on the selected Key Type.
 - If the key type is RSA or RSA-HSM, select **Key Size**. The available options are 2048, 3072, and 4096.
 - If the key origin is Native (Azure) and key type is EC or EC-HSM, select **Elliptical Curve Name**. The available options are P-56, P-384, P-521, and SECP256K1.
 - e. Select **Enabled** if you want to enable the rotated key.
2. Click **Update**.

A message **Key schedule updated successfully** is displayed on the screen.

Refreshing Azure Certificates

Refreshing is the process of downloading secrets created on the Azure key vault to CCKM. Certificates from all key vaults are refreshed at once.

To refresh certificates:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab. The list of available Azure certificates is displayed.
4. Click **Refresh**. The **This may take a while...** message is displayed.

Note

Refresh is a time intensive operation that could take several hours or days to complete. It will continue running in the background. Do you want to continue?

5. Click **Refresh** to continue.

A message **Refresh started...** is displayed on the screen. The refreshed certificates are listed on the **Cloud Keys > Azure > Certificates** tab.

To cancel the refresh:

1. Click **Cancel Refresh**. The **Cancel Refresh?** message is displayed. The action will terminate all currently active refresh operations. All progress will be lost and this action cannot be undone. Do you want to proceed to cancel refresh or cancel this action?
2. Click **Cancel Refresh**.

A message **Refresh cancelled successfully** is displayed on the screen.

Rotating Certificates (Add Version)

Note

To rotate Azure Certificates, **CCKM Users** require **Add Certificate** and **Upload Certificate** permissions.

To rotate a certificate:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab. The list of available Azure certificates is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **Rotate Now (Add Version)**. The Select Material Origin screen of the Add New Version wizard is displayed.
5. Select Key Material Origin. Depending on your requirements, select an appropriate option. Refer to the following sections for details:
 - [Generating Certificate](#)
 - [Importing Certificate](#)

Soft-Deleting Azure Certificates

Soft deleting is the process of deleting Azure certificates from the Azure vaults and CCKM. These certificates still exist on CCKM and in the Azure vaults. The soft-deleted certificates can be recovered.

Note

This operation can be performed only on the Azure certificates residing in the soft-enabled key vaults.

To soft-delete an Azure certificate:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab. The list of available Azure certificates is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **Soft Delete**. The Confirm Soft Delete dialog box is displayed.
5. Click **Soft Delete**.

A message **Certificate <certificate name> soft-deleted** is displayed on the screen. The status of the certificate changes to **SOFT - DELETED**.

Recovering Soft-Deleted Azure Certificates

If needed, you can recover a soft-deleted certificate.

To recover a soft-deleted Azure certificate:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab. The list of available Azure certificates is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **Recover**. The Confirm Recover Certificate dialog box is displayed.
5. Click **Recover**.

A message **Certificate <certificate name> recovered from soft-delete.** is displayed on the screen. The status of the certificate changes to **AVAILABLE**.

Purging Azure Certificates

Purging is the process of permanently deleting soft-deleted Azure certificates from the Azure vaults. However,

backup of the purged certificate can be restored on CCKM. If you wish to restore backup of the purged certificate, follow the steps mentioned in the [Restoring Backup](#) section.

Note

This operation can be performed only on the soft-deleted Azure certificates residing in the soft-enabled key vaults.

To purge an Azure certificate:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab. The list of available Azure certificates is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **Purge**. The Purge Azure certificate dialog box is displayed.
5. Select the **I wish to purge this certificate.** check box.
6. Click **Purge Certificate**.

Purging a certificate might take some time. After successful deletion, a message **Certificate <certificate name> hard deleted** is displayed on the screen. The status of the certificate changes to **DELETED**.

If needed, you can restore a purged certificate from its backup. Refer to [Restoring Backup](#) for details.

Restoring Backup

To restore a purged Azure certificate:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab. The list of available Azure certificates is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **Restore**. The Confirm Restore certificate dialog box is displayed.
5. Select the desired certificate vault from the **Select Vault** drop-down list.

Note

Restoration of certificates among cross-region vaults is not allowed.

6. Click **Restore Certificate**.

A message **Certificate <certificate name> restored** is displayed on the screen. The certificate is restored to the selected key vault. The certificate status changes to **AVAILABLE**.

Deleting Backup

Deleting a backup permanently removes the backup of a deleted Azure certificate from CCKM.

Note

This operation can be performed only on the certificates with **DELETED** status.

To delete the backup of an Azure certificate from CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Certificates** tab. The list of available Azure certificates is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **Delete Backup**. The Hard Delete Azure Certificate dialog box is displayed.
5. Select the **I wish to delete the backup of this certificate.** check box.
6. Click **Delete Certificate Backup**.

A message **Backup of Certificate <certificate name> deleted successfully** is displayed on the screen. The backup of the certificate is permanently deleted from CCKM.

Managing Azure Secrets

This section describes how to manage Azure Secrets on CCKM. Before proceeding, you must have an Azure key vault added to the CCKM. Refer to [Managing Azure Vaults](#) for details.

Adding Azure Secrets

To add an Azure Secret:

1. Open the **Cloud Key Manager** application.
 2. In the left pane, click **Cloud Keys > Azure**.
 3. Click the **Secrets** tab. The Add Secret wizard is displayed.
 4. Enter **Secret Name**.
-

5. Enter **Secret Value**.
6. Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types can be Standard and Premium. Vaults stored in Azure Managed HSM pools are not supported.
7. Enter **Content Type**.
8. (Optional) Set the secret activation and expiration dates.
 - Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the secret.
 - Select the **Set Expiration Date** check box and from the on-screen calendar, select the secret expiration date and time.
9. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.
 To add a tag:
 - a. Specify a tag name.
 - b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; : ' " _ **`
 - c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.
10. Click **Save**.

Viewing Azure Secrets

Search for Azure Secrets by Secret Name.

CCKM does not allow searching for secrets:

- By tag values using colon (:)
 - By "secret:value" pair using these characters:

`\ , : " %`

To view an Azure secret:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Secrets** tab. The list of available Azure secrets is displayed. The Azure Secrets tab displays the following details:

Field	Description
Secret Name	Unique, user-friendly alias of the secret. This is useful in searching for specific secrets.
Version ID	Current version of the secret. Click the expand icon () corresponding to a secret to view its versions.
Status	State of the secret. The status can be: • Available • Soft Deleted • Deleted
Cloud	Name of the cloud.
Creation Date	Date and time when the secret is created.
Expiry Date	Date and time when the secret will expire.
Key Vault	Name of the Azure key vault.
Region	Azure region where the secret is created. Click the filter icon () to view the list of supported Azure regions.

Sometimes, you might notice certain secrets are displayed as grayed out. This happens when the secrets are no longer accessible. For example, when:

- Any cloud permissions on the secrets are changed. The secrets are no longer accessible from the Azure connection.
- Connection is changed in KMS. The new connection does not have permissions to access the secrets.

Refreshing Azure Secrets

Refreshing is the process of downloading secrets created on the Azure key vault to CCKM. Secrets from all key vaults are refreshed at once.

To refresh secrets:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Secrets** tab. The list of available Azure secrets is displayed.

- Click **Refresh**. The **This may take a while...** message is displayed.

Note

Refresh is a time intensive operation that could take several hours or days to complete. It will continue running in the background. Do you want to continue?

- Click **Refresh** to continue.

A message **Refresh started...** is displayed on the screen. The refreshed secrets are listed on the **Cloud Keys > Azure > Secrets** tab.

To cancel the refresh.

- Click **Cancel Refresh**. The **Cancel Refresh?** message is displayed. The action will terminate all currently active refresh operations. All progress will be lost and this action cannot be undone. Do you want to proceed to cancel refresh or cancel this action?
- Click **Cancel Refresh**.

A message **Refresh cancelled successfully** is displayed on the screen.

Rotating Secrets (Add Version)

Note

To rotate Azure Secrets, **CCKM Users** require **Add Secret** and **Upload Secret** permissions.

To rotate a secret:

- Open the **Cloud Key Manager** application.
- In the left pane, click **Cloud Keys > Azure**.
- Click the **Secrets** tab. The list of available Azure secrets is displayed.
- Click the overflow icon () corresponding to the desired alias and click **Rotate Now (Add Version)**. The Add New Version screen is displayed.
- Enter **Secret Value**.
- Select the desired **Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types can be Standard and Premium. Vaults stored in Azure Managed HSM pools are not supported.

7. Enter **Content Type**.
8. (Optional) Set the secret activation and expiration dates.
 - Select the **Set Activation Date** check box and from the on-screen calendar, select the date and time to activate the secret.
 - Select the **Set Expiration Date** check box and from the on-screen calendar, select the secret expiration date and time.
9. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
 - CCKM allows the following characters in tag values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; : ' " _ **`
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

10. Click **Save**.

A message **Secret successfully rotated** is displayed on the screen. Navigate to **Cloud Keys > Azure > View/Edit > Versions** to view the versions of the rotated Azure secret.

Soft-Deleting Azure Secrets

Soft deleting is the process of deleting Azure secrets from the Azure vaults and CCKM. These secrets still exist on CCKM and in the Azure vaults. The soft-deleted secrets can be recovered.

Note

This operation can be performed only on the Azure secrets residing in the soft-enabled key vaults.

To soft-delete an Azure secret:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.

3. Click the **Secrets** tab. The list of available Azure secrets is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **Soft Delete**. The Confirm Soft Delete dialog box is displayed.
5. Click **Soft Delete**.

A message **Secret <secret name> soft-deleted** is displayed on the screen. The status of the secret changes to **SOFT-DELETED**.

Recovering Soft-Deleted Azure Secrets

If needed, you can recover a soft-deleted secret.

To recover a soft-deleted Azure secret:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Secrets** tab. The list of available Azure secrets is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **Recover**. The Confirm Recover Secret dialog box is displayed.
5. Click **Recover**.

A message **Secret <secret name> recovered from soft-delete.** is displayed on the screen. The status of the secret changes to **AVAILABLE**.

Purging Azure Secrets

Purging is the process of permanently deleting soft-deleted Azure secrets from the Azure vaults. However, backup of the purged secret can be restored on CCKM. If you wish to restore backup of the purged secret, follow the steps mentioned in the [Restoring Backup](#) section.

Note

This operation can be performed only on the soft-deleted Azure secrets residing in the soft-enabled key vaults.

To purge an Azure secret:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Secrets** tab. The list of available Azure secrets is displayed.

4. Click the overflow icon () corresponding to the desired alias and click **Purge**. The Purge Azure secret dialog box is displayed.
5. Select the **I wish to purge this secret.** check box.
6. Click **Purge Secret**.

Purging a secret might take some time. After successful deletion, a message **Secret <secret name> hard deleted** is displayed on the screen. The status of the secret changes to **DELETED**.

If needed, you can restore a purged secret from its backup. Refer to [Restoring Backup](#) for details.

Restoring Backup

To restore a purged Azure secret:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Secrets** tab. The list of available Azure secrets is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **Restore**. The Confirm Restore secret dialog box is displayed.
5. Select the desired secret vault from the **Select Vault** drop-down list.

Note

Restoration of secrets among cross-region vaults is not allowed.

6. Click **Restore Secret**.

A message **Secret <secret name> restored** is displayed on the screen. The secret is restored to the selected key vault. The secret status changes to **AVAILABLE**.

Deleting Backup

Deleting a backup permanently removes the backup of a deleted Azure secret from CCKM.

Note

This operation can be performed only on the secrets with **DELETED** status.

To delete the backup of an Azure secret from CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**.
3. Click the **Secrets** tab. The list of available Azure secrets is displayed.
4. Click the overflow icon () corresponding to the desired alias and click **Delete Backup**. The Hard Delete Azure Secret dialog box is displayed.
5. Select the **I wish to delete the backup of this secret.** check box.
6. Click **Delete Secret Backup**.

A message **Secret <secret name> backup deleted** is displayed on the screen. The backup of the secret is permanently deleted from CCKM.

Scheduling Operations

CCKM allows you to periodically refresh keys, rotate keys, or add KMS containers in the background. To refresh keys, rotate keys, or add KMS containers, you need to create schedule configuration. A schedule configuration defines when to run a refresh, rotation, or KMS container job. You can use either the basic or cron format to specify when the rotation, refresh, or KMS container job will run.

Specify basic format in the following order:

[Repeats, at]

Where,

- Repeats: frequency of the scheduler. Possible values are daily, weekly, monthly, and yearly.
- at: specific time at which the rotation or refresh operation will be performed. Possible value lies between 12:00-11:00 AM/PM UTC.

Specify cron format in the following order:

"* * * * *"

These five values indicate when the job should be executed. These values are mandatory and must be specified in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

Field	Allowed values	Allowed special characters
Minute	0-59	* / , -
Hour	0-23	* / , -
Day of month	1-31	* / , - ?
Month	1-12 or JAN-DEC	* / , -
Day of week	0-6 or SUN-SAT	* / , - ?

Adding Key Rotation Schedule

A key rotation schedule replaces selected keys with new cryptographic keys at the specified time.

To add key rotation schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. Select **Schedule Type** as **Key Rotation**.
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **Azure** from the **Cloud Name** drop-down list.
 - Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. Set the values on the **Schedule Config** screen.

You can schedule configuration for:

- **All Keys**: Refer to [Configuring Rotation Schedule for All Keys](#) for details.
- **Only Expiring Keys**: Refer to [Configuring Rotation Schedule for Expiring Keys](#) for details.
- **All Keys Based on Creation/Last Rotation Date**: Refer to [Configuring Rotation Schedule for All Keys Based on Creation or Last Rotation Date](#) for details.

Configuring Rotation Schedule for All Keys

To configure rotation schedule for all keys:

1. Configure **DURATION** for the scheduler. Specify the following values:
 - a. **Schedule Starts**
 - b. **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.

2. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either the **Basic** or the **Raw (Cron)** format to specify the value.
3. (Optional) Set the expiry of the new key that will be created through rotation. If not specified, the new key material never expires. For example, if you set the expiry time to **6 Hours**, the key material of the new key will expire in 6 hours.
4. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Configuring Rotation Schedule for Expiring Keys

To configure rotation schedule for expiring keys:

1. In the **SELECTION** section, select the keys to be rotated based on their expiry time. In the **Only rotate keys within _ of material expiration** field, select or enter the key expiry time. For example, if the value of **Only rotate keys within _ of material expiration** field is set to **4 Days**, then the rotation schedule is created for all the keys that will expire in 4 days.
2. Configure **DURATION** for scheduler. Specify the following values:

a. **Schedule Starts**

b. **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the **Schedule Ends** field becomes unavailable.

3. (Optional) Set the expiry of the new key that will be created through rotation. If not specified, the new key material never expires. For example, if you set the expiry time to **6 Hours**, the key material of the new key will expire in 6 hours.
4. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Configuring Rotation Schedule for All Keys Based on Creation or Last Rotation Date

To configure rotation schedule for all keys based on their creation or last rotation date:

1. In the **SELECTION** section, select the keys to be rotated based on their creation or last rotation date.

In the **Only rotate keys after _ of creation/last rotation date** field, select or enter the number of days, weeks, months, or years. For example, if the value of **Only rotate keys after _ of creation/last rotation date** field is set to **4 Days**, then all the keys will rotate after 4 days of their creation or last rotation date. The default duration is 7 Days.

2. Configure **DURATION** for scheduler. Specify the following values:

- a. **Schedule Starts**
- b. **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.

3. (Optional) Set the expiry of the new key that will be created through rotation. If not specified, the new key material never expires. For example, if you set the expiry time to **6 Hours**, the key material of the new key will expire in 6 hours.
4. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Adding Key Refresh Schedule

To add a key refresh schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. Select **Schedule Type** as **Key Refresh**.
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **Azure** from the **Cloud Name** drop-down list.
 - Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. On the **Schedule Config** screen, enter or select the following details.
 - a. Configure **DURATION** for the scheduler. Specify the following values:
 - **Schedule Starts**
 - **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.

- b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
- c. Click **Next**.

8. On the **Add Vaults** screen:

- a. Select the vault name from the **Refresh Keys, Certificates and Secrets from Selected Vault(s)** drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types can be Standard, Premium, and ManagedHSM.
- b. Click **+** button.

Similarly, you can add more vaults.

- c. Select the **Refresh All Vaults** option if you want to refresh all the existing vaults.
- d. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Automatic Cloud Key Discovery

A KMS container schedule automatically detects the KMS containers based on the Azure Cloud connection and adds them to the CipherTrust Cloud Key Manager.

Adding a KMS container (vault or subscription) to CCKM consumes one CCKM license count.

To add a KMS container schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Select Schedule Type screen of the Add New Schedule wizard is displayed.

Schedule Type

1. Select Schedule Type as **Add KMS Container**.
2. Click **Next**. The General Info screen is displayed.

General Info

1. Enter a unique **Schedule Name**.
 2. Select **Azure** from the **Cloud Name** drop-down list.
 3. (Optional) Add **Description** for the scheduler in a maximum of 250 characters.
-

4. (Optional) Select **Enable Schedule** to enable the schedule.
5. Select a **Define Job Type**. You can select either **Discover Only** or **Discover and Add**.

Note

Discover Only lists all the discovered containers. Discover and Add adds all the discovered containers to the CCKM.

6. Click **Next**. The Connection & ACL Info screen is displayed.

Connection & ACL Info

1. Select a **Connection**.
2. In the ACCESS CONTROL section.
 - a. Click **Keys**, **Certificates**, **Secrets**, or **Reports** tab.
 - b. Click **Assign User/Group**. The Assign User/Group dialog box is displayed.
 - c. Select the desired user or group from the **User/Group** drop-down list.
 - d. Click **Save**. The newly added user/group is displayed under **Name** in the ACCESS CONTROL section.
 - e. Grant the permissions to the user/group, as appropriate.

Granting additional permissions

To grant the additional permissions, select the check box under the desired operation corresponding to the desired users or groups. The following permissions can be granted.

- **Keys:** View, Add Key, Edit Key, Recover Key, Purge Key, Upload Key, Restore Key, Soft Delete Key, Delete Key Backup, Refresh Key
- **Certificates:** View, Add, Upload, Edit, Soft Delete, Recover, Purge, Restore, Delete Backup, Refresh
- **Secrets:** View, Add, Edit, Soft Delete, Recover, Purge, Restore, Delete Backup, Refresh
- **Reports:** View Report, Add Report, Download Report, Delete Report

To remove a permission, clear the check box under the desired operation.

- f. Remove a user or group.
 - a. Under **Unassign**, click **X** corresponding to the desired user/group. The Remove Group dialog box is displayed.

b. Click **Remove**.

3. Click **Next**. The Schedule Config screen is displayed.

Schedule Config

1. Configure **DURATION** for the scheduler. Specify the schedule start and end time:

- **Schedule Starts**: Specify time when the schedule starts.
- **Schedule Ends**: Unavailable by default, that is, the scheduler never expires.
- **Never**: Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.

2. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.

3. Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created schedule is displayed in the schedules list.

After the schedule is run successfully, Azure Key Vaults and Azure Subscriptions will be added to the CCKM. You can see them in the Vaults and Subscriptions sections, of the Azure Key Vaults page in the CCKM.

Adding Key Backup (Azure only) Schedule

A key backup schedule downloads and saves encrypted cloud key backup blobs at specified time intervals. These backups can be used to restore keys.

Warning

- The Physical and Virtual storage appliances have storage limits. Ensure that you have sufficient storage before enabling scheduled backup.
- If you do have sufficient storage, you can extend the storage, refer to [Resizing the Virtual Hard Disk After Deployment](#) for details.

To add a Key Backup (Azure only) schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Select Schedule Type screen of the Add New Schedule wizard is displayed.

Schedule Type

1. Select Schedule Type as **Key Backup (Azure only)**.
2. Click **Next**. The General Info screen is displayed.

General Info

1. Enter a unique **Schedule Name**.
2. (Optional) Add **Description** for the scheduler in a maximum of 250 characters.
3. (Optional) Select **Enable Schedule** to enable the schedule.
4. Click **Next**. The Schedule Config screen is displayed.

Schedule Config

Note

Backup Limit: If the sum of Scheduled and Manual Backups exceeds the maximum allowance of 30 instances, older backups will be deleted automatically.

1. Configure **DURATION** for the scheduler. Specify the schedule start and end time:
 - **Schedule Starts**: Specify time when the schedule starts.
 - **Schedule Ends**: Unavailable by default, that is, the scheduler never expires.
 - **Never**: Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.
2. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
3. Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created schedule is displayed in the schedules list.

Viewing/Editing Schedules

To view/edit a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The Schedules page displays following details:

Field	Description
Name	Unique name of the scheduler configuration.
Schedule Type	Type of the schedule. Possible types are: • Key Rotation • Key Refresh • Add KMS Container (GCP and Azure only) • Cloud Key Backup
Cloud Name	Name of the cloud.
Last Modified	Time when the schedule is modified.
Frequency	Frequency of the scheduler configuration.
Start Date	Creation time of the scheduler configuration.
End Date	Expiry time of the scheduler configuration.
Status	Status of the scheduler configuration. Possible values are: • Enabled • Disabled

- Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.
- Edit or configure the following fields:
 - Description.
 - Status of the scheduler configuration.
 - (For KMS Container schedule) Define Job Type.
 - **CONNECTION & ACL INFO** details, refer to the add KMS container [Connection & ACL Info](#) section.
 - Scheduler configuration parameters such as duration, frequency, and expiry time of the key. Refer to the key rotation [Schedule Config](#) screen, key refresh [Schedule Config](#) screen, or KMS container [Schedule Config](#) screen for details, as appropriate.
- Click **Update** to save the changes.

JOB HISTORY

The Schedules page also contains a section named JOB HISTORY. Every time a schedule is run, a Job is created.

- This section displays information related to a job such as **Run Date**, **Job ID**, **Status**, and **Errors**.
- To view the details of the job, click the **Job ID** link corresponding to the desired job. The Job details page is displayed. It shows the list of Subscriptions and Vaults along with their details.

- To download the reports of Subscriptions and Vaults, go to the top left of the Job details page, click **Download**, and select a **report** from the drop-down menu. The options are **Subscriptions Only**, **Vaults Only**, and **Subscriptions and Vaults**.

Note

You can download the report after the schedule has run successfully.

Disabling Schedules

To disable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Disable**. The Disable Schedule message is displayed.
4. Click **Disable Schedule**.

A message **Successfully disabled the schedule** is displayed on the screen.

Enabling Schedules

To enable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Enable**. The Enable Schedule message is displayed.
4. Click **Enable Schedule**.

A message **Successfully enabled the schedule** is displayed on the screen.

Manually Running Schedules

To manually run a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Run Now**.

The Run Now schedule is started in the background. A message **Schedule is running now. It will take a few seconds to finish.** is displayed on the screen. After the schedule is run successfully, a message **Successfully finished running the schedule** is displayed.

Deleting Schedules

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Delete Schedule**. The Delete Schedule message is displayed.
4. Click **Delete**.

A message **Successfully deleted schedule** is displayed on the screen.

Managing Azure Reports

CCKM provides options to generate key visibility reports based on:

- Key-related activities between CCKM and Azure Vault
- Track keys by their expiration dates
- What applications are using the keys

Azure reports are categorized as:

- **Key Activity Report:** Inspect individual Azure key histories by operations, for example, when they were refreshed, rotated, edited, or deleted. Also, use this report to compare key activities between CCKM and Azure Vault.
- **Key Aging Report:** Track keys by their expiration dates. Audit a range of dates, from past material deletions to future scheduled deletions, within the selected Azure Vault.
- **Service/Usage Report:** Monitor key usage by tracking services and applications consuming the keys. View when and where a service requests the use of each key.

Prerequisites

For managing Azure reports on CCKM, you need an Azure Log Analytics workspace ID. This ID needs to be configured in the Diagnostic settings of the key vault for which reports need to be generated. To determine the workspace ID:

1. Configure Log analytics workspace. Refer to [Create a Log Analytics workspace in the Azure portal](#) for details.

Note

The Log Analytics Reader role is required to access Log Analytics workspace to generate the Azure reports on CCKM. Usually, the role comes via inheritance when the Reader role is assigned to an App on subscription. If the App does not have the read permission at the subscription level, the read permission should be explicitly provided at the individual resource level (Log Analytics workspace).

2. Determine the Workspace ID of the desired Log Analytics workspace. Click the workspace name to view its details including the **Workspace ID**.
3. Navigate to the Key vaults page. This page shows the list of existing Azure key vaults.
4. Perform the following steps:
 - a. Under **Name**, click the desired key vault. The <key-vault> page is displayed.
 - b. Click **Diagnostic settings > Add diagnostic setting**.
 - c. Specify a **Diagnostic setting name**.
 - d. Under **Category details**, select **AuditEvent** and **AllMetrics**.
 - e. Under **Destination details**:
 - a. Select **Send to Log Analytics workspace**.
 - b. Select the desired **Subscription** from the drop-down list.
 - c. Select the desired **Log Analytics workspace** from the drop-down list.
5. Save the changes.

The changes may take some time to take effect. After the changes are effective, you can manage Azure reports on CCKM, as described below.

Creating Azure Reports

CCKM users with the **Add Report** permission can create reports.

To create an Azure report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Azure**. The Azure Reports page is displayed.
3. Click **Add Report**. The Choose Report Type and Name screen of the Add Azure Report wizard is displayed.
4. Select a report type. The options are:

-
- Key Activity Report
 - Key Aging Report
 - Service/Usage Report
5. Specify a **Report Name**. This is a mandatory field.
 6. Click **Next**. The Select Analytics Parameters screen of the wizard is displayed.
 7. Select an **Azure Vault** from the drop-down list. The list shows the available Azure key vaults with their types specified in parenthesis. The Azure key vault types are Standard, Premium, and ManagedHSM.
 8. *(Not applicable to Key Aging Report)* Specify a **Workspace ID**. Refer to [Prerequisites](#) for instructions to determine the Workspace ID.
 9. Click **Add**. The selected Azure Vault is displayed under Selected Log Analytics Parameters. Add more vaults, if required.
 10. Click **Next**. The Set Start and End Dates screen of the wizard is displayed.
 11. In the **Include Entries From** field, specify the start date and time for the report.
 - Click the field and select the date and time on the on-screen calendar.
 - Alternatively, enter the time in **MM/DD/YYYYHH:MM** format.
 12. In the **To** field, specify the end date and time for the report.
 13. Click **Save**.

A success message **Created report** is displayed. The newly created report is displayed under the Azure reports list. The report status becomes Running Now.

Viewing Azure Reports

CCKM users with the **View Report** permission can view reports.

The Azure Reports page displays the list of existing reports. Filter the reports by their names.

To view existing Azure reports:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Azure**. The list of available reports is displayed. The following details of the reports are displayed:

Column Name	Description
Report Name	Name of the report.

Column Name	Description
Run Date	When the report is run.
Type	Type of the report. The type can be: • Key Activity Report • Key Aging Report • Service/Usage Report
Start Date	Start date from when the report is generated. If a report is run now, its status becomes Running Now.
End Date	End date for the report.

Viewing Details of an Azure Report

CCKM users with the **View Report** permission can view reports.

To view the details of an Azure report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Azure**. The list of available reports is displayed.
3. Click the desired **Report Name** link. The details of report are displayed.

Alternatively, click the overflow icon () corresponding to the desired alias and click **View**.

Downloading Azure Reports

CCKM users with the **Download Report** permission can download reports.

To download an Azure report as a CSV file:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Azure**. The list of available reports is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Download**.

Alternatively, click the desired report under **Report Name**, and click **Download CSV** in the detail view.

The report is downloaded as a CSV file.

Deleting Azure Reports

CCKM users with the **Delete Report** permission can delete reports.

To delete an Azure report:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **Reports > Azure**. The list of available reports is displayed.
3. Click the overflow icon () corresponding to the desired alias and click **Delete**. The Confirm Delete Report message is displayed.
4. Click **Delete Report**.

The report is removed from the list of reports.

Viewing Azure Subscriptions

In Azure, a subscription is added automatically when a vault is added to the CipherTrust Cloud Key Manager. When the last vault associated with a subscription is removed from CCKM, the subscription is automatically removed from CCKM.

To view the list of Azure subscriptions added to the CipherTrust Manager:

1. Log on to the **CipherTrust Manager** GUI as administrator.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab shows the list of Azure vaults added to the CipherTrust Manager.
4. Click the **Subscriptions** tab. The tab displays the following details:

Column	Description
Subscription ID	ID of the Azure subscription.
Subscription Name	Name of the Azure subscription.
Created At	Date and time when the subscription is added to the CipherTrust Manager.
State	State of the subscription, enabled or disabled.

Performance Summary

This section provides metrics and results of Azure performance testing.

Test Setup

Test Runner	
Operating System	Ubuntu 20.4
RAM	32GB
Core	16

Appliance	
Build	2.8.0.7157
RAM	16GB

Process

Scripts are written on top of the Locust Framework for load and stress testing. The testing is done for create, read, update, and delete (CRUD) key operations with respect to different numbers of virtual users (VUS) over parallel connections, one for every virtual user.

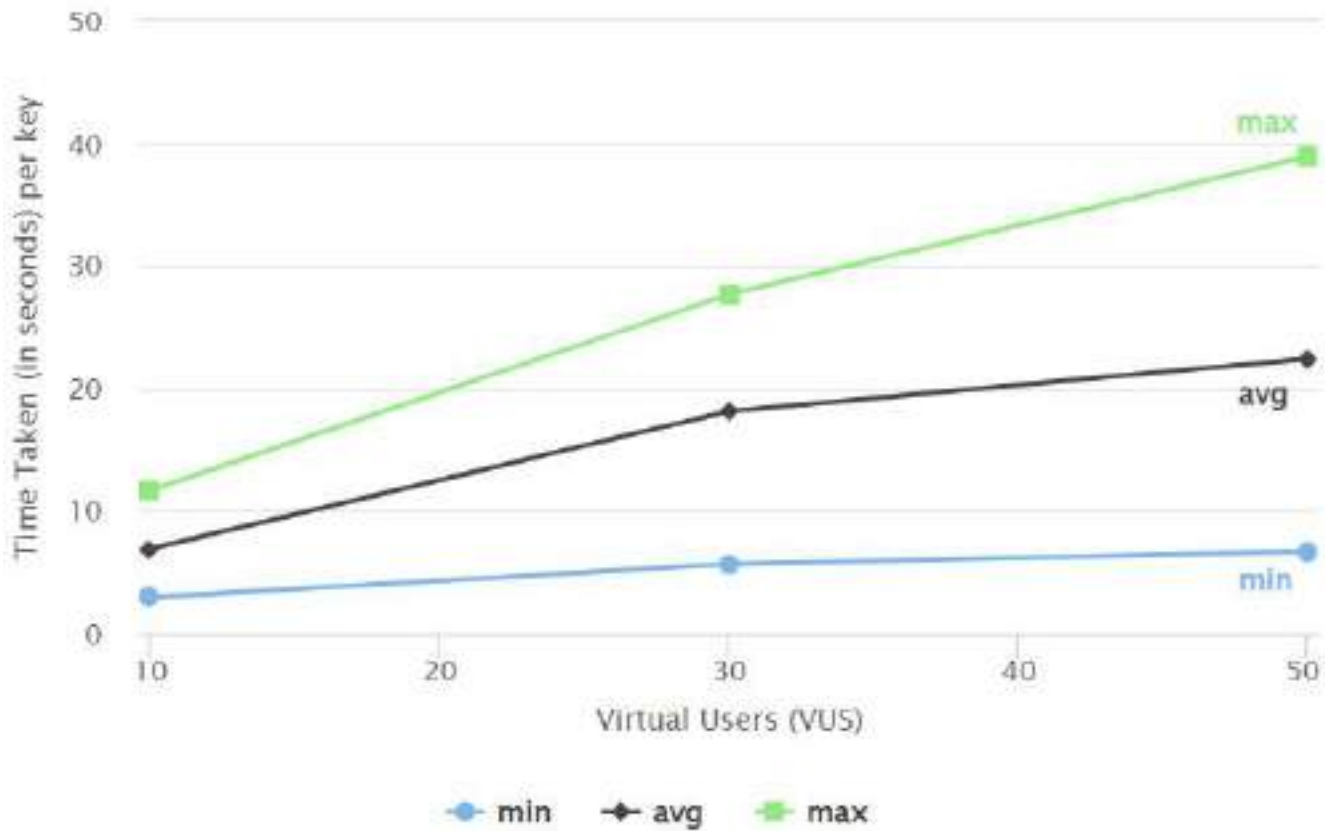
The testing also done to determine the response for uploading keys by a number (N) of virtual users in time (T).

CRUD Operations

Expand the headings below to view performance numbers.

Time taken per key upload with respect to virtual users

Virutal Users	Minimum	Average	Maximum
10	2.971	6.911	11.698
30	5.672	18.155	27.679
50	6.714	22.403	38.941



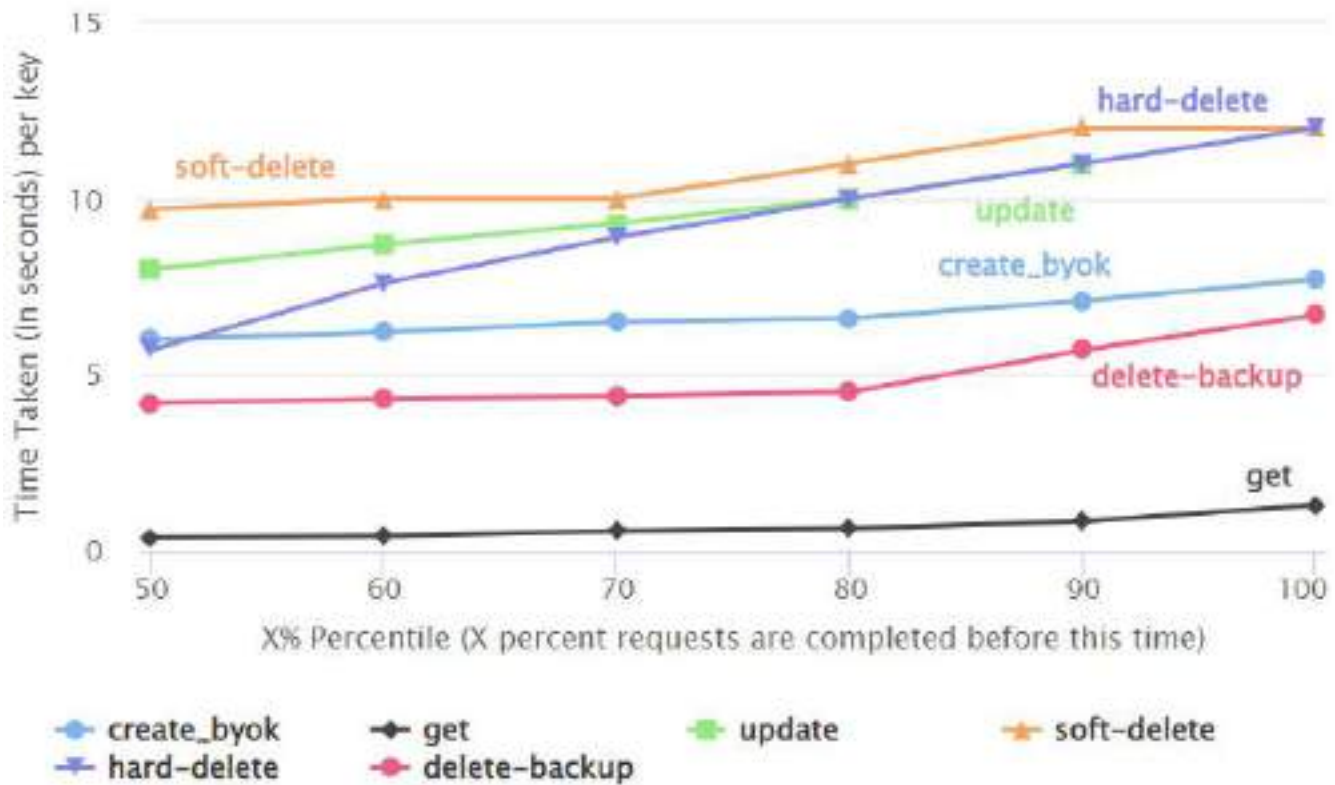
In the graph above:

- X-axis represents the increment of virtual users.
- Y-axis represents the time taken by a virtual user to perform BYOK operations.
- 3 line graphs display information for minimum (min), average (avg), and maximum (max) time taken by one of the virtual users running at that time.

Response with request coverage in percentile (10 VUS, 180 seconds)

X% percentile	create_byok	get	update	soft-delete	hard-delete	delete-backup
50	6	0.36	8	9.7	5.7	4.2
60	6.2	0.41	8.7	10	7.6	4.3
70	6.5	0.55	9.3	10	8.9	4.4
80	6.6	0.63	10	11	10	4.5
90	7.1	0.82	11	12	11	5.7
100	7.7	1.3		12	12	6.7

Here, X% percentile indicates that X percent requests are completed before this time.



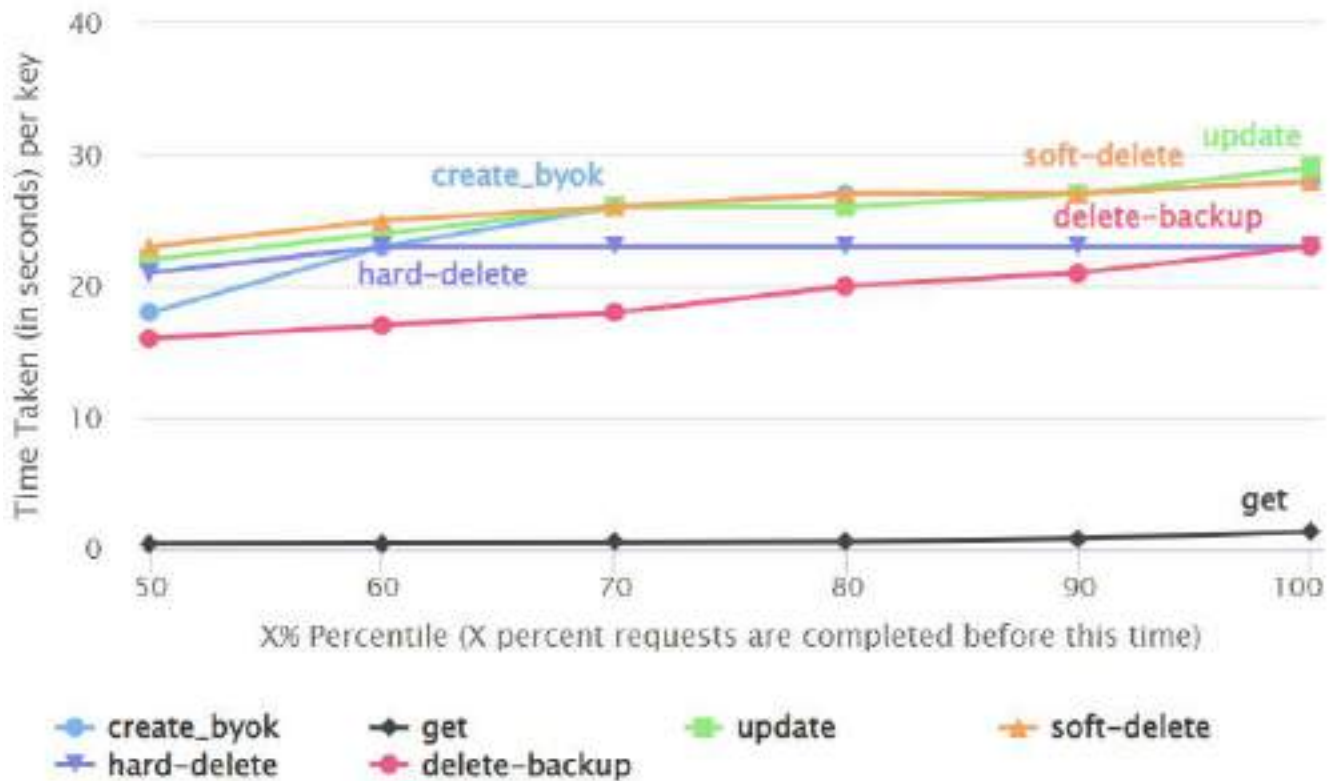
In the graph above:

- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
- Y-axis represents the time taken by a virtual user to perform any CRUD operation.
- An individual virtual user has performed these CRUD operations sequentially, whereas all the virtual users are spawned in parallel.

Response with request coverage in percentile (30 VUS, 180 seconds)

X% percentile	create_byok	get	update	soft-delete	hard-delete	delete-backup
50	18	0.4	22	23	21	16
60	23	0.44	24	25	23	17
70	26	0.48	26	26	23	18
80	27	0.56	26	27	23	20
90	27	0.8	27	27	23	21
100	28	1.3	29	28	23	23

Here, X% percentile indicates that X percent requests are completed before this time.



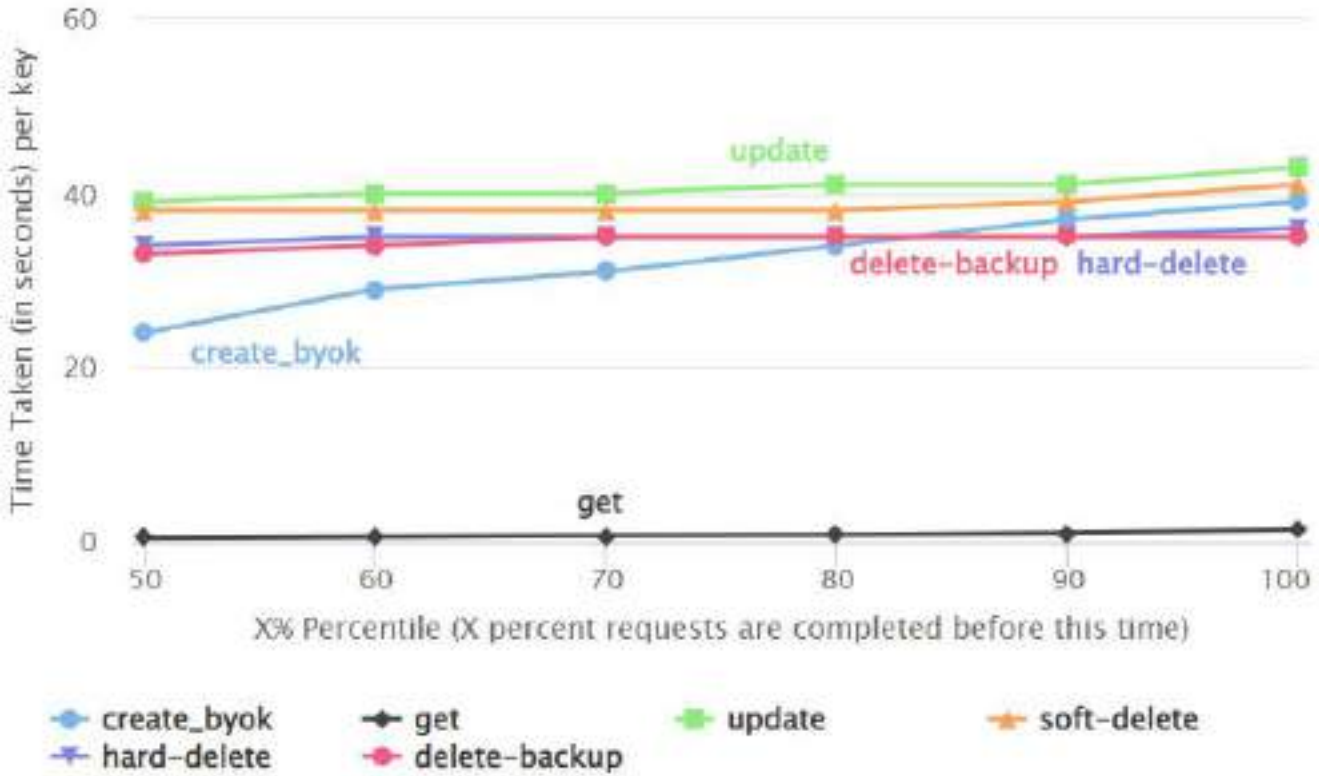
In the graph above:

- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
- Y-axis represents the time taken by a virtual user to perform any CRUD operation.
- An individual virtual user has performed these CRUD operations sequentially, whereas all the virtual users are spawned in parallel.
- t represents time taken for a key operation when N virtual users are running in parallel.

Response with request coverage in percentile (50 VUS, 180 seconds)

X% percentile	create_byok	get	update	soft-delete	hard-delete	delete-backup
50	24	0.43	39	38	34	33
60	29	0.55	40	38	35	34
70	31	0.69	40	38	35	35
80	34	0.78	41	38	35	35
90	37	0.99	41	39	35	35
100	39	1.4	43	41	36	35

Here, X% percentile indicates that X percent requests are completed before this time.



- In the graph above:
- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
 - Y-axis represents the time taken by a virtual user to perform any CRUD operation.
 - An individual virtual user has performed these CRUD operations sequentially, whereas all the virtual users are spawned in parallel.
 - t represents time taken for a key operation when N virtual users are running in parallel.

Key Upload

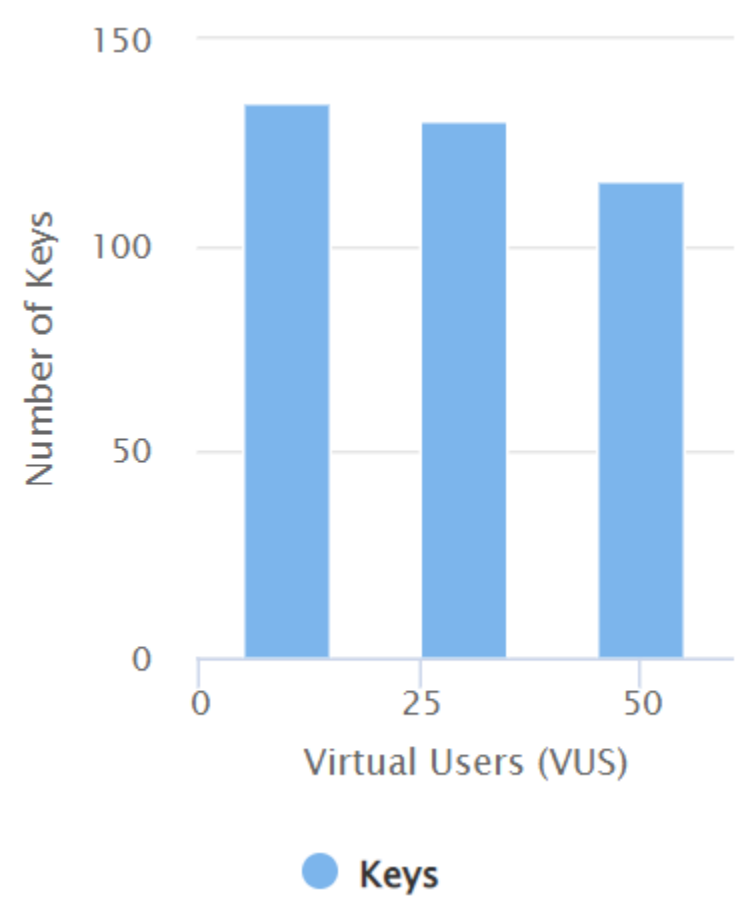
This section provides performance numbers for uploading keys by a number (N) of virtual users in time (T).

Expand the headings below to view performance numbers.

Number of keys uploaded by virtual users in 100 seconds

Virtual Users	Number of Keys
10	135

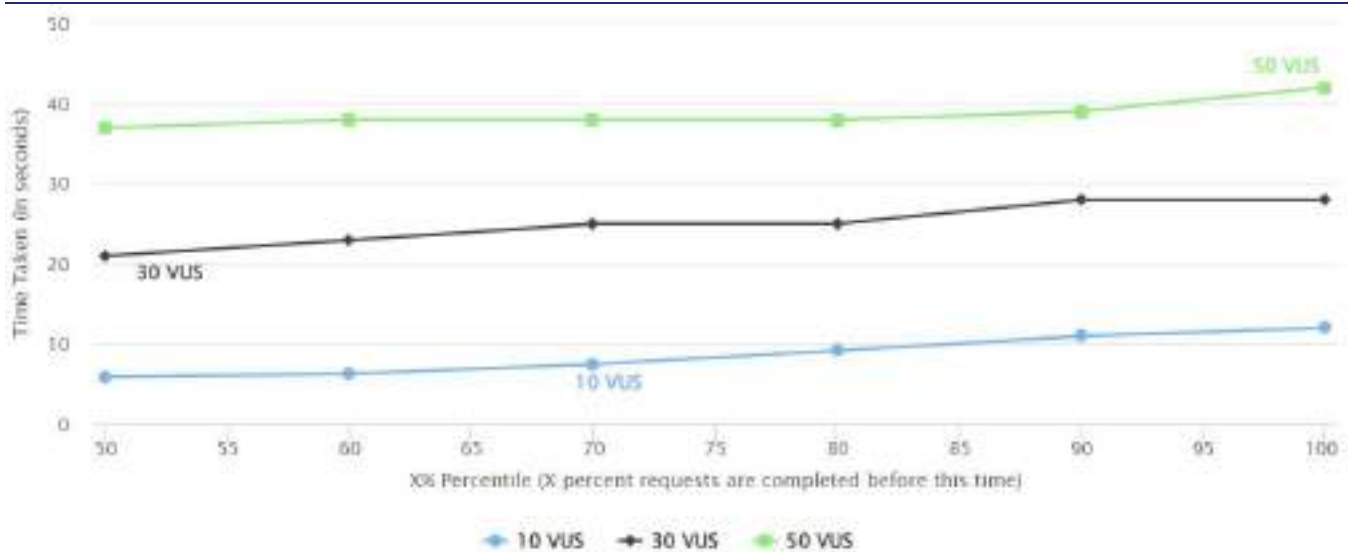
Virtual Users	Number of Keys
30	130
50	116



The graph above shows the number of keys uploaded in t time when N virtual users are spawned in parallel. Also, it indicates throttling condition breakpoints and helps you calculate throughput per second (TPS).

Response for upload with percentile request coverage

X% percentile	10 VUS	30 VUS	50 VUS
50	5.9	21	37
60	6.3	23	38
70	7.5	25	38
80	9.2	25	38
90	11	28	39
100	12	28	42



In the graph above:

- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
- Y-axis represents the time taken by a virtual user to perform the BYOK (upload key) operation.
- While calculating TPS, Locust informs about the time taken to complete X% requests when all the virtual users are spawned in parallel.
- The number of virtual users does not significantly affect the number of keys being uploaded. This proves the rate limit implemented by Azure side causes throttling.

Allowing AD Users to Manage Azure Vaults

This section describes how to allow AD users to manage Azure vaults using CCKM. Change the sample values according to your requirements.

Note

This feature works in the root domain only.

1. Create an LDAP connection (for example, `ldapcon`) on the CipherTrust Manager.

```
curl -k '<IP>/v1/usermgmt/connections' -H 'Authorization: Bearer AUTHTOKEN' -H 'Content-Type: application/json' -H 'accept: application/json' --data-binary $'{\n "name": "ldapcon",\n "strategy": "ldap",\n "options": {\n "server_url": "ldap://<ldap-server-ip>\",\n "root_dn": "dc=calculate",\n "uid_field": "ui
```



```
d",\n "user_dn_field": "dn",\n "bind_dn": "cn=ldapadmin,dc=calculate",\n "bind_password": "<password>",\n "group_base_dn": "dc=calculate",\n "group_member_field": "memberUid",\n "group_id_field": "cn",\n "group_filter": "objectClasses=posixGroup",\n "guid_field": "uid"\n }\n }
```

Here, `server_url` is the URL of the LDAP server.

2. Create a local group (for example, `localgroup`) on the CipherTrust Manager.

```
curl -k '<IP>/v1/usermgmt/groups' -H 'Authorization: Bearer AUTHTOKEN' -H 'Content-Type: application/json' -H 'accept: application/json' --data-binary '${\n "name": "localgroup"\n }' --compressed
```

3. Map the LDAP group on LDAP server (for example, `ldapgroup`) to the newly created `localgroup`.

```
curl -k '<IP>/v1/usermgmt/groupmaps' -H 'Authorization: Bearer AUTHTOKEN' -H 'Content-Type: application/json' -H 'accept: application/json' --data-binary '${\n "connection_name": "ldapcon",\n "connection_group_name": "ldapgroup",\n "group_name": "localgroup"\n }' --compressed
```

4. Map the LDAP group on LDAP server (for example, `ldapgroup`) to the "CCKM Users" group.

```
curl -k '<IP>/v1/usermgmt/groupmaps' -H 'Authorization: Bearer AUTHTOKEN' -H 'Content-Type: application/json' -H 'accept: application/json' --data-binary '${\n "connection_name": "ldapcon",\n "connection_group_name": "ldapgroup",\n "group_name": "CCKM Users"\n }' --compressed
```

5. Add an Azure connection. Refer to [Microsoft Azure](#) for details on required fields.
6. Add an Azure vault. Refer to [Adding Existing Vaults](#) for details.
7. Assign the `localgroup` to this vault.
8. Log on to the Ciphertrust Manager using `ldapgroup` over the LDAP connection, `ldapcon`.

```
username: ldapcon/<ldap-user-name>
password: <ldap-user-login-password>
```

The newly added Azure vault should be visible.

Configuring Connection to Azure Using Private Link

This section describes how to configure connection to azure using [private link](#).

Prerequisites

1. Integrate the Azure key vault with the Azure private link
2. Add a Microsoft Azure connection to the CipherTrust Manager

Adding the Integrated Azure Key Vault to CCKM

You need to add the integrated Azure key vault to CCKM. Refer to [Adding the existing Azure vaults](#) for details.

When adding the Azure key vault, if you do not see the name of the vault, there is missing direct connectivity or DNS resolution failure. You can resolve the DNS resolution failure by adding a DNS host.

To add a DNS host.

1. Log on to the **CipherTrust Manager** GUI.
2. In the left pane, click **Admin Settings > DNS Hosts**.
3. Click **Add DNS Host**.
4. Enter **Hostname**.
5. (Optional) Select either **Address** or **Host-Record**.
6. Enter **DNS IP Address**, it can be either a private IP or a public IP.
7. Click **Add DNS Host**.

Downloading Azure Metadata

Downloading Azure Keys Metadata

CCKM users with permissions to view the Azure keys can export and download the keys metadata displayed on the CCKM GUI. Filter the keys metadata to display on the GUI based on your requirements, for example, all keys created in the past 12 months, all keys residing in a specific vault, and all keys based on a specific algorithm. You can then download the filtered metadata as a CSV file.

To download the Azure keys metadata visible:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Azure**. The Keys tab of the Azure page is displayed.
3. Filter the keys metadata based on your requirements.

Use the Search box to filter the keys. To further filter the metadata, show/hide the desired columns and use the column-level filters.

4. Click **Download as CSV**.

The metadata of Azure keys and their key versions visible on the Keys tab is downloaded as a CSV file.

Downloading Azure Key Vaults Metadata

CCKM users with permissions to view the Azure key vaults can export and download the vaults metadata displayed on the CCKM GUI. Filter the vaults metadata to display on the GUI based on your requirements. You can then download the filtered metadata as a CSV file.

Metadata of a maximum of 100 Azure key vaults can be exported and downloaded.

To download the Azure key vaults metadata visible:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Azure Key Vaults**. The Vaults tab of the Azure Key Vaults page is displayed.
3. Filter the key vaults metadata based on your requirements.

Use the Search box to filter the key vaults. To further filter the metadata, show/hide the desired columns and use the column-level filters.

4. Click **Download as CSV**.

The metadata of Azure key vaults visible on the Vaults tab is downloaded as a CSV file.

Microsoft Double Key Encryption (DKE) Resources

CCKM provides support for Double Key Encryption (DKE) for Microsoft 365. With DKE, two keys are used to access content (such as files or documents) that is protected with the sensitivity labels that you apply when using any one of the supported Microsoft Office apps. You hold one key in CCKM, which you maintain full control over using Microsoft's Double Key Encryption service. The other is securely stored within Microsoft Azure. To decrypt and gain access to the protected content, both keys are required. Since Microsoft holds only one of the keys, this solution ensures that Microsoft cannot independently access this content. Only your authorized users can access the protected content.

Only your highly confidential data should be protected by DKE, which is typically a small percentage of your organization's overall data. For more information about DKE, guidance on what type of content to protect, the supported office applications, and other information, refer to [Double Key Encryption](#).

DKE encryption workflow

1. User adds a DKE label to a document. The Microsoft Office client connects to Azure RMS (Protection Service).

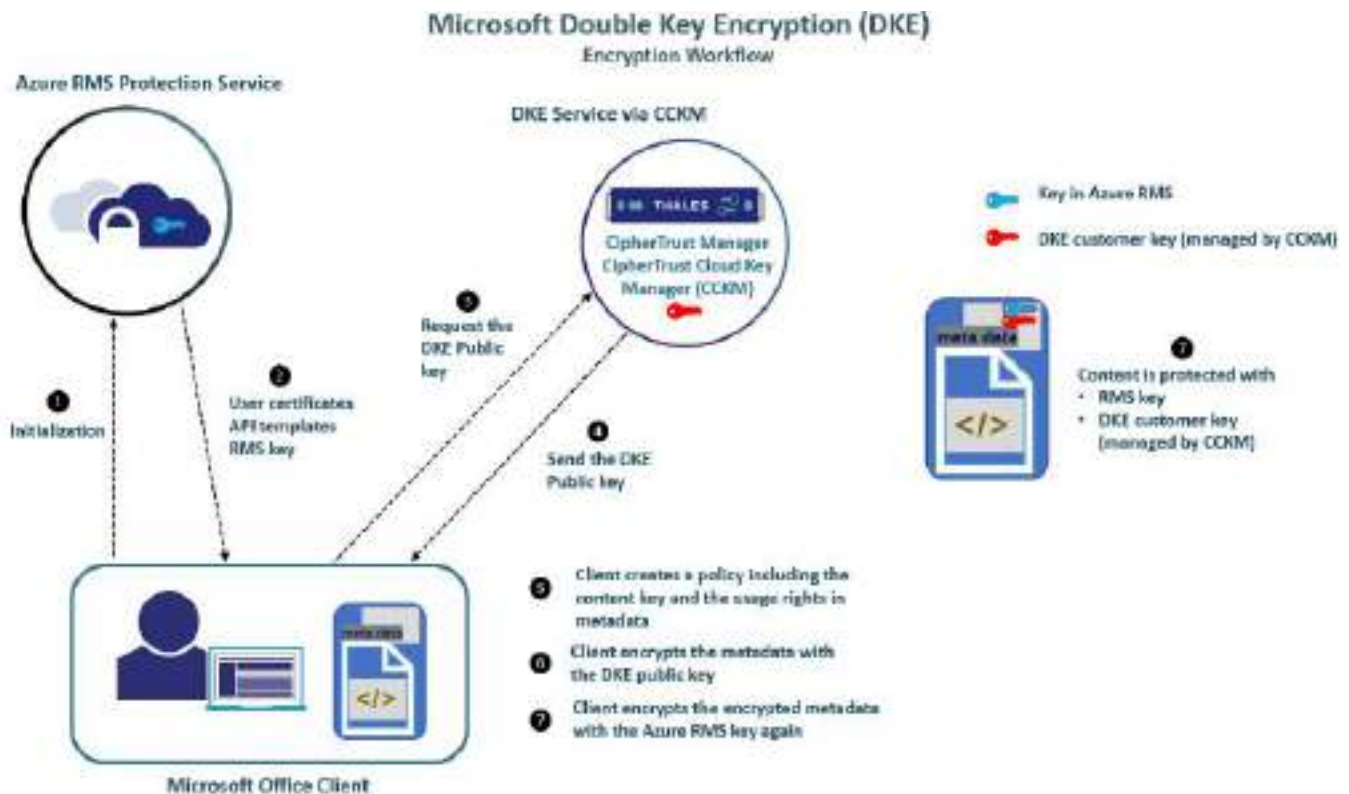
Note

For detailed information on DKE labeling requirements, see [DKE labeling requirements for Office apps](#).

To learn about the minimum supported versions for sensitivity labels, see [Minimum versions for sensitivity labels in Office apps](#).

2. User certificate, API templates, and Azure RMS key are sent to the Microsoft Office client.
3. Microsoft Office client requests CCKM for the DKE public key associated with the applied label.
4. CCKM verifies the request. After authorization succeeds, CCKM sends the DKE public key to the Microsoft Office client.
5. Microsoft Office client encrypts the document content with the content key.
6. Microsoft Office client encrypts the part of the metadata that controls access to the content with the DKE public key.
7. Microsoft Office client encrypts the encrypted metadata with the Azure RMS key.

The content is protected with the double-encrypted content key, RMS key and DKE public key. The following illustration shows the DKE encryption workflow:



DKE decryption workflow

1. User opens a document which has a DKE label. The Microsoft Office client first sends the double-encrypted metadata to Azure RMS (Protection Service).

2. Azure RMS (Protection Service) decrypts the double-encrypted metadata using the Azure RMS key and returns the simple-encrypted metadata.
3. Microsoft Office client sends the simple-encrypted metadata to CCKM and requests CCKM to decrypt it.
4. CCKM verifies the request and goes through the authorization process. When authorization passes successfully, CCKM decrypts the simple-encrypted metadata and returns the decrypted metadata to the Microsoft Office client.
5. Microsoft Office client uses the content key to decrypt the document. The user can view the document content.

The following illustration shows the DKE decryption workflow:

The following are the high-level steps to configure DKE within Microsoft and CipherTrust Manager/CCKM:

1. [Complete the prerequisites.](#)
2. [Configure DKE on CCKM.](#)
3. [Register your DKE service from Microsoft Azure portal.](#)

Prerequisites

In preparation for deploying DKE, ensure that the following prerequisites are met:

1. Ensure CipherTrust Manager/CCKM is installed and running. In addition, ensure to activate and install the CCKM license. Refer to the [CipherTrust Manager Deployment Guide](#) and [Licensing](#) for details.
2. The NTP server is added. Refer to [Add an NTP Server](#) for details.
3. Install a third-party CA certificate on the web interface.

Note

We recommend obtaining a CA-signed certificate for the web interface from trusted CAs that are part of the Microsoft Trusted Root Program. For more information, refer to [Microsoft Trusted Root Program participants](#).

4. Ensure the following.
 - If you have a load balancer that terminates the TLS connection and has mTLS disabled, the load balancer has a valid TLS certificate signed by a trusted CA.
 - If you have a load balancer that doesn't terminate the TLS connection or you don't have a load balancer in front of CipherTrust Manager, CipherTrust Manager has a valid TLS certificate signed by a trusted CA.

- Additional DNS record is configured with **dke.** prefix and pointing to the same IP Address of the CipherTrust Manager or SSL Passthrough Load Balancer.
 - Subject Alternative Name (SAN) in the TLS certificate includes **dke.** prefixed DNS name, along with other DNS names associated with CipherTrust Manager or SSL Passthrough Load Balancer.
5. **Microsoft 365 E5 License:** Windows clients must have a Microsoft 365 E5 license. Refer to information about [system and licensing requirements for DKE](#) for more information these licensing requirements.

Note

CCKM supports built-in sensitivity labeling with Office apps. CCKM does not support Azure Information Protection unified labeling client, which is to be deprecated in the future.

6. **User Presence in Azure Directory:** Users who have to create and share documents or files protected by DKE must be part of the Azure Active Directory (Azure AD) associated with the organization's tenant.

Note

- DKE is tested on the Windows and iOS devices. It is not tested on macOS and Android devices.
- To configure the B2B cross-tenant access settings, refer to [Configure cross-tenant access settings for B2B collaboration](#).

You can now Configure DKE on CCKM.

Configure DKE on CCKM

To configure DKE on CCKM, do the following:

1. From the CCKM GUI or API, create a DKE endpoint. Refer to [Creating Microsoft DKE Endpoints](#) for details on how to create a DKE endpoint from the CCKM GUI or [Creating Microsoft DKE Endpoints using the CCKM API](#).

Note

- Only Azure AD users are currently supported. The email addresses included in the API call must be associated with a user Azure AD. The email address parameter an optional field applicable only if `authorization_type` is set to email.

If you specify one or more email addresses in a DKE endpoint, this field is validated against the User Principal Name (UPN) claim in the JWT received during the decrypt operation. The UPN is the unique sign-in identifier for each user in Microsoft Entra ID. For guest access, where the UPN claim might not be present in the token, the 'email' claim in the token is used instead.

- A Microsoft Office 365 client user and email (if configured in CipherTrust Manager) must match.
- The *Issuer* field must be specific to the tenant configured in the Microsoft Azure Portal.
- The web interface port within CipherTrust Manager can be changed from the default port of 443 to another port. If you plan to change the default port for the CipherTrust Manager web interface, ensure to change it BEFORE configuring the Microsoft DKE service on CCKM. Also reflect this port change when creating a DKE endpoint on CCKM as part of the prerequisites for deploying DKE. **Changing the default port AFTER configuring Microsoft DKE is not supported.**

Register DKE Service

After creating a DKE endpoint, register your DKE service from the Microsoft Azure portal.

Note

- When registering your DKE service, ensure that **Application ID URI** matches the FQDN of the CipherTrust Manager. In the case where you are using a load balancer in front of the CipherTrust Manager, ensure that Application ID URI matches the FQDN of the load balancer. You can only register one DKE service per one instance of CipherTrust Manager.
- Also, ensure that the publisher domain on the *Branding and Properties* page is set to the domain in which CipherTrust Manager is deployed. For example, if CipherTrust Manager is `test.thalescpl.io`, then ensure that the publisher domain includes `thalescpl.io`.

Single Tenant Application

When to Use the Single Tenant Application

Single tenant applications are only available in the tenant they were registered in, also known as their home tenant. DKE-labeled data will only be used within home tenant, configured with one CCKM environment.

For more details, refer to [Tenancy in Microsoft Entra ID](#).

Register a Single Tenant Application

To register a single tenant application.

1. Sign in to the [Microsoft Azure portal](#).
2. Under Azure services, click **App registrations**.
3. Click **New registration**.
4. Enter **Name**, like SingleTenantApplication.
5. Select **Accounts in any organizational directory (Any Microsoft Entra ID tenant - Single Tenant)** as the account type.
6. Under **Redirect URI**:
 - a. Select **Web** as the platform.
 - b. Enter the **URI** of CipherTrust Manager, like `https://dke.thalescpl.io`.
7. Click **Register**.

Copy the **Application (client) ID** and **Directory (tenant) ID**.

Now configure the single tenant application that includes the following.

1. [Authenticate Tokens and ID Tokens](#)
2. [Adding a Certificate or Secret](#).

Note

- It is optional to add a certificate or secret.
- You will get the credentials after adding a certificate or secret. You will need the credentials during the role-based authentication.

3. [Adding an API permission](#).
4. [Exposing an API](#).
5. [Adding the Owner](#).

Note

It is optional to add the owner.

Authenticate Tokens and ID Tokens

To authenticate the tokens and ID tokens.

1. Go to the left pane, click **Manage > Authentication**.
2. Enable **Access tokens**
3. Enable **ID Tokens**.
4. Click **Save**.

Adding a Certificate or Secret

To add a certificate or secret.

1. Go to the left pane, click **Manage > Certificates & secrets**.
2. Click **New client secret**. The Add a client secret wizard is displayed.
3. Enter **Description**.
4. (Optional) Select the desired **Expires** (Expiry days) from the drop-down list.
5. Click **Add**.

Copy the **Value** and **Secret ID** of the secret. You will need the Value and Secret ID to [add an Azure connection on CCKM](#).

Adding an API Permission

For role-based authorization, the **User.Read.All** and **RoleManagement.Read.Directory** permissions are required to retrieve user details and their assigned roles.

To add an API permission.

1. Go to the left pane, click **Manage > API permissions**.
2. Click **Add a permission**. The Request API permissions wizard is displayed.
3. Click **Microsoft Graph**.
4. Click **Application permissions**.
5. Add the **User.Read.All** and **RoleManagement.Read.Directory** permissions.

To add the **User.Read.All** permission.

- a. In the Select permissions search box, search for **User**.
- b. Expand **User** from the result.

-
- c. Select **User.Read.All** from the User list.

To add the **RoleManagement.Read.Directory** permission.

- a. In the Select permissions search box, search for **RoleManagement**.
 - b. Expand **RoleManagement** from the result.
 - c. Select **RoleManagement.Read.Directory** from the RoleManagement list.
6. Click **Add permissions**.
 7. Click **Grant admin consent for {tenant}**. The Grant admin consent confirmation dialog box is displayed.
 8. Click **Yes**.

Note

If you do not have the admin rights, please ask your admin to grant the admin consent. You need to share the consent grant URI to your admin, like https://login.microsoftonline.com/{organization}/adminconsent?client_id={client-id}. In the URI, the {organization} is Directory (tenant) ID and {client-id} is Application (client) ID that you copied during the registration.

Exposing an API

Exposing an API includes adding the Application ID URI, adding a scope, and authorizing the client applications.

Adding the application ID URI.

1. Go to the left pane, click **Manage > Expose an API**.
2. Click **Add**. The Edit application ID URI is displayed.
3. Enter **Application ID URI**, the URI of the CipherTrust Manager.
4. Click **Save**.

Adding a scope.

1. Click **Add a scope**. The Add a scope wizard is displayed.
 2. Enter **Scope name**, like "user_impersonation".
 3. Select the Consent as **Admins and users**.
 4. Enter **Admin consent display name**.
 5. Enter **Admin consent description**.
-

-
6. (Optional) Enter **User consent display name**.
 7. (Optional) Enter **User consent description**.
 8. (Optional) Enable **State**.
 9. Click **Add scope**.

Authorizing the client applications.

You need to authorize the Microsoft Office and Azure Information Protection applications. Perform the below steps for both the applications.

1. Click **Add a client application**. The Add a client application wizard is displayed.
2. Enter **Client ID**.

Note

The Client ID of Microsoft Office and Azure Information Protection applications are d3590ed6-52b3-4102-aeff-aad2292ab01c and c00e9d32-3c8d-4a7d-832b-029040e7db99, respectively.

3. Enable **Authorized scopes**.
4. Click **Add application**.

Adding the Owner

To add the owner.

1. Go to the left pane, click **Manage > Owners**.
2. Click **Add owners**.
3. Search and select the desired owner.
4. Click **Select**.

Multitenant Application

When to Use the Multitenant Application

Multitenant applications are available to the users in both their home tenant and other tenants. Multitenant application can be used for the following.

- Cross-tenant access for B2B collaboration.

- Sharing one DKE endpoint across multiple Azure tenants.
- Using one CCKM environment for supporting DKE in multiple Azure tenants. For example Managed Service Providers need to create a multitenant application when their customers share one CCKM environment.

For more details, refer to [Tenancy in Microsoft Entra ID](#).

Register a Multitenant Application

Note

To trust the multitenant application, the admin of other tenant needs to add this application to their enterprise applications. The the admin of other tenant should enter and accept the consent grant URI, like `https://login.microsoftonline.com/{organization}/adminconsent?client_id={client-id}`.

Here, the {organization} is the **Directory (tenant) ID** of the other tenant and {client-id} is the **Application (client) ID** of the home tenant.

To create a multitenant application.

1. Sign in to the [Microsoft Azure portal](#).
2. Go to the left pane, click **Authentication**.
3. Click **App registrations**.
4. Click **New registration**.
5. Enter **Name**, like MultiTenantApplication.
6. Select **Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)** as the account type.
7. Under **Redirect URI**:
 - a. Select **Web** as the platform
 - b. Enter the **URI** of CipherTrust Manager, like `https://dke.thalescpl.io`.
8. Click **Register**.

Copy the **Application (client) ID** and **Directory (tenant) ID**.

Now configure the multitenant application that includes the following.

1. [Authenticate Tokens and ID Tokens](#)
2. [Adding a Certificate or Secret](#).

Note

- It is optional to add a certificate or secret.
- You will get the credentials after adding a certificate or secret. You will need the credentials during the role-based authentication.

3. Adding an API permission.

4. Exposing an API.

5. Adding the Owner.

Note

It is optional to add the owner.

Authenticate Tokens and ID Tokens

To authenticate the tokens and ID tokens.

1. Go to the left pane, click **Manage > Authentication**.
2. Enable **Access tokens**
3. Enable **ID Tokens**.
4. Click **Save**.

Adding a Certificate or Secret

To add a certificate or secret.

1. Go to the left pane, click **Manage > Certificates & secrets**.
2. Click **New client secret**. The Add a client secret wizard is displayed.
3. Enter **Description**.
4. (Optional) Select the desired **Expires** (Expiry days) from the drop-down list.
5. Click **Add**.

Copy the **Value** and **Secret ID** of the secret. You will need the Value and Secret ID to [add an Azure](#)

connection on CCKM.

Adding an API Permission

For role-based authorization, the `User.Read.All` and `RoleManagement.Read.Directory` permissions are required to retrieve user details and their assigned roles.

To add an API permission.

1. Go to the left pane, click **Manage > API permissions**.
2. Click **Add a permission**. The Request API permissions wizard is displayed.
3. Click **Microsoft Graph**.
4. Click **Application permissions**.
5. Add the `User.Read.All` and `RoleManagement.Read.Directory` permissions.

To add the `User.Read.All` permission.

- a. In the Select permissions search box, search for **User**.
- b. Expand **User** from the result.
- c. Select **User.Read.All** from the User list.

To add the `RoleManagement.Read.Directory` permission.

- a. In the Select permissions search box, search for **RoleManagement**.
- b. Expand **RoleManagement** from the result.
- c. Select **RoleManagement.Read.Directory** from the RoleManagement list.
6. Click **Add permissions**.
7. Click **Grant admin consent for {tenant}**. The Grant admin consent confirmation dialog box is displayed.
8. Click **Yes**.

Note

If you do not have the admin rights, please ask your admin to grant the admin consent. You need to share the consent grant URI to your admin, like `https://login.microsoftonline.com/{organization}/adminconsent?client_id={client-id}`. In the URI, the {organization} is **Directory (tenant) ID** and {client-id} is **Application (client) ID** that you copied during the registration.

Exposing an API

Exposing an API includes adding the Application ID URI, adding a scope, and authorizing the client applications.

Adding the application ID URI.

1. Go to the left pane, click **Manage > Expose an API**.
2. Click **Add**. The Edit application ID URI is displayed.
3. Enter **Application ID URI**, the URI of the CipherTrust Manager.
4. Click **Save**.

Adding a scope.

1. Click **Add a scope**. The Add a scope wizard is displayed.
2. Enter **Scope name**, like "user_impersonation".
3. Select the Consent as **Admins and users**.
4. Enter **Admin consent display name**.
5. Enter **Admin consent description**.
6. (Optional) Enter **User consent display name**.
7. (Optional) Enter **User consent description**.
8. (Optional) Enable **State**.
9. Click **Add scope**.

Authorizing the client applications.

You need to authorize the Microsoft Office and Azure Information Protection applications. Perform the below steps for both the applications.

1. Click **Add a client application**. The Add a client application wizard is displayed.
2. Enter **Client ID**.

Note

The Client ID of Microsoft Office and Azure Information Protection applications are d3590ed6-52b3-4102-aeff-aad2292ab01c and c00e9d32-3c8d-4a7d-832b-029040e7db99, respectively.

3. Enable **Authorized scopes**.
4. Click **Add application**.

Adding the Owner

To add the owner.

1. Go to the left pane, click **Manage > Owners**.
2. Click **Add owners**.
3. Search and select the desired owner.
4. Click **Select**.

Note

Microsoft DKE supports guest accounts for external users to open the encrypted documents. For more details, refer to [Guest accounts for external users to open encrypted documents](#).

Managing DKE Endpoints

This section describes how to manage DKE endpoints from CCKM.

Microsoft Double Key Encryption (DKE) service can access CipherTrust Cloud Key Manager (CCKM) after you meet some prerequisites. Refer to [Microsoft Double Key Encryption \(DKE\) Resources](#) for more information about Microsoft DKE and the prerequisites.

Part of the prerequisites is [creating a DKE endpoint](#). After you have created a DKE endpoint, you can:

- [View DKE endpoints](#)
- [View and edit details of a DKE endpoint](#)
- [Enable a DKE endpoint](#)
- [Disable a DKE endpoint](#)
- [Rotate a DKE endpoint](#)
- [Delete a DKE endpoint](#)
- [Archive a DKE endpoint](#)
- [Recover a DKE endpoint](#)

Note

Users of the CCKM Admins group can manage DKE endpoints, whereas the users of the CCKM Users group can only view the DKE endpoints.

Caution

Modify endpoints only through the CipherTrust Cloud Key Management (CCKM) GUI, API, or CLI. Do not modify the KEK using the **Keys** menu, the `ksctl keys` command, or the `/v1/vault/keys2` endpoint. Such modifications might make the KEK unavailable to the Microsoft DKE service.

Creating DKE Endpoints

From CCKM GUI, create a DKE endpoint to make it available to the Microsoft DKE service. You can either create a new key encryption key (KEK) or select an existing KEK for the getkey and decrypt operations. If you do not specify a key, CCKM will automatically create a CipherTrust Manager key for you.

Note

- Ensure that you have already created an authorized tenant, refer to [Managing DKE Authorized Tenants](#) for details.
- If you plan to use the authorization method of Role ID for the access settings of DKE keys, you are required to set up an Azure connection before creating a DKE endpoint. For more information on how to add an Azure connection, refer to [Microsoft Azure Connection](#).
- In the Microsoft Azure application, go to the API Permissions section and configure the **Directory.Read.All** permission. Ensure that the **Type** for this permission is Application.



API / Permission name	Type	Description	Admin consent requ...	Status
no Microsoft Graph (0)				See...
Directory.Read.All	Application	Read directory data	Yes	Consent required

To create a DKE endpoint from the CCKM GUI:

1. Log on to the **CipherTrust Manager** GUI as a member of the CCKM Admins group.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **Services > Microsoft DKE**. The Microsoft Double Key Encryption (DKE) page is

displayed.

4. Click **Create Endpoint**. The **Add Double Key Encryption (DKE) Endpoint** dialog box is displayed.
5. Under **General Info**, enter a unique **Name** and a description for the endpoint.
6. Click **Next**.
7. Under **Endpoint Settings**, provide the following settings:
 - **Key URI Hostname**. Enter the hostname at which CipherTrust Manager will be visible to Microsoft.
 - The Key URI Hostname will be used to construct the URL Microsoft will use to send requests to CipherTrust Manager.
 - The web interface of the CipherTrust Manager must have a valid TLS certificate signed by a trusted CA.
 - If the SSL termination takes place at the CipherTrust Manager, the Key URI Hostname must match the FQDN of the CipherTrust Manager. If the SSL termination takes place at the load balancer in front of the CipherTrust Manager, the Key URI Hostname must match the FQDN of the public-facing load balancer. Wildcard certificates are supported.
 - If you changed the web interface port within the {cm} from the default port (443) to another port (say 8443), add the same port when configuring the Key URI Hostname. Specify the new port number with the Key URI Hostname, for example, `example.com:8443`.
 - If you have a load balancer with SSL termination and mTLS disabled, the Key URI Hostname must match the FQDN of the public-facing load balancer.
 - If you have a load balancer without SSL termination, the Key URI Hostname must match `dke.<load balancer hostname>`.
 - If you don't have a load balancer in front of CipherTrust Manager, the Key URI Hostname must match `dke.<CipherTrust Manager hostname>`.
 - **Key Algorithm**. The only option available for the DKE key algorithm is `RSA_DECRYPT_OAEP_2048_SHA256`. This box is greyed out.
 - **Enable success audit events**. Use this toggle to enable or disable audit recording of successful Microsoft DKE operations on this endpoint. This is optional. By default, this toggle is turned on.
8. Click **Next**.
9. Under **Key Material**, choose to either create a new key or copy an existing key.

Note

If you do not specify a key, CCKM will automatically create a CipherTrust Manager key for you.

If creating a new key, select **Create New Key**, and enter the **Key Name**. If copying an existing key, select **Copy Existing Key**, and select a **CipherTrust (Local) Key** from the list.

Note

For Copy Existing Key, by default, the Algorithm and Key Size will be **RSA** and **2048**, respectively.

10. Click **Next**.
11. Under **Authorized Tenants**, select the Authorized Tenant from the drop-down list.

Note

Select at least one authorized tenant. Multiple authorized tenants with the same issuer are not allowed.

12. Click **Next**.
13. Under **Review and Add**, review the details you provided for the new endpoint. These details are divided into GENERAL INFO, ENDPOINT SETTINGS, KEY MATERIAL, and AUTHORIZED TENANTS sections.

Note

After the endpoint is added, KEY MATERIAL will no longer be editable.

14. Click **Add Endpoint**.

The endpoint creation starts. A Create Endpoint In Progress message is displayed on the screen. Leave the window open until the process is completed.

15. Click **Close**. The Add Double Key Encryption (DKE) Endpoint wizard is closed.

The newly created endpoint is displayed in the list of Microsoft DKE endpoints.

Viewing DKE endpoints

The **Microsoft Double Key Encryption (DKE)** page shows the list of existing DKE endpoints residing within a given URI hostname. Search for endpoints by endpoint name or key URI.

To view the list of DKE endpoints available on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > Microsoft DKE**. The Microsoft Double Key Encryption (DKE) page is displayed. The list of DKE endpoints added to CCKM is displayed. The page displays the following details:

Field	Description
Name	Name of the DKE endpoint.
Key URI Hostname	Base url hostname for KeyURI.
Status	State of the key. The status can be: • Enabled • Disabled
KEK Name	Name of KEK.
Key Version	Version of the key.
Algorithm	DKE key algorithm. The only option currently supported is RSA_DECRYPT_OAEP_2048_SHA256, which is the default value.
Authorization Method	Authorization type for DKE key: Email or Role ID.
Creation Date	Time when the endpoint was created.
Last Modified	Date and time the DKE endpoint was modified. Timestamp in format Day-Month-Year time in 24-hour notation.
Description	Description for the endpoint.

To view the custom columns, click the Customize View (⚙️) icon, select the desired option(s), and click **OK** to display the column(s).

Viewing and editing details of a DKE endpoint

After an endpoint is created, you can view and modify the endpoint details, such as the endpoint name and authorization parameters.

This section describes how to view the details of a DKE endpoint and update details relating to GENERAL INFO, ENDPOINT SETTINGS, and AUTHORIZATION PARAMETERS as needed.

To view and edit the details of Microsoft DKE endpoints on CCKM:

-
1. Log on to the **CipherTrust Manager** GUI as a member of the CCKM Admins group.
 2. Open the **Cloud Key Manager** application.
 3. In the left pane, click **Services > Microsoft DKE**. The Microsoft Double Key Encryption (DKE) page is displayed. The list of DKE endpoints added to CCKM is displayed.
 4. Click the **Name** link of the desired DKE endpoint.
Alternatively, click the overflow icon () corresponding to the desired DKE endpoint, and click **View/Edit**.

The edit view of the Microsoft Double Key Encryption (DKE) page is displayed.

Under VERSIONS

1. (Optional) Click **Rotate** to create a new version of the endpoint's encryption key. The Rotate Endpoint dialog box will open.
2. Click **Rotate**.

Under GENERAL INFORMATION

1. (Optional) Update the description of the DKE endpoint in the **Description** field.
2. Click **Update**.

Under ENDPOINT SETTINGS

1. (Optional) Enter a new **Key URI hostname** to use a different URI hostname.
2. (Optional) Depending on current setting, set the **Enable success audit events** toggle to enable or disable audit recording of successful operations related to Microsoft DKE operations on this endpoint.
3. Click **Update**.

Under AUTHORIZED TENANTS

(Optional) Add Authorized Tenants

1. Click **Add Authorized Tenants** to add multiple authorized tenants. The Add Authorized Tenants wizard will open.
2. Select the desired Authorized Tenant(s) from the list.
3. Click **Add Authorized Tenants**.

(Optional) Remove Authorized Tenants

Click the overflow icon () corresponding to the desired authorized tenant, and click **Remove**.

(Optional) View or Edit Authorized Tenants

Click the overflow icon () corresponding to the desired authorized tenant, and click **View/Edit**. Refer to [Managing DKE Authorized Tenants](#).

Under KEY SCHEDULE

1. (Optional) Select a **Rotation** from the drop-down list.
2. Click **Update**.

Enabling a DKE endpoint

From the Microsoft Double Key Encryption (DKE) page, you can enable a DKE endpoint. Enabling a DKE endpoint allows the getkey and decrypt/unwrap operations for the given Microsoft DKE endpoint.

To enable a DKE endpoint on CCKM:

1. Click the **Name** link of the desired DKE endpoint you wish to enable.
The edit view of the Microsoft Double Key Encryption (DKE) page displays.
2. Under the **Actions** drop-down menu, select **Enable**. The **Enable Endpoint** dialog box displays.

Alternatively, click the overflow icon () corresponding to the desired DKE endpoint, and click **Enable**. The **Enable Endpoint** dialog box displays.

3. Click **Enable** to proceed. After the endpoint is successfully enabled, a message displays indicating "Successfully enabled Microsoft DKE endpoint".

Disabling a DKE endpoint

From the Microsoft Double Key Encryption (DKE) page, you can disable a DKE endpoint. Disabling a DKE endpoint disallows the getkey and decrypt/unwrap operations for the given Microsoft DKE endpoint.

To disable a DKE endpoint on CCKM:

1. Click the **Name** link of the desired DKE endpoint you wish to disable.
The edit view of the Microsoft Double Key Encryption (DKE) page displays.
2. Under the **Actions** drop-down menu, select **Disable**. The **Disable Endpoint** dialog box displays.

Alternatively, click the overflow icon () corresponding to the desired DKE endpoint, and click **Disable**. The **Disable Endpoint** dialog box displays.

3. Click **Disable** to proceed. After the endpoint is successfully disabled, a message displays indicating "Successfully disabled Microsoft DKE endpoint".

Rotating a DKE endpoint

From the Microsoft Double Key Encryption (DKE) page, you can rotate a DKE endpoint, which adds a new KEK version to this endpoint. Subsequent encrypt/wrap operations this endpoint performs will use the new version of the key. Decrypt/Unwrap operations will use whichever version of the key was originally used to encrypt/wrap.

Rotating a DKE endpoint's KEK can be done even if the endpoint is disabled. Rotating the KEK regularly is a security best practice.

To rotate a DKE endpoint on CCKM:

1. Click the **Name** link of the desired DKE endpoint you wish to rotate.
The edit view of the Microsoft Double Key Encryption (DKE) page displays.
2. Under the **VERSIONS**, click **Rotate**. The **Rotate Endpoint** dialog box displays.
Alternatively, click the overflow icon () corresponding to the desired DKE endpoint, and click **Rotate**. The **Rotate Endpoint** dialog box displays.
3. Click **Rotate** to proceed. After the endpoint is successfully rotated, a message displays indicating "Successfully rotated Microsoft DKE endpoint". The new key version displays in list of key versions under the **VERSIONS** of the edit view of the Microsoft Double Key Encryption (DKE) page.

Deleting a DKE Endpoint

From the Microsoft Double Key Encryption (DKE) page, you can delete a DKE endpoint. The getkey and decrypt operations associated with this endpoint are also deleted from the database as part of the deletion. Essentially, the record of this endpoint is deleted from the database and CCKM. A deleted endpoint is no longer included in the list DKE endpoints displayed on the Microsoft Double Key Encryption (DKE) page.

Note

Before deleting a DKE endpoint, you must first archive it.

To delete a DKE endpoint on CCKM:

1. Click the **Name** link of the desired DKE endpoint you wish to delete.
The edit view of the Microsoft Double Key Encryption (DKE) page displays.
2. Under the **Actions** drop-down menu, select **Delete**. The **Delete Endpoint** dialog box displays.

Alternatively, click the overflow icon () corresponding to the desired DKE endpoint, and click **Delete**. The **Delete Endpoint** dialog box displays.
3. Enable the **I wish to delete this endpoint** checkbox and click **Delete**. After the endpoint is successfully deleted, a message displays indicating "Successfully deleted Microsoft DKE endpoint".

Archiving a DKE Endpoint

From the Microsoft Double Key Encryption (DKE) page, you can archive a DKE endpoint. Archiving allows you preserve a record of this endpoint in "Archived" state, which can be recovered thereafter.

Note

- An archived DKE endpoint does not consume a CCKM license.
- When a key is in archived state, it can be viewed, recovered, or deleted. It is included in the list of DKE endpoints on the **Microsoft Double Key Encryption (DKE)** page.
- A new KEK version cannot be added to an Archived DKE endpoint as this endpoint cannot be rotated.

To archive a DKE endpoint on CCKM:

1. Click the **Name** link of the desired DKE endpoint you wish to archive.
The edit view of the Microsoft Double Key Encryption (DKE) page displays.
2. Under the **Actions** drop-down menu, select **Archive**. The **Archive Endpoint** dialog box displays.

Alternatively, click the overflow icon () corresponding to the desired DKE endpoint, and click **Archive**. The **Archive Endpoint** dialog box displays.

3. Enable the **I wish to archive this endpoint** checkbox and click **Archive**. After the endpoint is successfully archived, a message displays indicating "Successfully archived Microsoft DKE endpoint".

Recovering a DKE Endpoint

From the Microsoft Double Key Encryption (DKE) page, you can recover a DKE endpoint. Recovering a DKE endpoint in an archived state allows you enable the endpoint thereby allowing subsequent requests for the getkey and decrypt/unwrap operations for the endpoint.

Note

An recovered DKE endpoint consumes a CCKM license.

To recover a DKE endpoint on CCKM:

1. Click the **Name** link of the desired DKE endpoint you wish to recover from an archived state.
The edit view of the Microsoft Double Key Encryption (DKE) page displays.
2. Under the **Actions** drop-down menu, select **Recover**. The **Recover Endpoint** dialog box displays.

Alternatively, click the overflow icon () corresponding to the desired DKE endpoint, and click **Recover**. The **Recover Endpoint** dialog box displays.

3. Enable the **I wish to recover this endpoint** checkbox and click **Recover**. After the endpoint is successfully recovered, a message displays indicating "Successfully recovered Microsoft DKE endpoint".

Managing DKE Authorized Tenants

This section describes how to manage DKE authorized tenants from CCKM.

- Create a DKE Authorized Tenant
- View DKE Authorized Tenants
- View and Edit Details of a DKE Authorized Tenant
- Delete a DKE Authorized Tenant

Creating a DKE Authorized Tenant

To create a DKE authorized tenant from the CCKM GUI:

1. Log in to the **CipherTrust Manager** GUI as a member of the CCKM Admins group.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **Services > Microsoft DKE**. The Microsoft Double Key Encryption (DKE) page is displayed.
4. Go to the AUTHORIZED TENANTS section.
5. Click **Create Tenant**. The General Info screen of the **Add Double Key Encryption (DKE) Authorized Tenant** wizard is displayed.

General Info

1. Enter the **Name**
2. Enter the **Description**.
3. Click **Next**. The Issuer Settings screen is displayed.

Issuer Settings

1. Enter a **Valid Issuer**. A trusted issuer for the DKE authorized tenant. An example of a valid issuer is `https://sts.windows.net/<azure_tenant_ID>/`.

Note

The following requirements apply for the issuer:

- Must match the issuer within the JWT that the CCKM receives.

- If the issuer's hostname is `https://sts.windows.net/<azure_tenant_ID>/`, you must include the trailing slash `/`.

2. Click **Next**. The Authorization Parameters screen is displayed.

Authorization Parameters

1. Select an authorization method to the access settings of DKE keys. The options are **Use Email(s)** and **Use Role(s)**.
2. Click the desired tab to view the instructions.

Use Email(s)

Use Email(s) allows your organization to authorize access to the DKE keys based on email addresses.

Under **Add Email(s)**, enter the email addresses that are authorized to use the DKE key and click **Add Email(s)**. The added emails are displayed in the list of Emails.

If you specify one or more email addresses in a DKE endpoint, this field is validated against the User Principal Name (UPN) claim in the JWT received during the decrypt operation. The UPN is the unique sign-in identifier for each user in Microsoft Entra ID. For guest access, where the UPN claim might not be present in the token, the 'email' claim in the token is used instead.

Note

The use of wildcards '*' and '?' are supported for email addresses. The following are examples of supported wildcard formats:

- `abc*@gmail.com`: supports any email that starts with abc and ends with "@gmail.com".
- `abc@?.com`: supports any email that starts with abc, contains "@" followed by at least one character and ends with ".com".
- `abc@?.?*`: supports any email that starts with abc, contains "@" followed by at least one character, followed by the dot character (.), and ends with at least one character.
- `?*@gmail.com`: supports any email that starts with at least one character and ends with "@gmail.com".

Use Role(s)

Use Role(s) allows your organization to authorize access to the DKE keys based on Active Directory groups.

The format is `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` where `x` is an alphanumeric character.

From the **Select Azure Connection** drop-down menu, select an Azure connection to use for this DKE authorized tenant, and add the role IDs that are authorized to use the DKE key.

Note

- An [Azure connection](#) must be created for role-based authorization.
- To create this connection, you can either register a new application on the Azure portal or use the application that was registered for DKE. For more information, refer to [Register DKE Service](#).
- The required API permissions for this connection are `User.Read.All` and `RoleManagement.Read.Directory`.
- If you're selecting role as the authorization method, ensure that the tenant ID from the connection must match the tenant ID in the Issuer field.

You can select the roles from the list and you can add the roles manually. Click the desired tab to view the instructions.

List

Click **List Roles** and select the desired roles from the list.

Manually

Scroll down the page, enter the **Role ID(s)** that are authorized to use the DKE key, and click **Add Role ID(s)**. The added role IDs are displayed in the list of Role IDs.

To find out the role ID in Azure:

- a. Log on to the Microsoft Entra portal.
- b. In the left pane, click **Identity > Users > All users**.
- c. On the Users page, search for the desired user.
- d. Under Display Name, click the user name link. The details of the selected user are displayed.
- e. Click **Assigned roles**. The Active assignments tab shows the roles currently assigned to the user.
- f. Note down the role for which you want to find out the role ID.
- g. In the left pane, click **Roles & admins > Roles & admins**. The Roles and administrators page displays all roles on the right.

- h. Search and select the role you noted down earlier. The "<user role> | Assignments" page is displayed.
- i. Under Manage, click **Description**. The Summary of the role is displayed.
- j. Copy the value of the **Template ID**.

3. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the authorized tenant details that you have provided. These details are divided into GENERAL INFO, ISSUER SETTINGS, and AUTHORIZATION PARAMETERS sections.

Before adding the authorized tenant, review all details. After the authorized tenant is added, certain features will no longer be editable.

1. Review the authorized tenant details displayed on the screen.

If the details are incorrect or you want to make any changes, click **Edit** next to the **GENERAL INFO**, **ISSUER SETTINGS**, and **AUTHORIZATION PARAMETERS** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Authorized Tenant**.

3. Click **Close**.

The newly created authorized tenant is displayed in the list of Microsoft DKE authorized tenants.

Viewing DKE Authorized Tenants

The **Microsoft Double Key Encryption (DKE)** page shows the list of existing DKE authorized tenants residing within a given ID. Search for the authorized tenants by name or ID.

To view the list of DKE authorized tenants available on CCKM:

1. Log in to the **CipherTrust Manager** GUI as a member of the CCKM Admins group.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **Services > Microsoft DKE**. The Microsoft Double Key Encryption (DKE) page is displayed.
4. Go to the **AUTHORIZED TENANTS** section. The list of DKE authorized tenants added to the CCKM is displayed. The page displays the following details:

Field	Description
Name	Name of the DKE authorized tenant.
Tenant ID	ID of the authorized tenant.
Authorization Method	Authorization type for DKE key: Email or Role ID.

Field	Description
Creation Date	Time when the authorized tenant was created.
Last Modified	Date and time the authorized tenant was modified. Timestamp in format Day-Month-Year time in 24-hour notation.
Description	Description of the authorized tenant.

Viewing and Editing Details of a DKE Authorized Tenant

After an authorized tenant is created, you can view and modify the authorized tenant details, such as the name and authorization parameters.

This section describes how to view the details of a DKE authorized tenant and update details relating to GENERAL INFORMATION and AUTHORIZATION PARAMETERS as needed.

To view or edit a DKE authorized tenant:

1. Log in to the **CipherTrust Manager** GUI as a member of the CCKM Admins group.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **Services > Microsoft DKE**. The Microsoft Double Key Encryption (DKE) page is displayed.
4. Go to the AUTHORIZED TENANTS section. The list of DKE authorized tenants added to CCKM is displayed.
5. Click the **Name** link of the desired DKE authorized tenant.

Alternatively, click the overflow icon () corresponding to the desired DKE authorized tenant, and click **View/Edit**.

The edit view of the Microsoft Double Key Encryption (DKE) page is displayed.

Under GENERAL INFORMATION

1. (Optional) Update the **Name**.
2. (Optional) Update the **Description**.
3. Click **Update**.

Under AUTHORIZATION PARAMETERS

1. (Optional) Depending on which authorization method you previously configured:

If using Email authorization

- In **Use Email(s)**, enter additional email addresses and click **Add Email(s)**. Your entries are displayed under **Email(s)**. You can also edit and delete the desired email address by clicking the **Edit** and **Delete**

icon, respectively.

If using Role authorization

- In **Use Role ID(s)**, you can change an Azure connection and add the additional role IDs. You need to click **Edit Authorized Roles**. The Edit Authorized Roles wizard is displayed.
- You can select the roles from the list and you can add the roles manually. Click the desired tab to view the instructions.

List

Select an **Azure Connection**.

Click **List Roles** and select the desired roles from the list.

Click **Edit Authorized Roles**.

Manually

Select an **Azure Connection**.

Scroll down the page, enter the **Role ID(s)** that are authorized to use the DKE key, and click **Add Role ID(s)**. The added role IDs are displayed in the list of Role IDs.

Click **Edit Authorized Roles**.

2. Click **Update**.

Deleting a DKE Authorized Tenant

To delete a DKE authorized tenant from CCKM:

1. Log in to the **CipherTrust Manager** GUI as a member of the CCKM Admins group.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **Services > Microsoft DKE**. The Microsoft Double Key Encryption (DKE) page is displayed.
4. Go to the **AUTHORIZED TENANTS** section.
5. Click the overflow icon () corresponding to the desired DKE authorized tenant, and click **Delete**. The **Delete Authorized Tenant** message is displayed.
6. Select **I wish to delete this authorized tenant**.
7. Click **Delete**.

After the authorized tenant is successfully deleted, a message displays indicating "Successfully deleted Authorized Tenant."

Microsoft DKE service performance summary

We have tested different environments to capture performance metrics for the CipherTrust Cloud Key Manager (CCKM) Microsoft Double Key Encryption (DKE) integration. The performance results provided within this document demonstrate the effects of deployment choices on throughput for Microsoft DKE key encrypt operations for one key. This can help you plan your Ciphertrust Manager deployment to meet your performance needs for Microsoft DKE integration.

All environments used an open source [k6.io](#) tool as the REST client to simulate DKE decrypt requests from Microsoft clients.

Note

- Actual performance numbers in your environment may be different. The results can vary based on factors, such as how and where the CipherTrust Manager is deployed, CipherTrust Manager resources, the location of cloud KMS, the key source of your choice, the network connectivity between the CipherTrust Manager, Microsoft DKE Cloud and your key source, as well as how the traffic is load balanced.
- This document, may at times, abbreviate Ciphertrust Manager as CM.

Tested environments

The following deployments were tested:

- Ciphertrust Manager and K6 client are on Azure
 - Endpoint is configured with email for authentication and tenancy is a single-tenant
 - Endpoint is configured with role for authentication and tenancy is a single-tenant
- Two node cluster of Ciphertrust Manager behind load balancer is on Azure and K6 client is also on Azure
 - Endpoint is configured with email for authentication and tenancy is a single-tenant
 - Endpoint is configured with role for authentication and tenancy is a single-tenant
- Ciphertrust Manager is on AWS and K6 client is on Azure
 - Endpoint is configured with email for authentication and tenancy is a single-tenant
 - Endpoint is configured with role for authentication and tenancy is a single-tenant

The following CipherTrust Manager release version was employed in the tested deployments:

Description	Value
Release version	2.19.0.13525

Test process

The test consisted of starting a given number of Virtual users to perform decrypt operations on the Microsoft DKE key. Each user simulated a separate thread. The total test duration was 40 seconds for each reading.

Microsoft DKE deployment results

- Ciphertrust Manager and K6 client are on Azure
- Two node cluster of Ciphertrust Manager behind load balancer is on Azure and K6 client is also on Azure
- Ciphertrust Manager is on AWS and K6 client is on Azure

Ciphertrust Manager and K6 client are on Azure

Endpoint is configured with email for authentication and tenancy is a single-tenant

4 CPU 16 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	4	16 GB	Email	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
500	40	719.1537/s	0.386
1000	40	691.8735/s	0.761
1500	40	698.5089/s	1.91
2000	40	696.6359/s	2.42
2500	40	686.3065/s	3.91
3000	40	680.1765/s	5.85
3500	40	674.6702/s	7.34
4000	40	677.9767/s	10.29

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
4500	40	670.3486/s	11.6
5000	40	658.0513/s	14.31
5500	40	669.4354/s	16.98
6000	40	656.8782/s	17.59
6500	40	656.8434/s	18.1
7000	40	648.2802/s	20.77
7500	40	645.4054/s	21.43
8000	40	631.6249/s	23.78
8500	40	616.0445/s	24.99
9000	40	612.0206/s	25.91
9500	40	606.1762/s	27.98
10000	40	587.7644/s	28.40
10500	40	615.8383/s	29.56
11000	40	536.2972/s	30.81
11500	40	605.6149/s	31.91

8 CPU 64 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	8	64 GB	Email	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
500	40	2235.25	0.48
1000	40	2206.81	0.997
1500	40	2164.34	1.58

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
2000	40	2132.32	2.04
2500	40	2088.57	2.65
3000	40	1953.85	3.34
3500	40	2020.11	3.47
4000	40	1999.76	3.96
4500	40	1965.17	4.43
5000	40	1923.28	4.98
5500	40	1897.71	5.42
6000	40	1842.33	5.96
6500	40	1870.59	6.19
7000	40	1856.14	6.47
7500	40	1823.36	7.62
8000	40	1808.89	8.51
8500	40	1796.92	8.79
9000	40	1790.66	8.94
9500	40	1775.63	9.11
10000	40	1769.44	9.35
10500	40	1753.41	9.78
11000	40	1734.98	10.23
11500	40	1654.83	10.67
12000	40	1601.37	10.89
12500	40	1575.79	11.56
13000	40	1537.71	12.78

Endpoint is configured with role for authentication and tenancy is a single-tenant

4 CPU 16 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	4	16 GB	Role	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
100	40	61.68/s	1.68
200	40	67.69738/s	3.23
300	40	69.34495/s	5.03
400	40	69.12255/s	7.34
500	40	68.75192/s	9.39
600	40	71.28082/s	11.68
700	40	69.92883/s	14.42
800	40	68.53326/s	17.05
900	40	58.15/s	17.95
1000	40	66.53222/s	21.69
1100	40	66.33385/s	24.89
1200	40	68.16234/s	25.92
1300	40	68.0612/s	27.87
1400	40	68.23079/s	30.83

8 CPU 64 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	8	64 GB	Role	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
100	40	112.3529	0.9033
200	40	130.4176	1.166
300	40	186.2838	1.63
400	40	198.626	2.2
500	40	199.8047	2.84
600	40	201.9552	3.37
700	40	205.7499	4.05
800	40	199.5846	4.69
900	40	203.8096	5.31
1000	40	203.2257	6.02
1100	40	186.581	7.27
1200	40	205.6667	7.37
1300	40	203.1419	8.27
1400	40	203.0554	8.88
1500	40	156.8448	9.6
1600	40	205.3333	10.71
1700	40	186.4951	11.23
1800	40	205.6395	11.99
1900	40	199.5259	12.83
2000	40	203.2975	13.22
2100	40	205.2013	14.88
2200	40	159.341	14.64
2300	40	200.0432	16.09
2400	40	205.7545	17.06
2500	40	200.5789	17.52
2600	40	202.3035	17.82
2700	40	204.228	19.4
2800	40	184.6921	20.53
2900	40	208.018	21.17
3000	40	197.1844	21.61
3100	40	198.0743	22.28
3200	40	144.5743	21.56
3300	40	197.3122	24.84

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
3400	40	185.4164	25
3500	40	193.5786	25.43
3600	40	204.8612	25.1
3700	40	200.8094	27.61
3800	40	192.4248	31.22
3900	40	201.0072	28.19
4000	40	192.0279	27.72
4100	40	192.9561	31.11
4200	40	193.7323	31.05
4300	40	195.0307	31.93

Two node cluster of Ciphertrust Manager behind load balancer is on Azure and K6 client is also on Azure

Endpoint is configured with email for authentication and tenancy is a single-tenant

4 CPU 16 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	4	16 GB	Email	Single-tenant
Ciphertrust Manager (Azure)	50 GB	4	16 GB	Email	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
1000	40	1426.218	0.857
2000	40	1415.462	1.65

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
3000	40	1392.507	2.74
4000	40	1397.226	5.45
5000	40	1376.842	7.46
6000	40	1368.934	9.93
7000	40	1349.188	10.49
8000	40	1333.471	12.64
9000	40	1344.829	13.84
10000	40	1335.012	15.35
11000	40	1326.402	15.6
12000	40	1292.741	17.73
13000	40	1284.117	19.29
14000	40	1301.552	20.51
15000	40	1262.398	22.83
16000	40	1249.673	22.45
17000	40	1243.957	25.32
18000	40	1279.684	25.97
19000	40	1231.426	27.25
20000	40	1253.279	28.43
21000	40	1272.611	29.88
22000	40	1248.092	31.63

8 CPU 64 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	8	64 GB	Email	Single-tenant
Ciphertrust Manager (Azure)	50 GB	8	64 GB	Email	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
500	40	4491.81	0.21935
1000	40	4480.58	0.47045
1500	40	4459.62	0.7329
2000	40	4427.75	0.97955
2500	40	4402.71	1.27
3000	40	4354.23	1.55
3500	40	4331.39	1.8
4000	40	4273.88	2.02
4500	40	4295.65	2.33
5000	40	4025.86	2.7
5500	40	4198.57	2.74
6000	40	4161.5	3.01
6500	40	4127.06	3.34
7000	40	4101.25	3.49
7500	40	4020.7	3.82
8000	40	4002.08	4.09
8500	40	3966.25	4.3
9000	40	3926.6	4.49
9500	40	3931.34	4.8
10000	40	3878.66	4.95
10500	40	3845.29	5.3
11000	40	3815.92	5.36
11500	40	3820.34	5.64

Endpoint is configured with role for authentication and tenancy is a single-tenant

4 CPU 16 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	4	16 GB	Role	Single-tenant
Ciphertrust Manager (Azure)	50 GB	4	16 GB	Role	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
200	40	124.9904/s	1.65
400	40	137.5975/s	3.27
600	40	135.3681/s	5.27
800	40	141.2373/s	7.22
1000	40	141.5366/s	9.15
1200	40	137.0066/s	11.94
1400	40	140.5206/s	14.28
1600	40	139.3487/s	15.99
1800	40	128.8245/s	19.48
2000	40	133.5563/s	20.52
2200	40	132.7215/s	23.19
2400	40	133.7492/s	24.71
2600	40	134.3285/s	27.66
2800	40	132.9436/s	32.29
3000	40	134.7265/s	33.18

8 CPU 64 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	8	64 GB	Role	Single-tenant

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	8	64 GB	Role	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
100	40	224.5829	0.413
200	40	262.4118	0.927
300	40	370.9625	1.26
400	40	398.1467	1.98
500	40	402.8871	2.28
600	40	404.0253	2.37
700	40	408.7726	3.74
800	40	399.0139	4.39
900	40	406.5677	5.63
1000	40	407.4815	6.11
1100	40	374.6689	7.41
1200	40	412.6834	7.91
1300	40	410.2992	8.05
1400	40	405.2126	8.73
1500	40	314.8471	9.42
1600	40	411.0924	10.72
1700	40	372.5126	11.59
1800	40	409.8423	12.16
1900	40	396.5648	12.87
2000	40	407.0129	13.54
2100	40	410.7446	14.37
2200	40	318.4212	14.96
2300	40	400.9657	15.33
2400	40	409.6612	16.09

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
2500	40	399.2125	17.43
2600	40	405.6537	18.21
2700	40	407.8768	19.65
2800	40	369.7722	20.75
2900	40	412.4881	21.54
3000	40	393.3157	21.61
3100	40	397.8474	22.17
3200	40	290.7689	21.56
3300	40	392.5216	22.94
3400	40	375.2994	23.85
3500	40	383.8099	24.12
3600	40	410.2336	24.64
3700	40	402.9118	25.53
3800	40	381.3346	28.91

Ciphertrust Manager is on AWS and K6 client is on Azure

Endpoint is configured with email for authentication and tenancy is a single-tenant

4 CPU 16 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	4	16 GB	Email	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
500	40	442.961541/s	1.9s

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
1000	40	436.601323/s	3.97s
1500	40	434.747918/s	5.92s
2000	40	434.218522/s	7.93s
2500	40	426.235678/s	9.91s
3000	40	429.075825/s	11.86s
3500	40	429.330793/s	13.64s
4000	40	421.797852/s	15.83s
4500	40	422.863646/s	17.52s
5000	40	426.06499/s	19.27s
5500	40	419.28467/s	21.2s
6000	40	396.629457/s	24.67s
6500	40	413.221187/s	25.12s
7000	40	409.802376/s	26.72s
7500	40	408.860147/s	28.64s
8000	40	408.169754/s	30.01s

8 CPU 64 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	8	64 GB	Email	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
500	40	1032.665456/s	0.77634
1000	40	1016.625/s	1.61
1500	40	1008.57/s	2.55

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
2000	40	1005.002/s	3.3
2500	40	987.4021/s	4.36
3000	40	967.4466/s	5.24
3500	40	973.8425/s	6.17
4000	40	1029.59/s	6.61
4500	40	1029.469/s	7.35
5000	40	991.4736/s	8.34
5500	40	1004.501/s	9.1
6000	40	922.9237/s	10.86
6500	40	940.0237/s	11.66
7000	40	931.4656/s	12.06
7500	40	932.715/s	13.13
8000	40	999.6635/s	13.2
8500	40	901.0721/s	15.29
9000	40	896.9337/s	16.08
9500	40	928.6059/s	16.74
10000	40	907.8969/s	17.62
10500	40	998.3297/s	16.52
11000	40	922.3116/s	17.88
11500	40	926.6163/s	19.51
12000	40	869.0461/s	20.15
12500	40	629.4951/s	21.26
13000	40	942.8241/s	20.14
13500	40	895.4106/s	24.17
14000	40	929.9176/s	21.49

Endpoint is configured with role for authentication and tenancy is a single-tenant

4 CPU 16 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	4	16 GB	Role	Single-tenant

K6 details

K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
200	40	70.588858/s	3s
400	40	73.897893/s	6.41s
600	40	73.435278/s	11.37s
800	40	71.563103/s	14.87s
1000	40	59.223803/s	18.47s
1200	40	68.782213/s	23.68s
1400	40	63.355995/s	27.76s
1600	40	69.152327/s	27.5s
1800	40	66.602176/s	31.89s
2000	40	64.757942/s	35.27s
2200	40	55.541948/s	37.44s

8 CPU 64 GB

Here are the details of this deployment:

CM details

CM environments	Volume	CPU	Memory	Authentication	Tenancy
Ciphertrust Manager (Azure)	50 GB	8	64 GB	Role	Single-tenant

K6 details

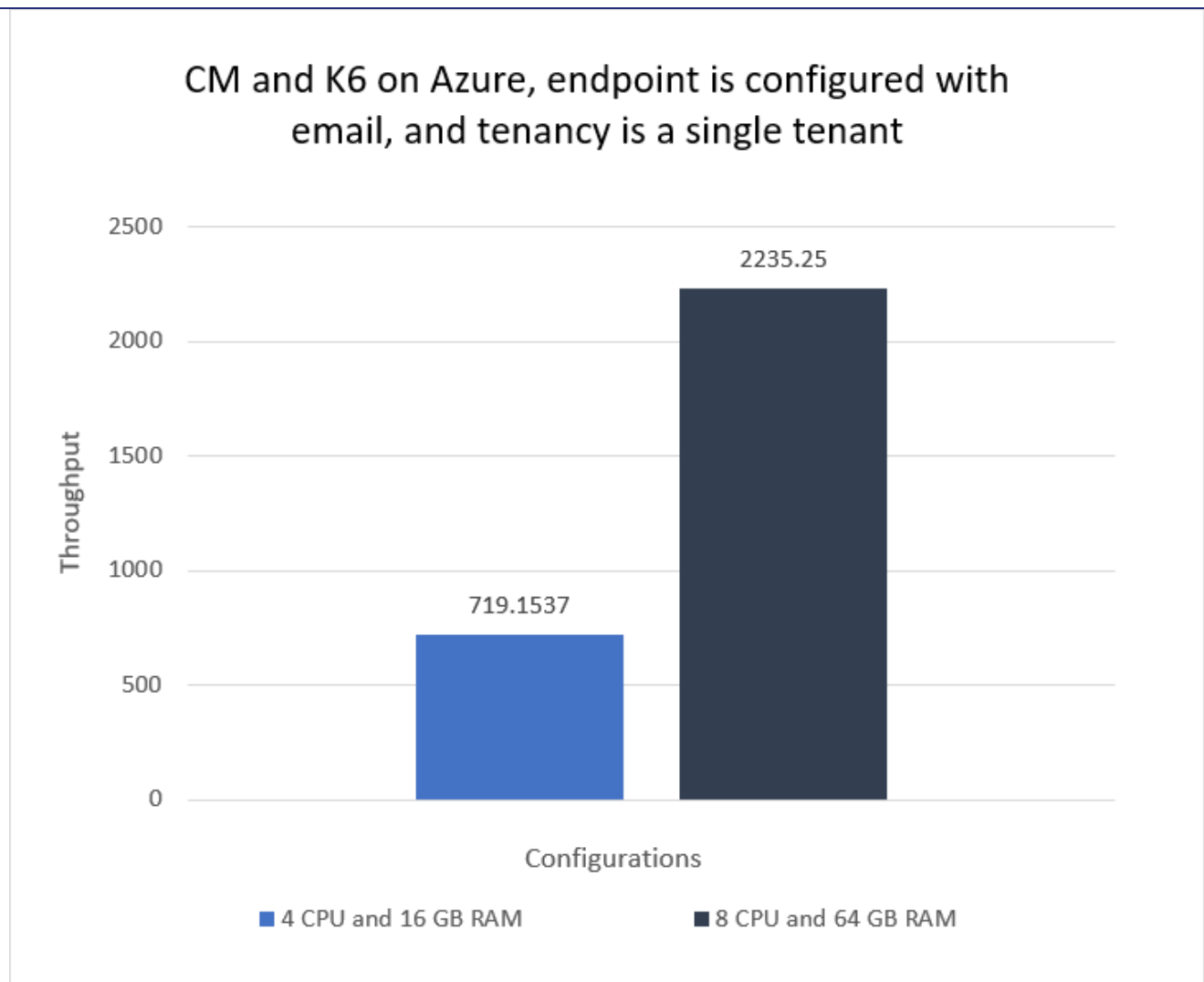
K6	Volume	CPU	Memory
K6 client (Azure)	50 GB	8	64 GB

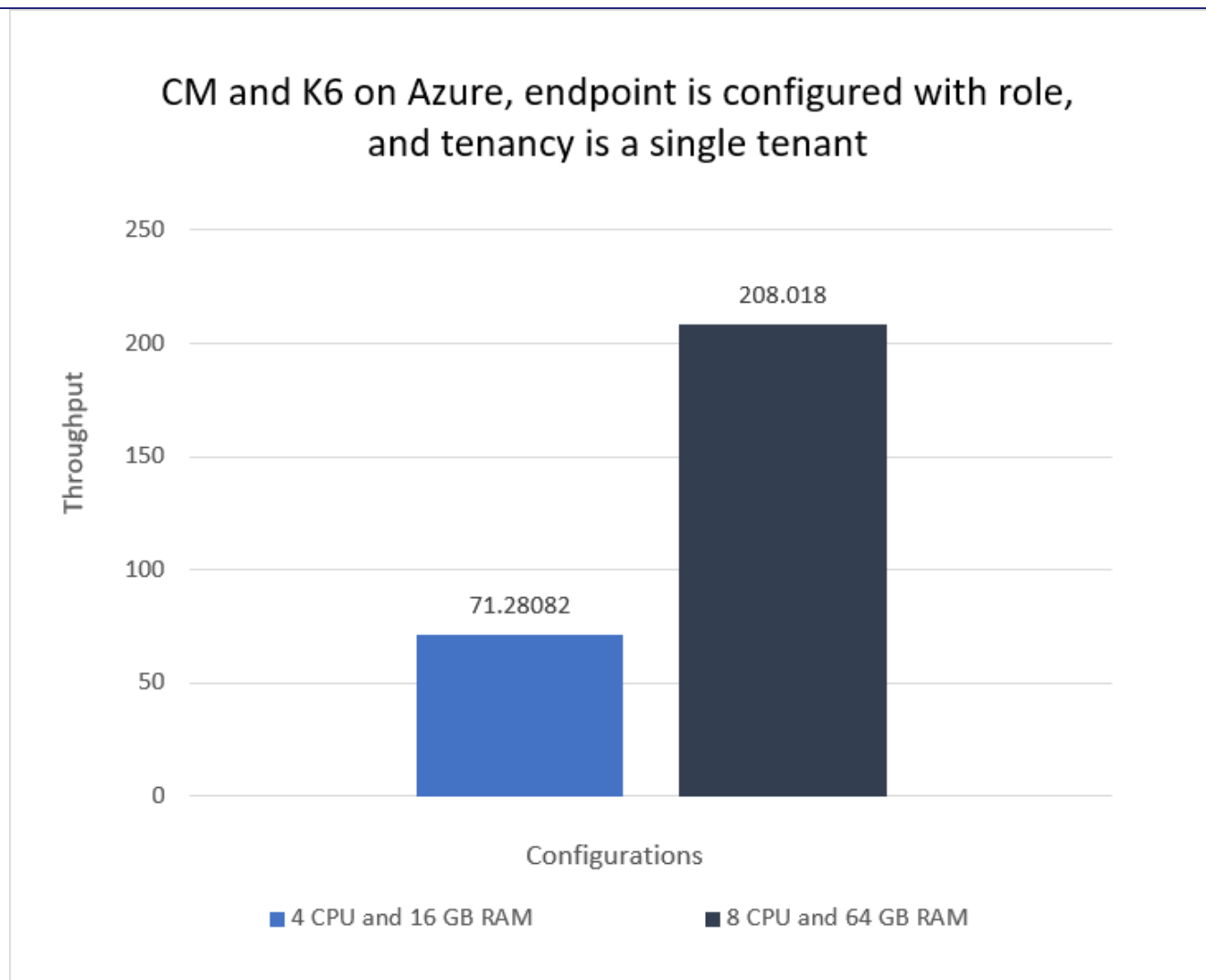
Results

Virtual users	Run time (seconds)	Operations/seconds	Time taken (P90) (seconds)
500	40	121.0911/s	4.3
1000	40	152.8317/s	8.54
1500	40	154.01/s	13.5
2000	40	123.0743/s	19.4
2500	40	149.6086/s	23.22
3000	40	147.4008/s	28.78
3500	40	144.5733/s	34.89
4000	40	135.8877/s	40.77
4500	40	138.8102/s	43.15
5000	40	130.9184/s	50.3
5500	40	140.8138/s	50.65
6000	40	148.9094/s	50.56

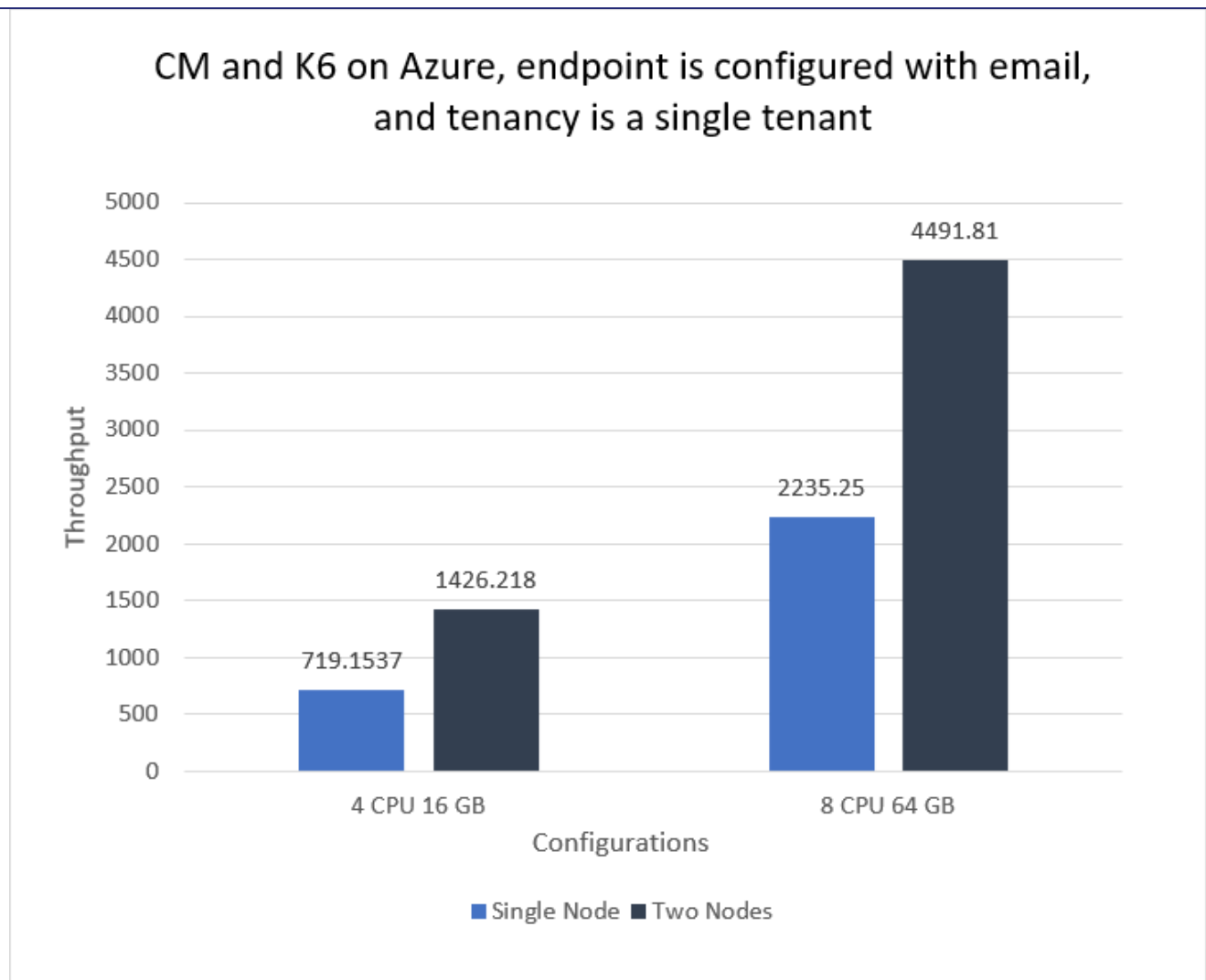
Conclusion

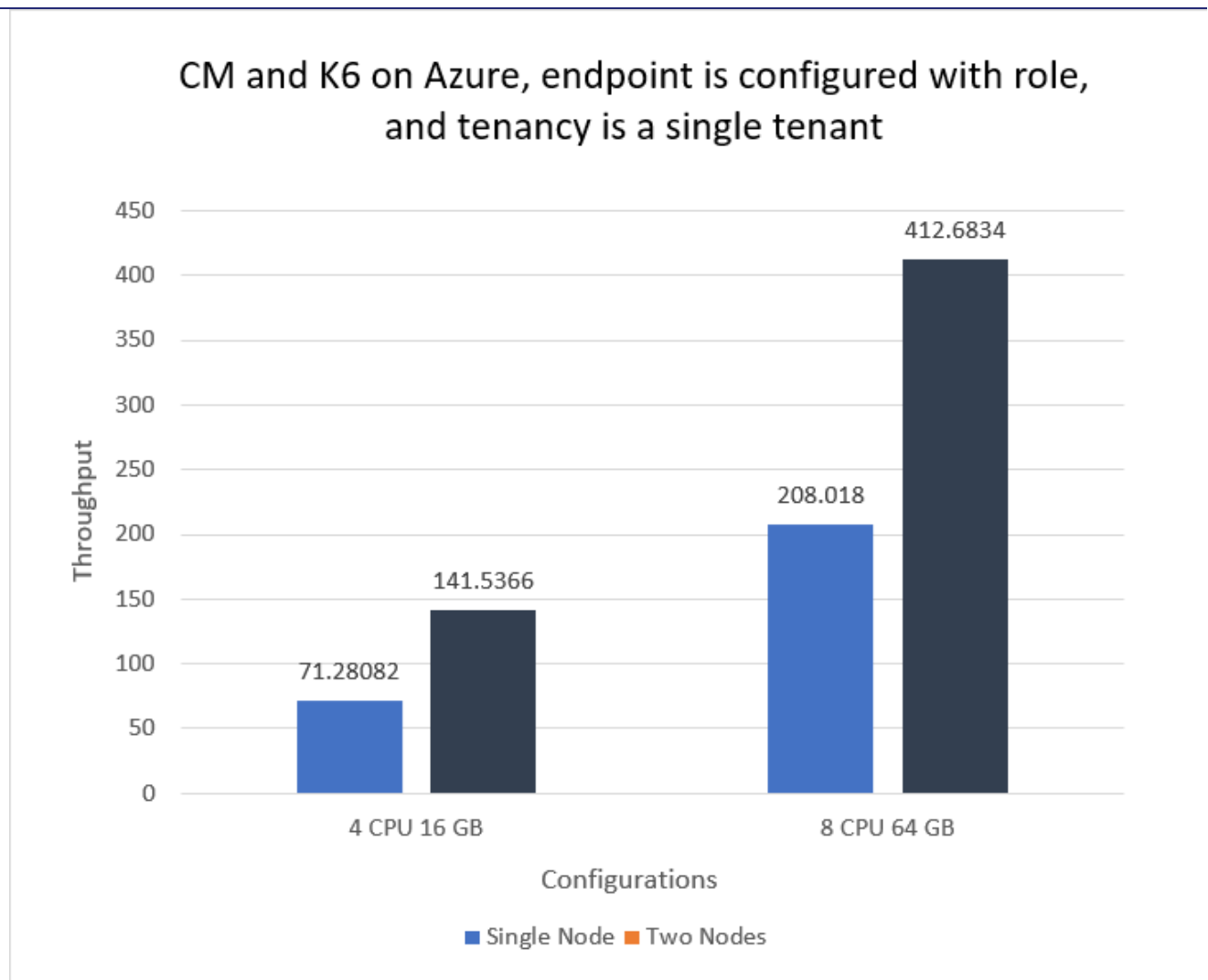
- Increasing CM RAM and CPU increases the performance.



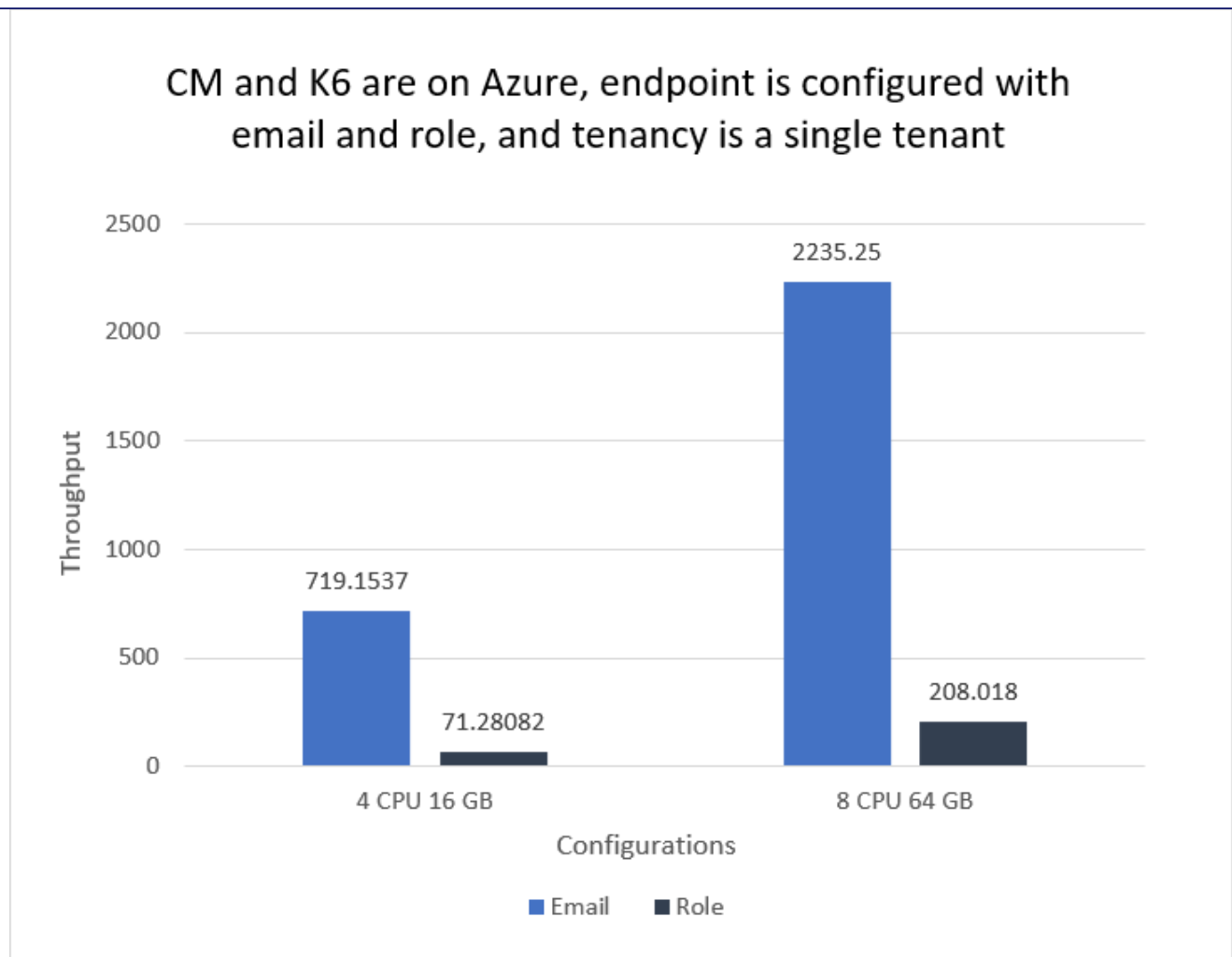


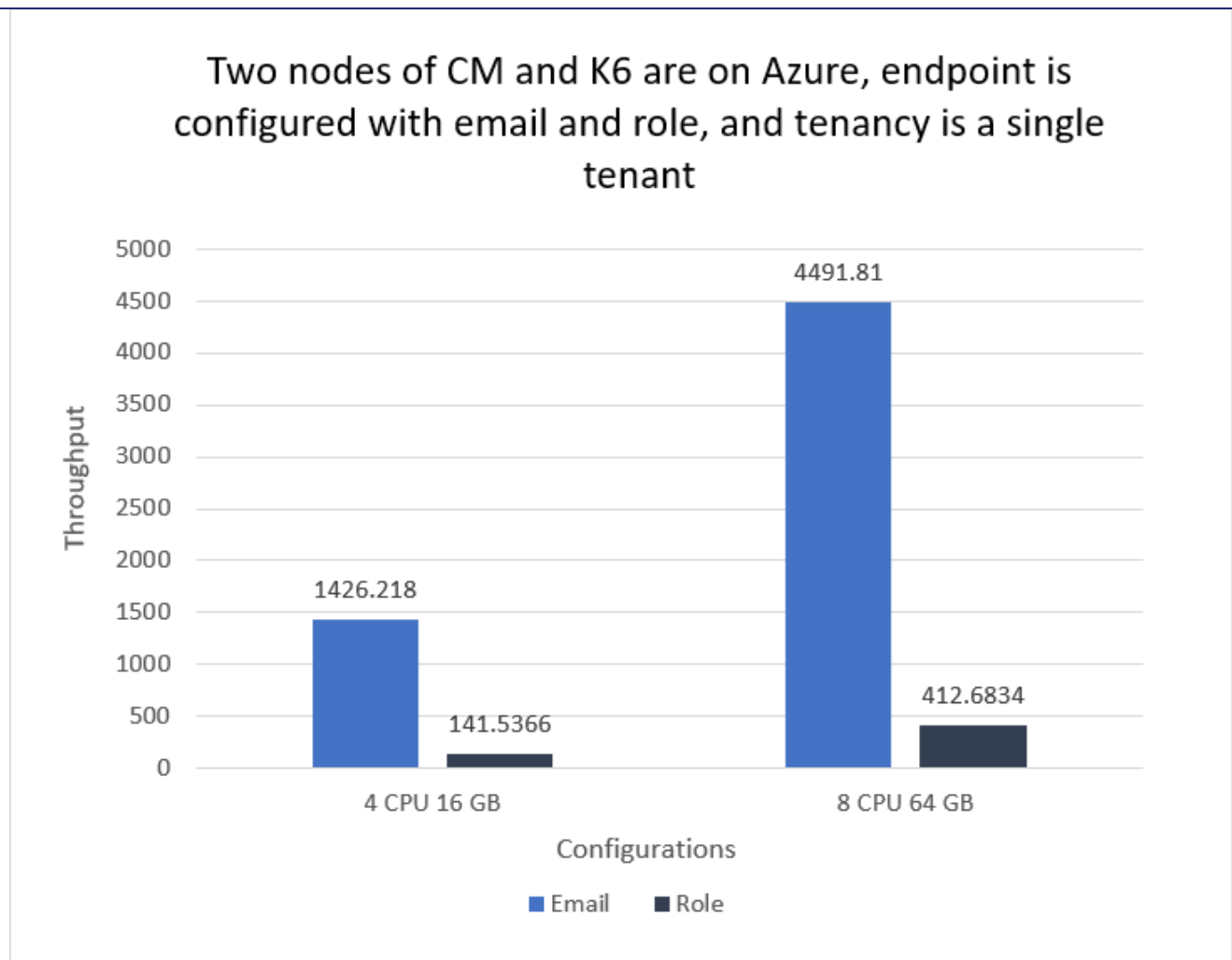
- Increasing the number of nodes enhances the performance.



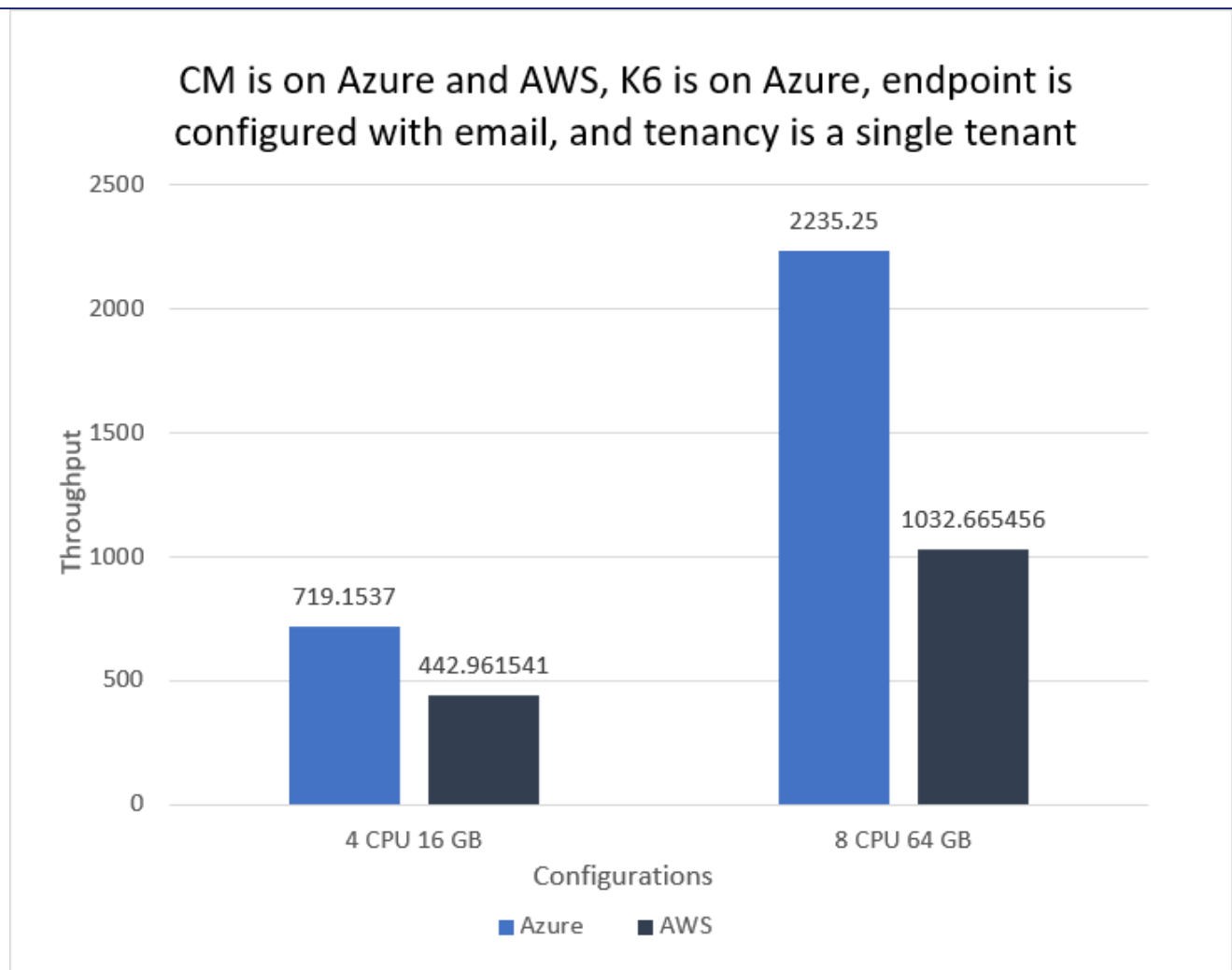


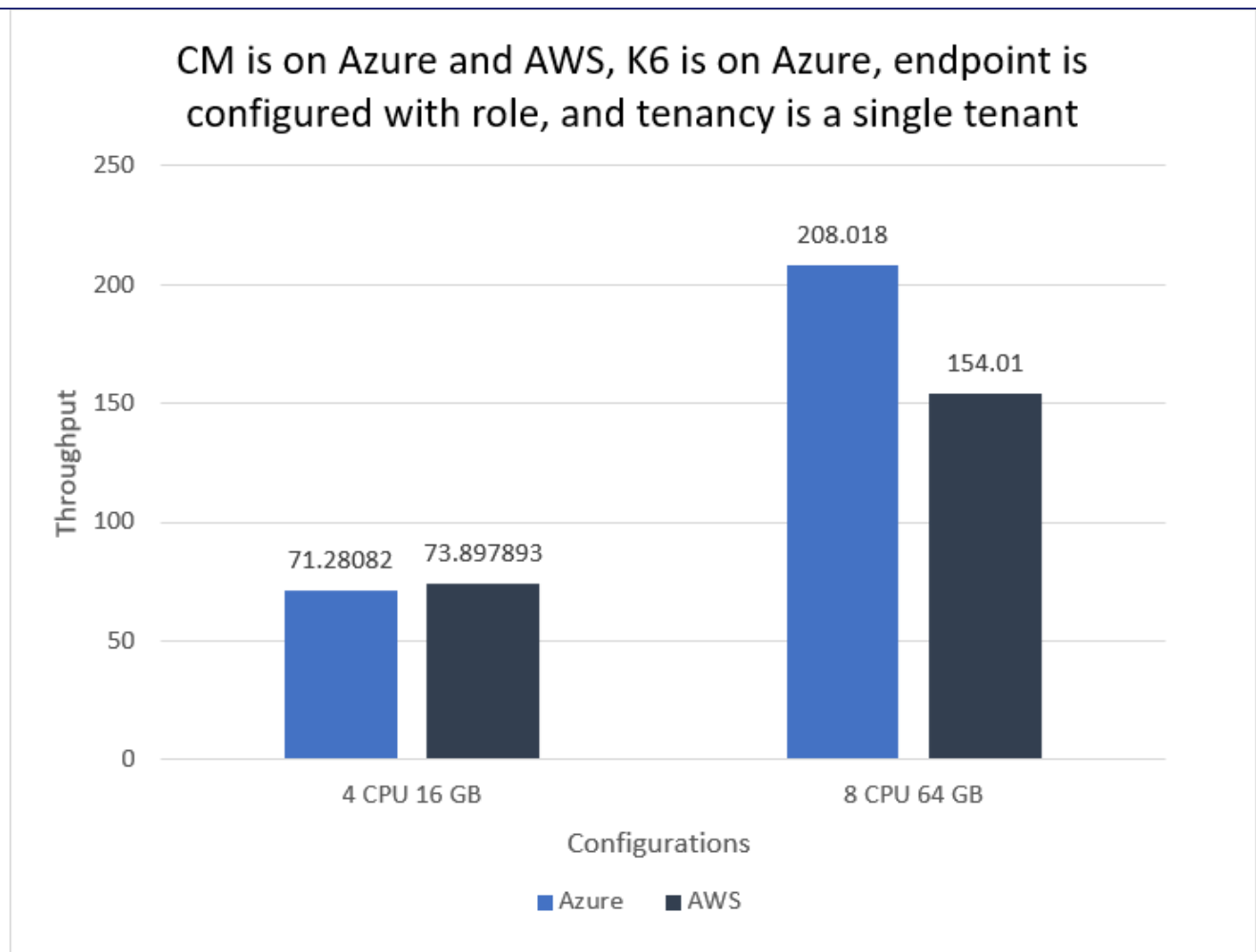
- The performance is better when the endpoint is configured with the email as compared to the role.





- The performance is better when the server and client are geographically close compared to when they are far apart.





Sizing the environment

To size the environment, you must evaluate the peak traffic of DKE by establishing the maximum number of requests hitting the DKE infrastructure. The following factors must be considered.

- The number of total users.
- The geographic distribution and flexibility in working hours.
 - Suppose a 100k users company named ABC is located in a one-time zone only. All of its employees have a shift from 9 am to 5 pm. So, there might be 95k users opening one particular DKE-protected email or document within 5 minutes at the start of their shift, which is 320 requests per second. The peaks likely exceed 1000 requests per second.
 - But suppose a 100k users company named XYZ is evenly distributed among 10 different time zones across the world. All of its employees have flexible working hours. A document opened at the start of their shift may be opened by 10k users in a 20-minute window, which is 10 requests per second. The peaks are likely below 50 requests per second.
- The usage of DKE in the protection of emails and documents. On an hourly basis, a user may open 20 DKE-

protected emails and up to 2 DKE-protected documents.

- Accessing the DKE templates and client configuration offline.
 - If a DKE-protected email thread or document is available offline for 7 days, opening it will result in one request per user.
 - If the offline caching is disabled, editing the same email thread or document may result in several requests per hour. At least one request each time the same user opens the same email or document.
- The number of total DKE-protected emails or documents opened by the average user on a daily basis.
 - If only the most sensitive content is DKE-protected, an average ABC user may access 2 DKE-protected emails or documents per day. It will generate 200k requests per day, which is less than 10 requests per second.
 - If all emails and documents are DKE-protected, the 100k users of XYZ can access 50 DKE-protected emails or documents per day. It may generate nearly 200 requests per second even with offline access, which is likely 500 requests without offline access.

To ensure high availability, DKE infrastructure must be fault-tolerant, behind a network load balancer.

If you already have the Managed Service Providers, a good way to estimate the load is by checking the number of documents accessed with the highest classifications like Strictly Confidential.

External CipherTrust Resources

This section describes prerequisites to manage the external CipherTrust resources on CCKM.

Prerequisites

Before you can use an external CipherTrust Manager as a key source in CCKM:

1. Make sure that the external CipherTrust Manager is installed and configured. Refer to [CipherTrust Manager Deployment](#) for details.
2. After the external CipherTrust Manager is configured, configure the connection to the external CipherTrust Manager. Refer to [Connection Manager](#) for details.

After ensuring the prerequisites, the CipherTrust Manager keys can be managed on the CipherTrust Manager. Refer to the following sections:

- [Managing External CipherTrust Domains](#)
- [Managing External CipherTrust Keys](#)

Managing External CipherTrust Domains

This section describes how to manage external CipherTrust domains on CCKM.

Before proceeding, make sure to fulfill [prerequisites](#).

Adding CipherTrust Domains

To add a CipherTrust domain to CCKM:

1. Log on to the **CipherTrust Manager** GUI as administrator.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **KMS Containers > CipherTrust Domains**. The CipherTrust Domains page is displayed.
4. Click **Add Domain**. The Add Existing Domain page is displayed.
5. From the **Connection** drop-down list, select the desired CM connection to the CipherTrust Manager.
6. Click **Save**.

The CipherTrust domain is added to CCKM.

A message **CipherTrust Domain added successfully** is displayed on the screen.

Refreshing CipherTrust Domains

Refreshing is the process of downloading keys created on the CipherTrust domains to CCKM.

Refreshing All Domains

To refresh all CipherTrust domains:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Ciphertrust Domains**. CipherTrust Domains page is displayed. This page displays the list of CipherTrust Domains.
3. Click **Refresh All**. The "This may take a while..." message is displayed.

Note

Refreshing CipherTrust domains is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed domains are listed on the **Cloud Keys > CipherTrust > CipherTrust (External) Keys** page. Refer to [Viewing CipherTrust \(External\) Keys](#) for details.

Viewing/Editing Details of CipherTrust Domains

The CipherTrust Domains page shows the list of existing CipherTrust domains. Search for domains by Domain Name or Connection.

Viewing CipherTrust Domains Details

To view the details of CipherTrust domains:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > CipherTrust Domains**. The CipherTrust Domains page displays the following details.

Column	Description
Domain Name	Name of the CipherTrust domain. Click the link to view more details about the domain.
Connection	Name of the CipherTrust connection.
Creation	When the domain was created.
Updated	When the domain was updated.

|Last Refreshed|When the domain was last refreshed. `Never` is displayed for domains that are never refreshed.|

Click the Customize View () icon, select the desired option, and click **OK** to display the column.

Modifying CipherTrust Domain Details

To modify the details of a CipherTrust domain:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > CipherTrust Domains**. The CipherTrust Domains page displays the list of added CipherTrust domains.
3. Click the overflow icon () corresponding to the desired CipherTrust domain and click **View/Edit Details**.

You can change the CM connection and its description, and modify user/group permissions on the CipherTrust domain. For details, refer to:

- [Changing the CM Connection](#)
- [Managing User Permissions on CipherTrust Domains](#)

Changing the CM Connection

To change the CM connection:

1. Expand **GENERAL INFO**.
2. From the **Connection** drop-down list, select the desired CM connection.
3. Click **Update**.

A message **Updated connection for this domain** is displayed on the screen.

Managing User Permissions on CipherTrust Domains

To work with CipherTrust, users/groups must have the minimum set of permissions that allow them to use the CipherTrust resources such as CipherTrust keys and domains. Initially, the user only has permission to view the keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform operations on the CipherTrust domain.

To add permission for a user/group:

1. Expand **ACCESS CONTROL**.
2. In the **ACCESS CONTROL** section, click **Assign User/Group**. The **Assign User/Group** screen is displayed.
3. From the **User/Group** drop-down list, select the user or group to be assigned permissions.
4. Click **Save**.

The newly added user/group is displayed under **Name** in the **ACCESS CONTROL** section.

CCKM allows the following operations on the CipherTrust domains:

- View Keys, Add Key, Delete Key
- Refresh Domain

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. Select the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A message **Updated access control for this Project** is displayed on the screen.

Removing a Permission

To remove a permission assigned to a user or group:

1. Clear the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A message **Updated access control for this Project** is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. Under **Unassign**, click the **X** button corresponding to the desired user/group. The **Remove User / Remove Group** screen is displayed.

Note

Removing this user/group will remove all permissions currently assigned to the user/group. Are you sure you want to continue?

2. Click **Remove**.

A message **Updated access control for this Project** is displayed on the screen.

Deleting CipherTrust Domains

To delete a CipherTrust domain:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > CipherTrust Domains**. The CipherTrust Domains page is displayed. The CipherTrust Domains page displays the list of added CipherTrust Domains.
3. Click the overflow icon () corresponding to the desired CipherTrust Domain and click **Delete**. The **Delete CipherTrust Domain** screen is displayed.
4. Click **Domain**.

A message **CipherTrust Domain deleted successfully** is displayed on the screen.

Managing External CipherTrust Keys

This section describes how to manage CipherTrust keys or CipherTrust (External) keys on CCKM. Before

proceeding, you must have an CipherTrust domain added to the CCKM. Refer to [Managing CipherTrust Domains](#) for details.

Adding CipherTrust (External) Keys

To add a CipherTrust (External) key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > CipherTrust**. The list of available CipherTrust (External) keys is displayed.
3. Click **Add Key**. The Add CipherTrust (External) Key screen of the Add CipherTrust (External) Key wizard is displayed.

Add CipherTrust (External) Key

1. **Select Domain** for the key. The key will be added to this domain.
2. (Optional) Enter the **Key Name**. This helps in uniquely identifying a key.
3. Select the key **Algorithm**. The algorithm can be AES (default), TDES, RSA, Elliptic Curve, HMAC-SHA1, HMAC-SHA256, HMAC-SHA384, and HMAC-SHA512.
4. Select the **Key Size** or **Curve**.

Note

If the algorithm is not Elliptic Curve, select the **Key Size**. If the algorithm is Elliptic Curve, select the **Curve**.

5. (Optional) Select the **Key Attributes**. The supported attributes are **Exportable** and **Deletable**.
6. (Optional) Select the **Key Usage(s)**.
7. (Optional) Enter **KMIP Mask Shorthand**.
8. (Optional) Enter **Labels**. A label is a label assigned to the key that consists of a user-defined key and a value.

To add a label:

- a. Specify a label name.
- b. Specify the label value.
 - CCKM allows the following characters in label values:
 - Alphanumeric characters

- Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; . ' " _ **`

- CCKM does not allow the colon (:) in label values.

c. Click the **+** button.

Similarly, you can add more labels. To remove a label, click the close (X) icon in the label name.

9. Click **Next**. The Review and Add screen is displayed. This screen shows the key details.

Tip

Before adding the CipherTrust (External) key, it is recommended to review its details. After the key is added, its certain features cannot be edited.

Review And Add

1. Review the key details. If the key requires any changes, click the **Back** link and modify the details.
2. Click **Save**.

After the key is created successfully, a success message **Key added successfully.** is displayed.

3. Click **Close**.

The newly created key is displayed in the keys list.

Viewing CipherTrust (External) Keys

The CipherTrust (External) Keys page displays the available CipherTrust (External) keys. Search for CipherTrust (External) keys by Key Name or UUID (Universally Unique Identifier) of the key.

To view a CipherTrust (External) key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > CipherTrust**. The CipherTrust (External) Keys page displays following details:

Field	Description
Key Name	Name of the CipherTrust (External) key. Click to view details of keys. Refer to Viewing CipherTrust (External) Keys for details.
Status	State of the Google Cloud key. The status can be: • Active

Field	Description
	• Pre-Active • Deactivated • Destroyed • Compromised • Destroyed Compromised • Deleted
Version	Version of the CipherTrust (External) key.
Owner	Owner of the CipherTrust (External) key.
Creation Date	Time when the key is added to CCKM.
Type	Type of the CipherTrust (External) key - Symmetric, Public, or Private.
Algorithm	Algorithm of the CipherTrust (External) key. The algorithm can be: • AES • TDES • RSA • Elliptic Curve • HMAC-SHA1 • HMAC-SHA256 • HMAC-SHA384 • HMAC-SHA512.
Size	Size of the CipherTrust (External) key.
Domain	ID of the CipherTrust domain where the key is created. Click on the ID to view the details of the domain.

To hide/display columns, click the Customize View (⚙️) icon, select or clear the desired check boxes, and click **OK**.

Viewing or Editing Details of CipherTrust Keys

To view or edit a CipherTrust key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > CipherTrust**. The list of available CipherTrust (External) keys is displayed.
3. Click the overflow icon (⋮) corresponding to the desired key and click **View/Edit Details**. Alternatively, you can click the key name link. The edit view of the key is displayed. The edit view is divided into:

-
- **GENERAL INFO:** View the key ID and add tags. Refer to [Adding/Removing Tags](#) for details.
 - **GENERAL INFO:** Add or update the Key Attributes, Key Usages, KMIP Mask Shorthand, Labels, and Key Metadata. Refer to [Updating General Info](#) for details.
 - **DATES:** Add and updated the deactivation date, protect start date, and protect stop date of the key. Refer to [Adding and Updating Dates](#)
 - **KEY SCHEDULE:** Add, update, and disable a key rotation schedule. Refer to [Adding or Changing Key Rotation Schedule](#) and [Disabling Key Rotation Schedule](#).
 - **KEY VERSIONS:** View details of key versions. Refer to [Viewing Key Version Details](#).

Updating General Info

1. Expand the GENERAL INFO section, if needed.
2. Add or update the following details.

- a. **Key Attributes**

- b. **Key Usages**

- c. **KMIP Mask Shorthand**

- d. **Labels.**

To add a label:

- a. Specify a label name.
 - b. Specify the label value.
 - CCKM allows the following characters in label values:
 - Alphanumeric characters
 - Special characters `** ! @ # $ % ) ( { } > < ? + - / \ [ ] ^ & + = | ~ ` , ; . ' " _ **`
 - CCKM does not allow the colon (:) in label values.
 - c. Click the + button.

Similarly, you can add more labels. To remove a label, click the close (X) icon in the label name.

- e. **Key Metadata**

3. Click **Update**.

Adding and Updating Dates

1. Expand the **DATES** section.
2. Add or Update **Deactivation Date**.
3. Add or Update **Process Start Date**.
4. Add or Update **Protect Stop Date**.
5. Click **Update**.

Viewing Key Version Details

To view the details of key versions, expand the **KEY VERSIONS** section. The list of key versions with their details are displayed:

Field	Description
Version	Version of the CipherTrust (External) key.
ID	
Status	State of the Google Cloud key. The status can be: • Active • Pre-Active • Deactivated • Destroyed • Compromised • Destroyed Compromised • Deleted
Exportable	
Deletable	
Never Exported	
Never Exportable	
Created	Time when the version is created.
Modified	Time when the version is modified.

Refreshing CipherTrust (External) Keys

Refreshing is the process of downloading keys from configured CipherTrust domains and updating their details on the CCKM GUI. You can refresh keys of all domains at once.

To refresh keys of all domains:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > CipherTrust**. The list of available CipherTrust (External) keys is displayed.

3. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refreshing all CipherTrust domains is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > CipherTrust > CipherTrust Keys** page.

Adding a Key Version

To add a new key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > CipherTrust**. The list of available CipherTrust (External) keys is displayed.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Add Version**. The Add Version dialog box is displayed.
5. Click **Yes, Add Version**. A success message **Version added successfully.** is displayed.

A new version is added to the key. The Version Count increases by one on the CipherTrust (External) keys page.

Deleting CipherTrust (External) Keys

To delete a CipherTrust (External) key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > CipherTrust**. The list of available CipherTrust (External) keys is displayed.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Delete**. The Delete Key dialog box is displayed.
5. Click **Delete Key**.

A success message **Delete is in progress** is displayed. It may take some time to reflect the changes. After a key is deleted, its status becomes **DeLeted** on the CipherTrust (External) Keys page.

Deleting a Key Version

To delete a key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > CipherTrust**. The list of available CipherTrust (External) keys is displayed.
3. Click the **Key Name** link of the desired key.

Alternatively, click the overflow icon () corresponding to the desired key and click **View/Edit**.

4. Expand **VERSIONS**.
5. Click the overflow icon () corresponding to the desired key version.
6. Click **Delete**. The Delete Version dialog box is displayed.
7. Click **Delete Key Version**.

A success message **Version deleted successfully**. is displayed. After a key version is deleted, its status becomes **DeletEd**.

DSM Resources

This section describes prerequisites to manage DSM resources on CCKM.

Prerequisites

Before you can use a DSM as a key source in CCKM:

- The DSM must be installed and configured and the DSM server certificate must be downloaded.
- A connection to the DSM must be configured on the CipherTrust Manager.

Refer to the sections below.

On the DSM

For any DSM you plan to use as key source, a DSM administrator must:

1. Install and configure a DSM. Refer to the latest version of the DSM Administration Guide.

Note

When adding a connection to the DSM on the CipherTrust Manager, you need to enter the DSM account information you gather during the installation and configuration of the DSM.

2. Create a DSM administrator account of type **All** or **Domain and Security Administrator** for CCKM. These administrators can create keys on the DSM.
3. Note the following information:
 - DSM fully-qualified host name and IP address.
 - Name and password of the DSM administrator created in previous step.
4. Export the DSM server certificate to your laptop or desktop using the Chrome Internet browser.

CCKM performs key operations through REST APIs exposed in the DSM. CCKM requires the DSM server certificate to invoke the APIs. When adding connection to the DSM as your key source, you need to upload the DSM certificate to the CipherTrust Manager.

Note

The steps to export a DSM server certificate using other browsers (for example, Firefox) may differ slightly.

To export your DSM server certificate from DSM using the Chrome browser:

- a. Enter `https://<DSM IP address>` in a browser. The DSM user login page is displayed.
- b. Log on to the web-based management console of the DSM.
- c. In the web URL, click the **Not Secure** section and click **View Certificate**.
- d. Click the **Details** tab of the certificate.
- e. Click **Copy to File**. The **Certificate Export Wizard** is displayed. This wizard is used to export your certificate in the appropriate format.
- f. On the **Export File Format** page, select the **Base-64 encoded X.509 (.CER)** format option.
- g. Click **Next**.
- h. On the **File to Export** page, enter the directory to which to save the file on your laptop or desktop and

specify the file name (for example, **DSM.cer**).

- i. Click **Finish** to complete the export.

You need to upload this DSM server certificate to the [DSM Connection](#) page of the CipherTrust Manager when adding a connection to the DSM as your key source.

On the CipherTrust Manager

After the DSM is configured and the DSM certificate is downloaded, a CipherTrust Manager configures the connection to the DSM. Refer to [Connection Manager](#) for details.

After ensuring the prerequisites, DSM domains and DSM keys can be managed on the CipherTrust Manager. Refer to the following sections:

- [Managing DSM Domains](#)
- [Managing DSM Keys](#)

Managing DSM Domains

This section describes how to manage DSM domains on CCKM.

Before proceeding, make sure to fulfill [prerequisites](#).

Adding DSM Domains

To add a DSM domain to CCKM:

1. Log on to the **CipherTrust Manager** GUI as administrator.
2. Open the **Cloud Key Manager** application.
3. In the left pane, click **KMS Containers > DSM Domains**. The DSM Domains page is displayed.
4. Click **Add Domain**. The Add Existing Domain page is displayed.
5. From the **Connection** drop-down list, select the desired connection to the DSM.
6. From the **Domain** drop-down list, select the desired DSM domain. The drop-down list shows existing domains of the DSM linked to the selected connection.

Tip

Select multiple DSM domains to add them at once.

7. Click **Add**.
-

The DSM domain is added to CCKM.

A message **Domain added successfully...** is displayed on the screen.

Refreshing DSM Keys

Refreshing is the process of downloading keys created on the DSM domains to CCKM. Refresh can be achieved using any of the following ways:

- [Refreshing Specific Domains](#)
- [Refreshing All Domains](#)

Refreshing Specific Domains

To refresh a specific domain:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > DSM Domains**. The DSM Domains page is displayed. This page displays the list of DSM domains.
3. Click the overflow icon () corresponding to the desired DSM domain and click **Refresh Now**.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

After successful refresh, the refreshed keys are listed on the **Cloud Keys > DSM > DSM Keys** page. Refer to [Viewing DSM Keys](#) for details.

Refreshing All Domains

To refresh all DSM domains:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > DSM Domains**. The DSM Domains page is displayed. This page displays the list of DSM domains.
3. Click **Refresh All**. The "This may take a while..." message is displayed.

Note

Refreshing all DSM domains is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > DSM > DSM Keys** page. Refer to [Viewing DSM Keys](#) for details.

Viewing/Editing Details of DSM Domains

The DSM Domains page shows the list of existing DSM domains. Search for domains by Domain Name or Connection.

Viewing DSM Domains Details

To view the details of DSM domains:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > DSM Domains**. The DSM Domains page displays the following details.

Column	Description
Domain Name	Name of the DSM domain. Click the link to view more details about the domain.
Connection	Name of the DSM connection.
Admin Type	Type of the DSM administrator linked with the domain. The type can be: • SYSTEM_ADMIN • DOMAIN_ADMIN • SECURITY_ADMIN • SECURITY_AND_DOMAIN_ADMIN • ALL_ADMIN
Last Refreshed	When the domain was last refreshed. Never is displayed for domains that are never refreshed.
Created At	When the domain was created.

Click the Customize View () icon, select the desired option, and click **OK** to display the column.

Modifying DSM Domain Details

To modify the details of a DSM domain:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > DSM Domains**. The DSM Domains page displays the list of added DSM domains.
3. Click the overflow icon () corresponding to the desired DSM domain and click **View/Edit Details**.

You can change the DSM connection and its description, and modify user/group permissions on the DSM domain. For details, refer to:

- [Changing the DSM Connection](#)
- [Managing User Permissions on DSM Domains](#)

Changing the DSM Connection

To add permission for a user/group:

1. Expand **GENERAL INFO**.
2. From the **Connection** drop-down list, select the desired DSM connection.
3. Click **Update**.

A message **Updated connection for this domain** is displayed on the screen.

Managing User Permissions on DSM Domains

To work with DSM, users/groups must have the minimum set of permissions that allow them to use the DSM resources such as DSM keys and domains. Initially, the user only has permission to view the keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform operations on the DSM domain.

To add permission for a user/group:

1. Expand **ACCESS CONTROL**.
2. In the **ACCESS CONTROL** section, click **Assign User/Group**. The **Assign User/Group** screen is displayed.
3. From the **User/Group** drop-down list, select the user or group to be assigned permissions.
4. Click **Save**.

The newly added user/group is displayed under **Name** in the **ACCESS CONTROL** section.

CCKM allows the following operations on the DSM domains:

- View Keys, Add Key, Edit Key, Delete Key
- Refresh Domain
- Remove

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. Select the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A message **Updated access control for this domain** is displayed on the screen.

Removing a Permission

To remove a permission assigned to a user or group:

1. Clear the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A message **Updated access control for this domain** is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. Under **Remove**, click the **X** button corresponding to the desired user/group. The **Remove User / Remove Group** screen is displayed.
2. Click **Remove**.

Note

Removing this user/group will remove all permissions currently assigned to the user/group. Are you sure you want to continue?

3. Click **Remove**.

A message **Updated access control for this key domain** is displayed on the screen.

Deleting DSM Domains

To delete a DSM domain:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > DSM Domains**. The DSM Domains page displays the list of added DSM domains.

- Click the overflow icon () corresponding to the desired DSM domain and click **Delete**. The **Delete DSM Domain** screen is displayed.

Warning

The deleted domain's keys will no longer be available on the DSM Keys page, but the keys will still exist on the DSM. If you later add this domain with the same ID, the keys will be available again.

- Select **I wish to delete this domain**.
- Click **Delete Domain**.

A message **Domain deleted successfully** is displayed on the screen.

Managing DSM Keys

This section describes how to manage DSM keys on CCKM. Before proceeding, you must have a DSM domain added to the CCKM. Refer to [Managing DSM Domains](#) for details.

CCKM supports two types of DSM keys:

- **Symmetric:** A randomly generated key is used to encrypt and decrypt the data.
- **Asymmetric:** A public and private RSA key pair is used to encrypt and decrypt the data. The public key encrypts while the private key decrypts the data.

Adding DSM Keys

To add a DSM key:

- Open the **Cloud Key Manager** application.
- In the left pane, click **Cloud Keys > DSM**. The list of available DSM keys is displayed.
- Click **Add Key**. The Add DSM Key dialog box is displayed.
- On the **Add DSM Key** screen:
 - Select Encryption Type.** The options are **Symmetric** and **Asymmetric**. Depending on the selected encryption type, the **Algorithm** option change. Also, the **Encryption Mode** field is displayed for symmetric keys.
 - Specify a **DSM Key Name**. This helps in uniquely identifying a key. The key name is mandatory.
 - Provide a **Description** of the key.
 - Select the desired **DSM Domain** for the key. The key will be added to this domain. The DSM domain is

mandatory.

- e. (Optional) Select **Expiration Date**. The key will expire on this date and time.

When you select **Expiration Date**, the **UTC** field appears under it. Click the field and use the on-screen calendar to specify the key expiration date and time. Alternatively, you can manually specify the expiration date and time in the format, **MM/DD/YYYY 00:00**.

Note

Make sure the expiration time exists in the future, otherwise, the key creation fails with an error.

- f. (Optional) Select **Extractable**. If selected, the key material created on the DSM can be extracted for management from CCKM.
- g. Select the key **Algorithm**. The algorithm is mandatory. Based on the encryption type, the supported algorithms are:
- **Symmetric**: AES-128, AES-256, ARIA-128, and ARIA-256.
 - **Asymmetric**: RSA-1024, RSA-2048, RSA-3072, and RSA-4096.
- h. (Optional, applicable to symmetric keys) Select the **Encryption Mode**. The options are CBC, CBC-CS1, and XTS.
- i. Click **Next**. The Review DSM Key screen is displayed. This screen shows the key details.

Tip

Before adding the DSM key, it is recommended to review its details. After the key is added, its certain features cannot be edited.

5. On the **Review DSM Key** screen:

- a. Review the key details. If the key requires any changes, click the **Edit** link and modify the details.
- b. Click **Create**.

The message **Create Key is in progress** is displayed. It may take some time to create the key. Leave the window open until the key creation process is complete. After the key is created successfully, a success message **Your key was successfully created. Close window to return to all keys.** is displayed.

- c. Click **OK**.

The newly created key is displayed in the keys list.

Note

Creation of a DSM key fails if:

- The selected DSM domain is full. If the key creation fails, ensure enough free disk space is available on the domain and retry the key creation.
- The DSM user credentials are incorrect or the password has expired. Check the credentials or reset the user password, as appropriate, test the connection, and retry the key creation.
- The DSM domain admin is different from the DSM admin used for creating the DSM connection.

Viewing DSM Keys

The DSM Keys page displays the available DSM keys. Search for DSM keys by Key Name or UUID (Universally Unique Identifier) of the key.

To view a DSM key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > DSM**. The DSM Keys page displays following details:

Field	Description
Key Name	Name of the DSM key. Click to view details of keys. Refer to Viewing DSM Keys for details.
State	State of the DSM key - ACTIVE or DELETED.
UUID	UUID of the DSM key.
Key Type	Type of the DSM key - Symmetric or Asymmetric.
Algorithm	Algorithm of the DSM key. The algorithm can be: Symmetric • AES128 • AES256 • ARIA128 • ARIA256 Asymmetric • RSA1024 • RSA2048 • RSA3072 • RSA4096

Field	Description
Creation Date	Time when the key is added to CCKM.
Expiration Date	Time when the DSM key will expire.
Domain Name	Name of the DSM domain where the key is created.
Versioned	Whether the DSM key is versioned.

To hide/display columns, click the Customize View (⚙️) icon, select or clear the desired check boxes, and click **OK**.

Viewing Details of a DSM Key

The DSM Keys page displays the list of available keys with their details. After a key is created, you can view its description, the key name, encryption type, domain, algorithm, and encryption.

The key description is displayed under the GENERAL INFO section and key versions are displayed under the KEY VERSIONS section.

To view details of a DSM key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > DSM**. The list of available DSM keys is displayed.
3. Click the **Key Name** link of the desired key. The mini detail view displays the key details.

Alternatively, click the overflow icon (⋮) corresponding to the desired key and click **View/Edit**.

The KEY VERSIONS section shows the following details:

Field	Description
Key Hash	Hash of the DSM key.
Version	Version of the DSM key.
State	State of the key - ACTIVE or DELETED.
Algorithm	Algorithm of the key.
Creation Date	Time when the key is added to CCKM.
Expiration Date	Time when the DSM key will expire.
Reason Key Was Rotated	Explains why the key is rotated.

Refreshing DSM Keys

Refreshing is the process of downloading keys from configured DSM domains and updating their details on the CCKM GUI. You can refresh keys of all domains at once.

To refresh keys of all domains:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > DSM**. The list of available DSM keys is displayed.
3. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refreshing all DSM Domains is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > DSM > DSM Keys** page.

Deleting DSM Keys

To delete a DSM key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > DSM**. The list of available DSM keys is displayed.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Delete**. The Delete Key dialog box is displayed.
5. Select **I wish to delete this key**.
6. Click **Delete**.

A success message **Delete is in progress** is displayed. It may take some time to reflect the changes. After a key is deleted, its status becomes **DeLeted** on the DSM Keys page.

Scheduling Operations

CCKM allows you to periodically refresh keys in the background. To refresh keys, you need to create schedule configuration. A schedule configuration defines when to run a refresh job. You can use either basic format or cron format to specify the time when the refresh job will run.

Specify basic format in the following order:

[Repeats, at]

Where,

- **Repeats:** frequency of the scheduler. Possible values are daily, weekly, monthly, and yearly.
- **at:** specific time at which the refresh operation will be performed. Possible value lies between 12:00-11:00 AM/PM UTC.

Specify cron format in the following order:

" * * * * * "

These five values indicate when the job should be executed. These values are mandatory and must be specified in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

Field	Allowed values	Allowed special characters
Minute	0-59	* / , -
Hour	0-23	* / , -
Day of month	1-31	* / , - ?
Month	1-12 or JAN-DEC	* / , -
Day of week	0-6 or SUN-SAT	* / , - ?

Adding Key Refresh Schedule

To add a key refresh schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. Under Select Schedule Type, select **Key Refresh**.
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **DSM** from the **Cloud Name** drop-down list.
 - (Optional) Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. On the **Schedule Config** screen, enter or select the following details.

- a. Configure **DURATION** for the scheduler. Specify the following values:

- **Schedule Starts**
- **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.

- b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will refresh keys. Select either **Basic** or **Raw (Cron)** format to specify the value.

- c. Click **Next**.

8. On the **Add Domains** screen:

- a. Select the domain name from the **Refresh Keys from Selected Domain(s)** drop-down list.
- b. Click the **+** button.

Similarly, you can add more domains.

- c. Select the **Refresh All Domains** option if you want to refresh all the existing domains.
- d. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Viewing/Editing Schedules

To view/edit a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The Schedules page displays following details:

Field	Description
Name	Unique name of the scheduler configuration.
Schedule Type	Type of the schedule. Possible types are: • Key Rotation • Key Refresh • Add KMS Container (GCP and Azure only) For DSM, only Key Refresh is supported.
Cloud Name	Name of the cloud. For DSM schedules, it is DSM.
Last Modified	Time when the schedule is modified.
Frequency	Frequency of the scheduler configuration.

Field	Description
Start Date	Start time of the scheduler configuration.
End Date	Expiry time of the scheduler configuration.
Status	Status of the scheduler configuration. Possible values are: • Enabled • Disabled

- Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.
- Edit or configure the following fields:

GENERAL INFO: Update the general information about the schedule:

- Update **Description** of the schedule.
- Enable or disable the scheduler configuration. Select or clear the **Enable Schedule** check box.

SCHEDULE CONFIG: Update the scheduler configuration parameters such as:

- DURATION:** Specify the period in which a schedule actively runs its jobs. Specify **Schedule Starts** and **Schedule Ends** dates. If you select **Never**, the schedule never expires.
- FREQUENCY:** Specify how often the schedule refreshes domains. Use **Basic** or **RAW (Cron)** option to specify the frequency.

DOMAINS: Under **Refresh Keys from Selected Domain(s)**:

- Add a new domain to the key refresh schedule.
- Remove an existing domain from the key refresh schedule.

- Click **Update** under each modified section.

The Schedules page also contains a section named **JOB HISTORY** that contains all the related information about a schedule such as **Run Date**, **Job ID**, **Status**, and **Errors**.

Disabling Schedules

To disable a schedule configuration:

- Open the **Cloud Key Manager** application.
- In the left pane, click **Schedules**. The list of available schedules is displayed.
- Click the overflow icon () corresponding to the desired schedule and click **Disable**. The Disable Schedule message is displayed.

-
4. Click **Disable Schedule**.

A message **Successfully disabled the schedule** is displayed on the screen.

Enabling Schedules

To enable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Enable**. The Enable Schedule message is displayed.
4. Click **Enable Schedule**.

A message **Successfully enabled the schedule** is displayed on the screen.

Manually Running Schedules

To manually run a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Run Now**.

The Run Now schedule is started in the background. A message **Schedule is running now. It will take a few seconds to finish**. is displayed on the screen. After the schedule is run successfully, a message **Successfully finished running the schedule** is displayed.

Deleting Schedules

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Delete Schedule**. The Delete Schedule message is displayed.
4. Click **Delete**.

A message **Successfully deleted schedule** is displayed on the screen.

Google Cloud Resources

Google Cloud resources include Google Cloud projects, key rings, and keys. Manage Google Cloud keys stored in key rings linked to Google Cloud projects in your Google Cloud service account using CCKM.

The high-level steps are:

1. Complete the [prerequisites](#).
2. Fetch the list of Google Cloud projects linked with your account. Note down the project IDs, they are needed when viewing and creating key rings inside projects.
3. Get the list of available Google Cloud locations (regions). These are required when viewing the key rings linked with a project in a location.
4. View the list of key rings in a linked Google Cloud project. Specify the connection name, project ID, and project location. You will add desired key rings to the CipherTrust Manager.
5. Add the desired key rings to the CipherTrust Manager. Specify the connection name, project ID, and key rings (array).

The key rings are displayed on the CipherTrust Manager. You can manage the added key rings and their keys from CCKM.

6. Manage keys. Add new native Google Cloud keys, create and upload keys based on external key source, manage key versions, and remove keys.

Note

CCKM doesn't support Google BYOK for Google Trusted Partner Clouds (TPC).

Prerequisites

Before you can manage Google Cloud resources on CCKM:

1. [Create a Project Using an Organization](#)
2. [Create a Project Using GCP User Not Belonging to an Organization](#)
3. [Enable Billing on the Project](#)
4. [Enable the Required APIs](#)
5. [Create a Key Ring](#)
6. [Create a Service Account](#)
7. [Grant the Service Account Access to Resources](#)
8. [Add Google Connection on CipherTrust Manager](#)

Note

You must increase the number of CCKM KMS API quotas for the projects you plan to add to CCKM. The recommended quota range is 800–1000 read/write operations. For more information, see [View and manage quotas](#).

Create a Project Using an Organization

The Organization resource is the root in the Google Cloud Platform (GCP) resource hierarchy. An organization can contain folders which can contain projects which can contain resources such as Google Cloud KMS. Refer to [Google Cloud resource hierarchy](#) for details.

To get an organization resource, sign up for Google Workspace or Cloud Identity and verify your domain. One Google Workspace or Cloud Identity account contains exactly one organization resource.

After you have an organization, you can optionally create folders to organize your projects. Finally, create a project. Refer to [Creating and Managing Organizations](#) for details.

Create a Project Using GCP User Not Belonging to an Organization

It is recommended to create a project using a GCP user that does not belong to an organization.

An existing GCP user can create projects even if the user does not belong to an organization. The projects and all resources under them are tied to the user account that is an employee in the company. If the employee leaves the company, the project is deleted with the user.

When creating a project with a user account, the default organization is "No Organization".

Enable Billing on the Project

If using an Organization, make sure you have the Billing Account Administrator role before performing the following steps.

1. Sign in to the [Manage billing accounts](#) page in the Google Cloud Console.
2. Add a billing account. Refer to [Create, modify, or close your Cloud Billing account](#) for details.
3. On the Billing page, click the **MY PROJECTS** tab to view the list of projects.
4. Click the menu (three vertical dots) for the project that you want to enable billing for.
5. Select **Change billing**.
6. Select the desired billing account.
7. Click **Set account**.

Refer to [Enable, disable, or change billing for a project](#) for more details on managing billing for a project.

Enable the Required APIs

1. Sign in to the [Google Cloud Console](#).
2. In the search bar **Search products and resources**, search for:
 - Cloud Key Management Service (KMS) API and enable the API.
 - Cloud Resource Manager API and enable the API.

Create a Key Ring

1. Sign in to the [Google Cloud Console](#).
2. Go to the left navigation menu.
3. Click **Security > Key Management**.
4. Create a key ring.

Create a Service Account

1. Sign in to the [Google Cloud Console](#).
2. Go to the left navigation menu.
3. Click **IAM & Admin > Service Accounts**.
4. Create a service account.

Grant the Service Account Access to Resources

IAM permissions are grouped into roles. Roles are assigned to members on resources. GCP provides predefined roles, but you can create your own custom roles, if required.

To manage Google Cloud keys in the key ring you created [above](#) using CCKM, assign the following roles to the service account:

- The **Browser** (`roles/browser`) role on the project that contains the key ring
- The **Cloud KMS Admin** (`roles/cloudkms.admin`) role on the key ring you created
- The **Logs Viewer** (`roles/logging.viewer`) role on the key ring you created

Note

- Search for the roles listed above using `name:roles/<rolename>`

- You can also assign the service account the role of **Resource Manager Organization Viewer** (`roles/resourcemanager.organizationViewer`). Assigning this role allows CCKM users to see the display name of the organization resource, instead of seeing only its ID.

Permissions are inherited down the GCP resource hierarchy. Therefore, you can also grant the service account the previous roles at the organization level, folder level, or project level.

To create custom roles for your service account, refer to the following mapping of CCKM actions to IAM KMS permissions.

CCKM Action	Cloud IAM KMS Permissions
Synchronize	cloudkms.keyRings.list cloudkms.keyRings.getIamPolicy cloudkms.cryptoKeys.getIamPolicy cloudkms.cryptoKeys.list cloudkms.cryptoKeyVersions.list
Create key	cloudkms.cryptoKeys.create
Upload key	cloudkms.cryptoKeys.create cloudkms.importJobs.create cloudkms.importJobs.setIamPolicy cloudkms.importJobs.get cloudkms.importJobs.useToImport cloudkms.cryptoKeyVersions.create cloudkms.cryptoKeys.update
Add key version/Rotate native	cloudkms.cryptoKeyVersions.create cloudkms.cryptoKeys.update
Add key version/ Rotate BYOK	cloudkms.importJobs.create cloudkms.importJobs.setIamPolicy cloudkms.importJobs.get cloudkms.importJobs.useToImport cloudkms.cryptoKeyVersions.create cloudkms.cryptoKeys.update
Update key	cloudkms.cryptoKeys.update
Update key version	cloudkms.cryptoKeys.update cloudkms.cryptoKeyVersions.update
Update key policy	cloudkms.cryptoKeys.getIamPolicy cloudkms.cryptoKeys.setIamPolicy cloudkms.cryptoKeys.update
Schedule delete key material	cloudkms.cryptoKeyVersions.destroy
Cancel schedule delete key material	cloudkms.cryptoKeyVersions.restore
Get the public key of an asymmetric key	cloudkms.cryptoKeyVersions.viewPublicKey

Add Google Connection on CipherTrust Manager

Before you can add a Google Cloud key ring to the CCKM, a connection to your Google Cloud service account must already exist on the CipherTrust Manager. A CipherTrust Manager administrator manages connections to external resources on the **Access Management > Connections Management** page of the CipherTrust Manager GUI. Refer to [Connection Manager](#) for details.

When configuring the connection, the CipherTrust Manager Administrator requires a key file (a JSON file). This file can be generated on your Google Cloud Console. Refer to the Google Cloud documentation for details.

Warning

Thales strongly discourages creating a Google connection using a service account key file that grants permission to [root of trust keys](#).

After the connection is configured, you can view the linked Google Cloud projects and manage the key rings in those projects. You can manage Google Cloud key rings and keys on the CipherTrust Manager.

Refer to the following sections:

- [Google Cloud Projects](#)
- [Google Cloud Key Rings](#)
- [Google Cloud Keys](#)
- [Google Cloud Reports](#)
- [Scheduling Operations](#)
- [Downloading Google Metadata](#)

Google Cloud Projects

This section describes how to manage Google Cloud projects on CCKM. Before proceeding, make sure that a connection to a Google Cloud service account is configured on the CipherTrust Manager.

After you have configured a connection, Google Cloud projects can be added, modified, or deleted on the Projects tab of the Google page.

Adding a Google Cloud Project

To add a Google Cloud project on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**. The Google page contains two tabs: Key Rings and Projects.

3. Click the **Projects** tab.
4. Click **Add Existing Project**. The **Add Existing Google Project** screen displays.
5. Under **Select Method**, select **Select From List**. This option allows you to select an existing project from a list of connections stored in CipherTrust Connection Manager.

The **Manually Enter Project ID** option allows you manually enter a project ID to connect to a project without credentials. This option is only applicable if you are integrating the Google Cloud External Key Manager (EKM) service with CCKM to create and manage endpoints in CCKM for Google Cloud EKM service to access Key Encryption Keys (KEKs).

6. Select the desired connection from the **Connection** drop-down list. The linked Google Cloud projects are auto-populated in the **Project ID** drop-down list.
7. Select the desired project ID from the **Project ID** drop-down list.
8. (Optional) Use the **Enable success audit events** toggle to enable or disable audit recording of successful operations within the given Google cloud project. This toggle is set to enable, by default.
9. Click **Add Project**.

A success message is displayed on the screen and the selected project ID is displayed on the **Projects** tab.

Viewing Google Projects

The Projects tab shows the list of existing Google projects. Search for projects by Name or Connection.

To view the list of Google projects available on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**. The Google page displays the Google Cloud projects and key rings added to CCKM, not all available on Google Cloud.
3. Click the **Projects** tab. The list of projects added to CCKM is displayed. The tab displays the following details:

Field	Description
Project ID	ID of the project.
Name	Name of the project.
Connection	Name of the Google Cloud connection with the CipherTrust Manager.
Organization	Organization of the project.
Date Added	Time when the project was added to CCKM.

To view the custom columns, click the Customize View () icon, select the desired option, and click **OK** to display

the column.

Viewing Details of a Google Cloud Project

To view the details of a Google Cloud project on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**.
3. Click the **Projects** tab. The list of projects added to CCKM is displayed.
4. On the Projects tab, click the **Name** link of the desired project.

Alternatively, click the overflow icon () corresponding to the desired project, and click **View/Edit**.

The edit view of the Google Projects page shows additional details of the selected project under the ACCESS CONTROL and GENERAL INFO sections.

Update General Information for a Google Project

From the details page of a Google project, the **Connection** name, which is uneditable, is displayed along with the **Enable success audit events** toggle. This toggle allows you change the setting for the audit recording of successful EKM operations within a Google project by enabling or disabling this feature.

To update the Google project

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**.
3. Click the **Projects** tab. The list of projects added to CCKM is displayed.
4. On the Projects tab, click the **Name** link of the desired project.

Alternatively, click the overflow icon () corresponding to the desired project, and click **View/Edit**.

The edit view of the Google Projects page shows additional details of the selected project under the ACCESS CONTROL and GENERAL INFO sections.

5. Under GENERAL INFO, use the **Enable success audit events** toggle to change the current setting.
6. Click **Update**.

Managing User Permissions on Google Cloud Projects

To work with the Google Cloud, users/group must have the minimum set of permissions that allow them to use the Google Cloud resources, such as keys and Google Cloud projects. Initially, the user only has permission to view the keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are a member of the CCKM Users group will be granted permissions to perform operations on Google Cloud projects.

Adding Permissions for a User/Group

To add permissions for a user/group:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**.
3. Click the **Projects** tab.
4. On the Projects tab, click the **Name** link of the desired project.

Alternatively, click the overflow icon () corresponding to the desired project, and click **View/Edit**.

5. In the ACCESS CONTROL section, click **Assign User/Group**. The **Assign User/Group** dialog box is displayed.
6. Select the desired user or group from the **User/Group** drop-down list.
7. Click **Save**.

The newly added user/group is displayed under **Name** in the ACCESS CONTROL section. You can now grant additional permissions to the user/group, as appropriate. Refer to [Granting Permission to Perform an Operation](#) for details.

Allowed Operations

CCKM allows the following operations on Google Cloud projects for CryptoSpace(s): View, Create, Edit, Delete, Block, Unblock, and Unassign.

CCKM allows the following operations on Google Cloud projects for CryptoSpace Endpoints: View, Enable, Disable, and Unassign.

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. In the ACCESS CONTROL section, click **CryptoSpace(s)** or **CryptoSpace Endpoints** tab.
2. Select the check box under the desired operation corresponding to the desired users or groups.
3. Click **Update**.

A success message is displayed on the screen.

To revoke permissions from a user/group, refer to [Removing a Permission](#) for details.

Removing a Permission

To remove a permission assigned to a user or group:

1. In the ACCESS CONTROL section, click **CryptoSpace(s)** or **CryptoSpace Endpoints** tab.
2. Clear the check box under the desired operation corresponding to the desired users or groups.
3. Click **Update**.

A success message is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. In the ACCESS CONTROL section, click **CryptoSpace(s)** or **CryptoSpace Endpoints** tab.
2. Under **Unassign**, click the **X** button corresponding to the desired user/group.
3. On the **Remove User / Remove Group** screen, click **Remove**.

Note

Removing this user/group will remove all permissions currently assigned to the user/group.

4. Click **Remove** to confirm the action. To cancel the change, click **Cancel**.

A success message is displayed on the screen.

Removing a Google Cloud Project

Google Cloud projects can be removed from the Projects tab of the Google page. Search for the Google Cloud projects using Name or Connection.

To remove a Google Cloud project from CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**.
3. Click the **Projects** tab. The list of projects added to CCKM is displayed.
4. Click the overflow icon () corresponding to the project you want to remove.

Warning

The removed project's keys will no longer be available on the Google Keys page, but the keys will still exist on the Google Cloud. If you later add this project with the same ID, the keys will be available again.

5. Select **I wish to delete this project**.

6. Click **Delete Project**.

A message **Project deleted successfully** is displayed on the screen.

Google Cloud Key Rings

This section describes how to manage Google Cloud key rings on CCKM. Before proceeding, a connection to your Google Cloud account must exist on the CipherTrust Manager. Refer to [Connection Manager](#) for details.

After the connection is configured, you can add key rings linked with a project on the CipherTrust Manager. Google Cloud key rings can be added, viewed, modified, or deleted on the Key Rings tab of the Google page.

Adding an Existing Google Cloud Key Ring

Adding an existing Google Cloud key ring to CCKM requires selecting a Google Cloud project and location of existing key rings. Based on these, the list of relevant key rings is retrieved from the Google Cloud project.

To add an existing Google Cloud key ring to CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**. The Google page contains two tabs: Key Rings and Projects. Key Rings is the default tab.
3. Click **Add Existing Key Ring**. The Select Connection Information screen is displayed.
4. Select the Google **Connection**. The linked Google Cloud projects are auto-populated in the **Project ID** drop-down list.
5. Select the **Project ID** where the Google Cloud key ring resides. The project locations are auto-populated in the **Location** drop-down list.
6. Select the key ring **Location**.
7. Click **Next**. The Add Key Rings screen is displayed. The screen shows the available key rings based on the Connection, Project ID, and Location you selected on the previous screen. Only the key rings not already added to CCKM are displayed.
8. Select the desired key rings.

Tip

To select multiple key rings, select the check boxes corresponding to them.
To select all the retrieved key rings, select the check box to the left of the Selected label (over the **Name** heading).

- Click **Save**.

The selected key rings are displayed on the Key Rings tab.

Viewing Key Rings

The Key Rings tab of the Google page shows the list of key rings available in the projects added to the CipherTrust Manager. Search for the key rings by Name or Connection.

To view the list of key rings available on CCKM:

- Open the **Cloud Key Manager** application.
- In the left pane, click **KMS Containers > Google**. The Google page displays the Google Cloud projects and key rings added to CCKM, not all available on Google Cloud.

The Key Rings tab shows the list of key rings added to CCKM is displayed. The tab displays the following details:

Field	Description
Name	Name of the key ring.
Location	Location of the project linked with the key ring.
Project ID	ID of the linked project.
Organization	Organization of the key ring.
Connection	Name of the Google Cloud connection with the CipherTrust Manager.
Last Refreshed	Time when the key ring was refreshed the last.
Created	Time when the key ring was created.

To view the custom columns, click the Customize View (⚙️) icon, select the desired option, and click **OK** to display the column.

Refreshing Key Rings

Refreshing is the process to download keys created in Google Cloud key rings to the CCKM. You can refresh

keys from individual or all Google Cloud key rings.

- [Refreshing Specific Key Rings](#)
- [Refreshing All Key Rings](#)

Refreshing Specific Key Rings

To refresh a key ring:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**. The Google page is displayed. The Key Rings tab displays the key rings added to CCKM.
3. Click the overflow icon () corresponding to the desired key ring and click **Refresh Now**.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Google > Google Keys** page. Refer to [Viewing Google Cloud Keys](#) for details.

Refreshing All Key Rings

To refresh all key rings:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**. The Google page is displayed. The Key Rings tab displays the key rings added to CCKM.
3. On the Key Rings tab, click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all key rings is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Google > Google Keys** page. Refer to [Viewing Google Cloud Keys](#) for details.

Viewing Details of a Key Ring

To view the details of a key ring on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**.
3. On the Key Rings tab, click the **Name** link of the desired key ring.

Alternatively, click the overflow icon () corresponding to the desired key ring, and click **View/Edit Details**.

The edit view of the Google Key Rings page shows additional details of the selected key ring including the configured ACCESS CONTROL.

Managing User Permissions on Google Cloud Key Rings

To work with the Google Cloud, users/group must have the minimum set of permissions that allow them to use the Google Cloud resources such as keys and Google Cloud key rings. Initially, the user only has permission to view the keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform operations on Google Cloud key rings.

Adding Permissions for a User/Group

To add permissions for a user/group:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**.
3. On the Key Rings tab, click the **Name** link of the desired key ring.

Alternatively, click the overflow icon () corresponding to the desired key ring, and click **View/Edit Details**.

4. In the ACCESS CONTROL section, click **Assign User/Group**. The **Assign User/Group** dialog box is displayed.
5. Select the desired user or group from the **User/Group** drop-down list.
6. Click **Save**.

The newly added user/group is displayed under **Name** in the ACCESS CONTROL section. You can now grant additional permissions to the user/group, as appropriate. Refer to [Granting Permission to Perform an Operation](#) for details.

Allowed Operations

CCKM allows the following operations on Google Cloud key rings:

-
- View Keys, Add Key, Edit / Disable/ Enable, Synchronize, Upload
 - Schedule Destroy, Cancel Destroy

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. In the ACCESS CONTROL section, select the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

To revoke permissions from a user/group, refer to [Removing a Permission](#) for details.

Removing a Permission

To remove a permission assigned to a user or group:

1. In the ACCESS CONTROL section, clear the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. In the ACCESS CONTROL section, under **Unassign**, click the **X** button corresponding to the desired user/group.
2. On the **Remove User / Remove Group** screen, click **Remove**.

Note

Removing this user/group will remove all permissions currently assigned to the user/group.

3. Click **Remove** to confirm the action. To cancel the change, click **Keep It**.

A success message is displayed on the screen.

Removing a Google Cloud Key Ring

Google Cloud key rings can be removed from the Key Rings tab of the Google page. Search for existing key rings using Name or Connection.

Note

When a Google Cloud key ring is removed from CCKM, it is also deleted from Google Cloud.

To remove a key ring from CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**. The Google page is displayed. The list of key rings added to CCKM is displayed.
3. Click the overflow icon () corresponding to the key ring you want to remove.
4. Click **Delete**.

Google Cloud Keys

This section describes how to manage Google Cloud keys on CCKM. Before proceeding, you must have a Google Cloud key ring added to the CCKM. Refer to [Google Cloud Key Rings](#) for details.

- [Key Types](#)
- [Key Creation Methods and Sources](#)
- [Creating/Uploading New Key Material](#)
- [Cloning Existing Key Material](#)
- [Viewing Google Cloud Keys](#)
- [Refreshing Google Cloud Keys](#)
- [Viewing Versions of a Key](#)
- [Disabling a Key Version](#)
- [Enabling a Key Version](#)
- [Downloading Public Key of an Asymmetric Version](#)
- [Scheduling Destruction of a Key Version](#)
- [Canceling Scheduled Destruction of a Key Version](#)

-
- [Enabling All Versions of a Key](#)
 - [Disabling All Versions of a Key](#)
 - [Scheduling Destruction of All Versions of a Key](#)
 - [Canceling Scheduled Destruction of All Key Versions](#)
 - [Adding a Key Version](#)
 - [Viewing or Editing Details of Google Cloud Keys](#)
 - [Re-Importing Google Cloud Keys](#)

Key Types

CCKM supports two types of Google Cloud keys:

- **Symmetric:** A randomly generated key is used to encrypt and decrypt the data.
- **Asymmetric:** A public and private RSA key pair is used to encrypt and decrypt the data. The public key encrypts while the private key decrypts the data.

Key Creation Methods and Sources

Methods to create Google Cloud keys using CCKM are:

- **Creating/Uploading New Key Material:** Add key material by creating and uploading new source key or creating new native key. The key source can be:
 - **CipherTrust (External):** A new key is first created on the external CipherTrust Manager. Then, this key material is uploaded to Google Cloud to create a new Google Cloud key. As the key material is uploaded from the external CipherTrust Manager, the key origin is **CipherTrust (External)**.
 - **CipherTrust (Local):** A new key is first created on the CipherTrust Manager. Then, this key material is uploaded to Google Cloud to create a new Google Cloud key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.
 - **Google (Native):** A new key is directly created on Google Cloud using the native Google application. The key origin is **NATIVE**.
 - **Luna HSM:** A new Luna HSM key is first created on the CipherTrust Manager. Then, this key material is uploaded to Google Cloud to create a new Google Cloud key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

- **Vormetric DSM:** A new DSM key is first created on the CipherTrust Manager. Then, this key material is uploaded to Google Cloud to create a new Google Cloud key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.
- **Cloning Existing Key Material:** Clone key material from an existing key to create a new key. The key source can be:
 - **CipherTrust (Local):** An existing local CipherTrust Manager key is first cloned on the CipherTrust Manager. Then, the cloned key material is uploaded to Google Cloud to create a new Google Cloud key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.
 - **CipherTrust (External):** An existing key is first cloned on the external CipherTrust Manager. Then, this key material is uploaded to Google Cloud to create a new Google Cloud key. As the key material is uploaded from the external CipherTrust Manager, the key origin is **CipherTrust (External)**.
 - **Luna HSM:** An existing Luna HSM key is first cloned on the CipherTrust Manager. Then, the key material is uploaded to Google Cloud to create a new Google Cloud key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

- **Vormetric DSM:** An existing DSM key is first cloned on the CipherTrust Manager. Then, the key material is uploaded to Google Cloud to create a new Google Cloud key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.

Creating/Uploading New Key Material

To add a Google Cloud key by creating/uploading new key material:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click **Add Key**. The Select Material Origin screen of the Add Google Key wizard is displayed.

-
- Under **Select Method**, select **Create/Upload New Key Material**. The **Select Source** section appears. Depending on your requirements, select from the following:

- **CipherTrust (External)**: Refer to [Uploading CipherTrust \(External\) Key Material](#) for details.
- **CipherTrust (Local)**: Refer to [Uploading CipherTrust \(Local\) Key Material](#) for details.
- **Google (Native)**: Refer to [Creating Google \(Native\) Key Material](#) for details.
- **Luna HSM**: Refer to [Uploading Luna HSM Key Material](#) for details.
- **Vormetric DSM**: Refer to [Uploading Vormetric DSM Key Material](#) for details.

Refer to [Key Creation Methods and Sources](#) for details on key sources.

Uploading CipherTrust (External) Key Material

Upload the key material of an external CipherTrust Manager key using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

- Select **CipherTrust (External)**.
- Click **Next**. The Configure CipherTrust (External) Key screen is displayed.

Configure CipherTrust (External) Key

- Select Key Type**. The options are **Symmetric** and **Asymmetric**.
 - **Symmetric**: Creates and uploads a symmetric key.
 - **Asymmetric**: Creates and uploads an asymmetric key. Additional fields **Algorithm** and **Key Size** are displayed.
- Select Domain** from the drop-down list. The drop-down list shows the external CM domains linked with the configured external CM connection.
- Enter a **Key Name**. A new key with this name will be created on the CipherTrust Manager and its key material will be uploaded to Google Cloud.
- (Applicable to **Asymmetric** keys) Select an **Algorithm** for the key.
- (Applicable to **Asymmetric** keys) Select a **Key Size / Key Curve** for the key. The options vary based on the selected **Algorithm**.

- Click **Next**.

Configure Destination (Google) Key

- (Applicable to **Asymmetric** keys) Select the **Key Purpose**. A key purpose specifies the operation that the key

can be used to perform. Depending on the key algorithm selected on the previous screen, the options are:

- **Decrypt:** Enables the key for RSA encryption. This option is not applicable to EC algorithms.
 - **Sign:** Enables the key for elliptic curve signing and RSA signing.
2. Enter a user-friendly alias as the **Key Name**. This will be the key name on Google Cloud. This name helps uniquely identify a Google Cloud key. By default, the Key Name you specified on the previous screen is populated.

The key name can only contain alphanumeric characters and dashes.
 3. Select the desired **Key Ring** from the drop-down list.
 4. (Optional) Select **Restrict key versions to import only** to specify that the newer versions of the key can only be imported. After key creation, this field will become non-editable. Google native key rotation will not be allowed for them. That is, the **Google Automatic Rotation** check box will be disabled for them.
 5. (Optional) Configure the duration for which the key can remain in the soft-deleted state before it is destroyed from CCKM. The default duration is 30 days.
 - a. Select **Soft Delete Wait Time**.
 - b. Specify **Soft Delete Wait Time (Days)**. The duration can range from 1 to 120 days. For import-only keys, the minimum duration is 0 days.

Note

- The duration applies to all versions of the key created in the future.
- The soft-delete wait time can't be modified after the key is created.
- The soft-deleted key will be removed from CCKM after the number of days specified in **Soft Delete Wait Time (Days)**.

6. (Optional, applicable to **Symmetric** keys) Select **Google Automatic Rotation** and specify the following:
 - a. Select **Rotation Period** from the drop-down list. This is the time period (in days) after which the key will rotate automatically, after the date specified in **Starting on** (see below). The options are 30, 90, 180, and 365 days.

To specify a custom frequency, select **Custom** and specify the rotation frequency in the **Rotate key every (days)** field that appears. The default value is **1** day.
 - b. Click the **Starting on** field, and select the key expiration date and time from the on-screen calendar.

This is the time when the automatic key rotation becomes effective. Next automatic key rotation will

happen after the specified **Rotation Period**.

For example, if you set **Rotation Period** as 30 days and set **Starting on** as today, the key will automatically rotate after 30 days from today.

7. Select a **Protection Level** for the key. A protection level specifies how cryptographic operations are performed. The options are:
 - **Software:** Crypto operations are performed in software.
 - **HSM:** Crypto operations are performed in an HSM.
8. (Applicable to **Asymmetric** keys) Select an **Algorithm** for the key. The options vary based on the **Algorithm**, **Key Size / Key Curve** (selected on the Configure Source Key screen), and key purpose.
9. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

Note

For Google label requirements, refer to [Labeling keys | Cloud KMS Documentation | Google Cloud](#).

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
- c. Click the + button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

10. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. **Select Schedule to Apply**. The drop-down list shows the available key rotation schedules.
 2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust (Local):** The local CipherTrust Manager.
 - **CipherTrust (External):** The external CipherTrust Manager.
 - **Luna:** HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM:** Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
 - **Native:** Native cloud, Google.
-

-
3. Select an **Algorithm**. For the **Native** Key Origin, the Algorithm field is unavailable.
 4. Click **Next**.

The Review and Add Key screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Google Key wizard is closed.

The newly created key is displayed in the list of Google Cloud keys.

Uploading CipherTrust (Local) Key Material

Upload the local key material using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

1. Select **CipherTrust (Local)**.
2. Click **Next**. The Configure CipherTrust (Local) Key screen is displayed.

Configure CipherTrust (Local) Key

1. **Select Key Type**. The options are **Symmetric** and **Asymmetric**.
 - **Symmetric**: Creates and uploads a symmetric key.
 - **Asymmetric**: Creates and uploads an asymmetric key. Additional fields **Algorithm** and **Key Size** are displayed.
2. Enter a **Key Name**. A new key with this name will be created on the CipherTrust Manager and its key material will be uploaded to Google Cloud.

3. (Applicable to **Asymmetric** keys) Select the **Algorithm**. The options are **RSA** and **EC**.
4. (Applicable to **Asymmetric** keys) Select the **Key Size / Key Curve** based on the algorithm:
 - For the **RSA** algorithm, select the **Key Size**. The options are 2048, 3072, and 4096.
 - For the **EC** algorithm, select the **Key Curve**. The options are prime256v1, secp384r1, and secp256k1.
5. Click **Next**.

Configure Destination (Google) Key

1. (Applicable to **Asymmetric** keys) Select the **Key Purpose**. A key purpose specifies the operation that the key can be used to perform. Depending on the key algorithm selected on the previous screen, the options are:
 - **Decrypt**: Enables the key for RSA encryption. This option is not applicable to EC algorithms.
 - **Sign**: Enables the key for elliptic curve signing and RSA signing.
2. Enter a user-friendly alias as the **Key Name**. This will be the key name on Google Cloud. This name helps uniquely identify a Google Cloud key. By default, the Key Name you specified on the previous screen is populated.

The key name can only contain alphanumeric characters and dashes.

3. Select the desired **Key Ring** from the drop-down list.
4. (Optional) Select **Restrict key versions to import only** to specify that the newer versions of the key can only be imported. After key creation, this field will become non-editable. Google native key rotation will not be allowed for them. That is, the **Google Automatic Rotation** check box will be disabled for them.
5. (Optional) Configure the duration for which the key can remain in the soft-deleted state before it is destroyed from CCKM. The default duration is 30 days.
 - a. Select **Soft Delete Wait Time**.
 - b. Specify **Soft Delete Wait Time (Days)**. The duration can range from 1 to 120 days. For import-only keys, the minimum duration is 0 days.

Note

- The duration applies to all versions of the key created in the future.
- The soft-delete wait time can't be modified after the key is created.
- The soft-deleted key will be removed from CCKM after the number of days specified in **Soft Delete Wait Time (Days)**.

6. (Optional, applicable to **Symmetric** keys) Select **Google Automatic Rotation** and specify the following:

- a. Select **Rotation Period** from the drop-down list. This is the time period (in days) after which the key will rotate automatically, after the date specified in **Starting on** (see below). The options are 30, 90, 180, and 365 days.

To specify a custom frequency, select **Custom** and specify the rotation frequency in the **Rotate key every (days)** field that appears. The default value is **1** day.

- b. Click the **Starting on** field, and select the key expiration date and time from the on-screen calendar.

This is the time when the automatic key rotation becomes effective. Next automatic key rotation will happen after the specified **Rotation Period**.

For example, if you set **Rotation Period** as 30 days and set **Starting on** as today, the key will automatically rotate after 30 days from today.

7. Select a **Protection Level** for the key. A protection level specifies how cryptographic operations are performed. The options are:

- **Software:** Crypto operations are performed in software.
- **HSM:** Crypto operations are performed in an HSM.

8. (Applicable to **Asymmetric** keys) Select an **Algorithm** for the key. The options vary based on the **Algorithm**, **Key Size / Key Curve** (selected on the Configure Source Key screen), and key purpose.

9. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

Note

For Google label requirements, refer to [Labeling keys | Cloud KMS Documentation | Google Cloud](#).

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
- c. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

10. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. **Select Schedule to Apply.** The drop-down list shows the available key rotation schedules.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust (Local):** The local CipherTrust Manager.
 - **CipherTrust (External):** The external CipherTrust Manager.
 - **Luna:** HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM:** Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
 - **Native:** Native cloud, Google.
3. Select an **Algorithm**. For the **Native** Key Origin, the Algorithm field is unavailable.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Google Key wizard is closed.

The newly created key is displayed in the list of Google Cloud keys.

Creating Google (Native) Key Material

Create the key material directly using native Google application.

Select Material Origin > Select Source

1. Select **Google (Native)**.

2. Click **Next**. The Configure Source Key screen is displayed.

Configure Source Key

1. **Select Key Type**. The options are:
 - **Symmetric**: Creates a symmetric key. Additional field **Google Automatic Rotation** is displayed.
 - **Asymmetric**: Creates an asymmetric key. Additional fields **Select Key Purpose** and **Algorithm** are displayed.
2. (Applicable to **Asymmetric** keys) **Select Key Purpose**. A key purpose specifies the operation that the key can be used to perform. The options are:
 - **Decrypt**: Enables the key for RSA encryption.
 - **Sign**: Enables the key for elliptic curve signing and RSA signing.
3. Enter the **Key Name**. This name helps uniquely identify a key.
4. Select the desired **Key Ring** from the drop-down list.
5. (Optional) Configure the duration for which the key can remain in the soft-deleted state before it is destroyed from CCKM. The default duration is 30 days.
 - a. Select **Soft Delete Wait Time**.
 - b. Specify **Soft Delete Wait Time (Days)**. The duration can range from 1 to 120 days. For import-only keys, the minimum duration is 0 days.

Note

- The duration applies to all versions of the key created in the future.
- The soft-delete wait time can't be modified after the key is created.
- The soft-deleted key will be removed from CCKM after the number of days specified in **Soft Delete Wait Time (Days)**.

6. (Optional, applicable to **Symmetric** keys) Select **Google Automatic Rotation** and specify the following:
 - a. Select **Rotation Period** from the drop-down list. This is the time period (in days) after which the key will rotate automatically, after the date specified in **Starting on** (see below). The options are 30, 90, 180, and 365 days.

To specify a custom frequency, select **Custom** and specify the rotation frequency in the **Rotate key every (days)** field that appears. The default value is **1** day.

-
- b. Click the **Starting on** field, and select the key expiration date and time from the on-screen calendar.

This is the time when the automatic key rotation becomes effective. Next automatic key rotation will happen after the specified **Rotation Period**.

For example, if you set **Rotation Period** as 30 days and set **Starting on** as today, the key will automatically rotate after 30 days from today.

7. Select a **Protection Level** for the key. A protection level specifies how cryptographic operations are performed. The options are:
 - **Software:** Crypto operations are performed in software.
 - **HSM:** Crypto operations are performed in an HSM.
8. (Applicable to **Asymmetric** keys) Select an **Algorithm** for the key. The options vary based on the selected key purpose.
9. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

Note

For Google label requirements, refer to [Labeling keys | Cloud KMS Documentation | Google Cloud](#).

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
- c. Click the + button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

10. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. **Select Schedule to Apply.** The drop-down list shows the available key rotation schedules.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust (Local):** The local CipherTrust Manager.
 - **CipherTrust (External):** The external CipherTrust Manager.
 - **Luna:** HSM Luna. Also select a partition from the **Select Partition** drop-down list.

- **DSM:** Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
- **Native:** Native cloud, Google.

3. Select an **Algorithm**. For the **Native** Key Origin, the Algorithm field is unavailable.

4. Click **Next**.

The Review and Add Key screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, NATIVE KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **NATIVE KEY** and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the NATIVE KEY and KEY SCHEDULES sections becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Google Key wizard is closed.

The newly created key is displayed in the list of Google Cloud keys. The origin of the key is **NATIVE**.

Uploading Luna HSM Key Material

Upload the key material using Luna HSM to configure the source key.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Select Material Origin > Select Source

1. Select **Luna HSM**.
2. Click **Next**. The Configure Source Key screen is displayed.

Configure Source Key

1. Enter a **Luna HSM Key Label**. A new key with this name will be created on the Luna HSM and its key material will be uploaded to Google Cloud.
2. Select the **Partition ID** of the desired Luna HSM partition.
3. Select the key **Mechanism**. The supported key mechanisms are:
 - CKM_RSA_FIPS_186_3_AUX_PRIME_KEY_PAIR_GEN
 - CKM_RSA_X9_31_KEY_PAIR_GEN
 - CKM_RSA_FIPS_186_3_PRIME_KEY_PAIR_GEN
 - CKM_RSA_PKCS_KEY_PAIR_GEN
4. Select the **Key Size** from the available options. The supported sizes are 2048, 3072, and 4096.
5. Select the **Key Attributes**. The options are:
 - Modifiable, Extractable, Sensitive (all *three* are selected for a **BYOK Compatible** key)
 - Encrypt, Decrypt, Wrap, Unwrap
 - Sign, Verify, Derive
6. Click **Next**.

Configure Destination (Google) Key

1. **Select Key Purpose**. A key purpose specifies the operation that the key can be used to perform. The options are:
 - **Decrypt**: Enables the key for RSA encryption. This option is not applicable to EC algorithms.
 - **Sign**: Enables the key for elliptic curve signing and RSA signing.
2. Enter a user-friendly alias as the **Key Name**. This will be the key name on Google Cloud. This name helps uniquely identify a Google Cloud key. By default, the Luna HSM Key Label you specified on the previous screen is populated.

The key name can only contain alphanumeric characters and dashes.
3. Select the desired **Key Ring** from the drop-down list.
4. (Optional) Select **Restrict key versions to import only** to specify that the newer versions of the key can only be imported. After key creation, this field will become non-editable. Google native key rotation will not be allowed for them.
5. (Optional) Configure the duration for which the key can remain in the soft-deleted state before it is destroyed from CCKM. The default duration is 30 days.

- a. Select **Soft Delete Wait Time**.
- b. Specify **Soft Delete Wait Time (Days)**. The duration can range from 1 to 120 days. For import-only keys, the minimum duration is 0 days.

Note

- The duration applies to all versions of the key created in the future.
- The soft-delete wait time can't be modified after the key is created.
- The soft-deleted key will be removed from CCKM after the number of days specified in **Soft Delete Wait Time (Days)**.

6. Select the **Protection Level** for the key. A protection level specifies how cryptographic operations are performed. The options are:
 - **Software**: Crypto operations are performed in software.
 - **HSM**: Crypto operations are performed in an HSM.
7. Select an **Algorithm** for the key. The options vary based on the key size and the purpose.
8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

Note

For Google label requirements, refer to [Labeling keys | Cloud KMS Documentation | Google Cloud](#).

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
- c. Click the + button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. **Select Schedule to Apply**. The drop-down list shows the available key rotation schedules.

2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust (Local)**: The local CipherTrust Manager.
 - **CipherTrust (External)**: The external CipherTrust Manager.
 - **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
 - **Native**: Native cloud, Google.
3. Select an **Algorithm**. For the **Native** Key Origin, the Algorithm field is unavailable.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Google Key wizard is closed.

The newly created key is displayed in the list of Google Cloud keys.

Uploading Vormetric DSM Key Material

Upload the key material using Vormetric DSM to configure the source key.

Select Material Origin > Select Source

1. Select **Vormetric DSM**.
2. Click **Next**. The Configure Source Key screen is displayed.

Configure Source Key

1. **Select Encryption Type.** The options are **Symmetric** and **Asymmetric**.
 - **Symmetric:** Creates and uploads a symmetric key.
 - **Asymmetric:** Creates and uploads an asymmetric key. Additional fields **Algorithm** and **Key Size** are displayed.
2. Enter a **DSM Key Name**. A new key with this name will be created on the DSM and its key material will be uploaded to Google Cloud.
3. (Optional) Provide a **Description** for the key.
4. Select a **DSM Domain** for the key. The drop-down list shows the DSM domains linked with the configured DSM connection.
5. (Optional) Select the **Expiration Date** check box, select the key expiration date and time from the on-screen calendar.
6. (Applicable to **Asymmetric** keys) Select an **Algorithm** for the key. The options are RSA-2048, RSA-3072, and RSA-4096.
7. Click **Next**.

Configure Destination (Google) Key

1. (Applicable to **Asymmetric** keys) **Select Key Purpose.** A key purpose specifies the operation that the key can be used to perform. The options are:
 - **Decrypt:** Enables the key for RSA encryption. This option is not applicable to EC algorithms.
 - **Sign:** Enables the key for elliptic curve signing and RSA signing.
2. Enter a user-friendly alias as the **Key Name**. This will be the key name on Google Cloud. This name helps uniquely identify a Google Cloud key. By default, the DSM Key Name you specified on the previous screen is populated.

The key name can only contain alphanumeric characters and dashes.
3. Select the desired **Key Ring** from the drop-down list.
4. (Optional) Select **Restrict key versions to import only** to specify that the newer versions of the key can only be imported. After key creation, this field will become non-editable. Google native key rotation will not be allowed for them. That is, the **Google Automatic Rotation** check box will be disabled for them.
5. (Optional) Configure the duration for which the key can remain in the soft-deleted state before it is destroyed from CCKM. The default duration is 30 days.
 - a. Select **Soft Delete Wait Time**.

- b. Specify **Soft Delete Wait Time (Days)**. The duration can range from 1 to 120 days. For import-only keys, the minimum duration is 0 days.

Note

- The duration applies to all versions of the key created in the future.
- The soft-delete wait time can't be modified after the key is created.
- The soft-deleted key will be removed from CCKM after the number of days specified in **Soft Delete Wait Time (Days)**.

6. (Optional, applicable to **Symmetric** keys) Select **Google Automatic Rotation** and specify the following:
 - a. Select **Rotation Period** from the drop-down list. This is the time period (in days) after which the key will rotate automatically, after the date specified in **Starting on** (see below). The options are 30, 90, 180, and 365 days.

To specify a custom frequency, select **Custom** and specify the rotation frequency in the **Rotate key every (days)** field that appears. The default value is **1** day.

- b. Click the **Starting on** field, and select the key expiration date and time from the on-screen calendar.

This is the time when the automatic key rotation becomes effective. Next automatic key rotation will happen after the specified **Rotation Period**.

For example, if you set **Rotation Period** as 30 days and set **Starting on** as today, the key will automatically rotate after 30 days from today.

7. Select a **Protection Level** for the key. A protection level specifies how cryptographic operations are performed. The options are:
 - **Software**: Crypto operations are performed in software.
 - **HSM**: Crypto operations are performed in an HSM.
8. (Applicable to **Asymmetric** keys) Select an **Algorithm** for the key. The options vary based on the key purpose.
9. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

Note

For Google label requirements, refer to [Labeling keys | Cloud KMS Documentation | Google Cloud](#).

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
- c. Click the + button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

10. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. **Select Schedule to Apply**. The drop-down list shows the available key rotation schedules.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust (Local)**: The local CipherTrust Manager.
 - **CipherTrust (External)**: The external CipherTrust Manager.
 - **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
 - **Native**: Native cloud, Google.
3. Select an **Algorithm**. For the **Native** Key Origin, the Algorithm field is unavailable.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as

appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Google Key wizard is closed.

The newly created key is displayed in the list of Google Cloud keys.

Cloning Existing Key Material

To add a new Google Cloud key by cloning key material existing on the CipherTrust Manager, Luna HSM, or the DSM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click **Add Key**. The Select Material Origin screen of the Add Google Key wizard is displayed.
4. Under **Select Method**, select **Clone Existing Key Material**. The **Select Source** section appears. Depending on your requirements, select from the following:
 - **CipherTrust (External)**: Refer to [Cloning CipherTrust \(External\) Key Material](#) for details.
 - **CipherTrust (Local)**: Refer to [Cloning CipherTrust \(Local\) Key Material](#) for details.
 - **Luna HSM**: Refer to [Cloning Luna HSM Key Material](#) for details.
 - **Vormetric DSM**: Refer to [Cloning Vormetric DSM Key Material](#) for details.

Refer to [Key Creation Methods and Sources](#) for details on these key sources.

Cloning CipherTrust (External) Key Material

Clone and upload the key material of an external CipherTrust Manager key using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

1. Select **CipherTrust (External)**.
2. Click **Next**. The Select CipherTrust (External) Key screen is displayed.

Select CipherTrust (External) Key

1. **Select Domain** from the drop-down list. The drop-down list shows the external CM domains linked with the configured external CM connection.
2. Select an **Algorithm**. The options are **AES**, **RSA**, and **EC**.
3. Select the **Key Size** based on the key type:
 - For an **AES** key, the option is 256.
 - For an **RSA** key, the options are 2048, 3072, and 4096.
 - For an **EC** key, the options are prime256v1, secp384r1, and secp256k1.
4. Select the desired **CipherTrust (External) Key** from the list.
5. Click **Next**. The Configure Destination (Google) Key screen is displayed.

Configure Destination (Google) Key

1. (Applicable to **Asymmetric** keys) **Select Key Purpose**. A key purpose specifies the operation that the key can be used to perform. The options are:
 - **Decrypt**: Enables the key for RSA encryption. This option is not applicable to EC algorithms.
 - **Sign**: Enables the key for elliptic curve signing and RSA signing.
2. Enter a user-friendly alias as the **Key Name**. This will be the key name on Google Cloud. This name helps uniquely identify a Google Cloud key. By default, the CipherTrust Key Name you specified on the previous screen is populated.

The key name can only contain alphanumeric characters and dashes.

3. Select the desired **Key Ring**. The drop-down list shows the key rings in the project location linked with the configured Google Cloud connection.
4. (Optional) Select **Restrict key versions to import only** to specify that the newer versions of the key can only be imported. After key creation, this field will become non-editable. Google native key rotation will not be allowed for them. That is, the **Google Automatic Rotation** check box will be disabled for them.
5. (Optional) Configure the duration for which the key can remain in the soft-deleted state before it is destroyed from CCKM. The default duration is 30 days.
 - a. Select **Soft Delete Wait Time**.
 - b. Specify **Soft Delete Wait Time (Days)**. The duration can range from 1 to 120 days. For import-only keys, the minimum duration is 0 days.

Note

- The duration applies to all versions of the key created in the future.
- The soft-delete wait time can't be modified after the key is created.
- The soft-deleted key will be removed from CCKM after the number of days specified in **Soft Delete Wait Time (Days)**.

6. (Optional) Select **Google Automatic Rotation** and specify the following:

- a. Select **Rotation Period** from the drop-down list. This is the time period (in days) after which the key will rotate automatically, after the date specified in **Starting on** (see below). The options are 30, 90, 180, and 365 days.

To specify a custom frequency, select **Custom** and specify the rotation frequency in the **Rotate key every (days)** field that appears. The default value is **1** day.

- b. Click the **Starting on** field, and select the key expiration date and time from the on-screen calendar.

This is the time when the automatic key rotation becomes effective. Next automatic key rotation will happen after the specified **Rotation Period**.

For example, if you set **Rotation Period** as 30 days and set **Starting on** as today, the key will automatically rotate after 30 days from today.

7. Select a **Protection Level** for the key. A protection level specifies how cryptographic operations are performed. The options are:

- **Software**: Crypto operations are performed in software.
- **HSM**: Crypto operations are performed in an HSM.

8. Select the **Algorithm** for the key.

9. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

Note

For Google label requirements, refer to [Labeling keys | Cloud KMS Documentation | Google Cloud](#).

To add a tag:

-
- a. Specify a tag name.
 - b. Specify the tag value.
 - c. Click the + button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

10. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. **Select Schedule to Apply**. The drop-down list shows the available key rotation schedules.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust (Local)**: The local CipherTrust Manager.
 - **CipherTrust (External)**: The external CipherTrust Manager.
 - **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
 - **Native**: Native cloud, Google.
3. Select an **Algorithm**. For the **Native** Key Origin, the Algorithm field is unavailable.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY and KEY SCHEDULES sections becomes Complete and the Key ID link is displayed, the key is created successfully.

-
3. Click **Close**. The Add Google Key wizard is closed.

The newly created key is displayed in the list of Google Cloud keys.

Cloning CipherTrust (Local) Key Material

Clone and upload the local key material using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

1. Select **CipherTrust (Local)**.
2. Click **Next**. The Select CipherTrust (Local) Key screen is displayed.

Select CipherTrust (Local) Key

1. Select an **Algorithm**. The options are **AES**, **RSA**, and **EC**.
2. Select the **Key Size** based on the key type:
 - For an **AES** key, the option is 256.
 - For an **RSA** key, the options are 2048, 3072, and 4096.
 - For an **EC** key, the options are prime256v1, secp384r1, and secp256k1.
3. Select the desired **CipherTrust (Local) Key** from the list.
4. Click **Next**. The Configure Destination (Google) Key screen is displayed.

Configure Destination (Google) Key

1. (Applicable to **Asymmetric** keys) **Select Key Purpose**. A key purpose specifies the operation that the key can be used to perform. The options are:
 - **Decrypt**: Enables the key for RSA encryption. This option is not applicable to EC algorithms.
 - **Sign**: Enables the key for elliptic curve signing and RSA signing.
2. Enter a user-friendly alias as the **Key Name**. This will be the key name on Google Cloud. This name helps uniquely identify a Google Cloud key. By default, the CipherTrust Key Name you specified on the previous screen is populated.

The key name can only contain alphanumeric characters and dashes.

3. Select the desired **Key Ring**. The drop-down list shows the key rings in the project location linked with the configured Google Cloud connection.
4. (Optional) Select **Restrict key versions to import only** to specify that the newer versions of the key can only be imported. After key creation, this field will become non-editable. Google native key rotation will not be allowed for them. That is, the **Google Automatic Rotation** check box will be disabled for them.

5. (Optional) Configure the duration for which the key can remain in the soft-deleted state before it is destroyed from CCKM. The default duration is 30 days.

- a. Select **Soft Delete Wait Time**.
- b. Specify **Soft Delete Wait Time (Days)**. The duration can range from 1 to 120 days. For import-only keys, the minimum duration is 0 days.

Note

- The duration applies to all versions of the key created in the future.
- The soft-delete wait time can't be modified after the key is created.
- The soft-deleted key will be removed from CCKM after the number of days specified in **Soft Delete Wait Time (Days)**.

6. (Optional) Select **Google Automatic Rotation** and specify the following:

- a. Select **Rotation Period** from the drop-down list. This is the time period (in days) after which the key will rotate automatically, after the date specified in **Starting on** (see below). The options are 30, 90, 180, and 365 days.

To specify a custom frequency, select **Custom** and specify the rotation frequency in the **Rotate key every (days)** field that appears. The default value is **1** day.

- b. Click the **Starting on** field, and select the key expiration date and time from the on-screen calendar.

This is the time when the automatic key rotation becomes effective. Next automatic key rotation will happen after the specified **Rotation Period**.

For example, if you set **Rotation Period** as 30 days and set **Starting on** as today, the key will automatically rotate after 30 days from today.

7. Select a **Protection Level** for the key. A protection level specifies how cryptographic operations are performed. The options are:

- **Software**: Crypto operations are performed in software.
- **HSM**: Crypto operations are performed in an HSM.

8. Select the **Algorithm** for the key.

9. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

Note

For Google label requirements, refer to [Labeling keys | Cloud KMS Documentation | Google Cloud](#).

To add a tag:

- a. Specify a tag name.
- b. Specify the tag value.
- c. Click the + button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

10. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. **Select Schedule to Apply**. The drop-down list shows the available key rotation schedules.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust (Local)**: The local CipherTrust Manager.
 - **CipherTrust (External)**: The external CipherTrust Manager.
 - **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
 - **Native**: Native cloud, Google.
3. Select an **Algorithm**. For the **Native** Key Origin, the Algorithm field is unavailable.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as

appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY and KEY SCHEDULES sections becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Google Key wizard is closed.

The newly created key is displayed in the list of Google Cloud keys.

Cloning Luna HSM Key Material

Clone and upload the Luna HSM key material using the CipherTrust Manager to configure the source key.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Select Material Origin > Select Source

1. Select **Luna DSM**.
2. Click **Next**. The Select Luna HSM Key screen is displayed.

Select Luna HSM Key

1. Select the **Key Size**. The options are 2048, 3072, and 4096.
2. Select the desired **HSM Key** from the list.
3. Click **Next**. The Configure Destination (Google) Key screen is displayed.

Configure Destination (Google) Key

1. **Select Key Purpose**. A key purpose specifies the operation that the key can be used to perform. The options are:
 - **Decrypt**: Enables the key for RSA encryption. This option is not applicable to EC algorithms.
 - **Sign**: Enables the key for elliptic curve signing and RSA signing.
2. Enter a user-friendly alias as the **Key Name**. This will be the key name on Google Cloud. This name helps uniquely identify a Google Cloud key. By default, the DSM Key Name you specified on the previous screen is populated.

The key name can only contain alphanumeric characters and dashes.

3. Select the desired **Key Ring**. The drop-down list shows the key rings in the project location linked with the configured Google Cloud connection.
4. (Optional) Select **Restrict key versions to import only** to specify that the newer versions of the key can only be imported. After key creation, this field will become non-editable. Google native key rotation will not be allowed for them.
5. (Optional) Configure the duration for which the key can remain in the soft-deleted state before it is destroyed from CCKM. The default duration is 30 days.
 - a. Select **Soft Delete Wait Time**.
 - b. Specify **Soft Delete Wait Time (Days)**. The duration can range from 1 to 120 days. For import-only keys, the minimum duration is 0 days.

Note

- The duration applies to all versions of the key created in the future.
- The soft-delete wait time can't be modified after the key is created.
- The soft-deleted key will be removed from CCKM after the number of days specified in **Soft Delete Wait Time (Days)**.

6. Select the **Protection Level** for the key. A protection level specifies how cryptographic operations are performed. The options are:
 - **Software**: Crypto operations are performed in software.
 - **HSM**: Crypto operations are performed in an HSM.
7. Select the **Algorithm** for the key. The options vary based on the key purpose.
8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

Note

For Google label requirements, refer to [Labeling keys | Cloud KMS Documentation | Google Cloud](#).

To add a tag:

-
- a. Specify a tag name.
 - b. Specify the tag value.
 - c. Click the + button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. **Select Schedule to Apply**. The drop-down list shows the available key rotation schedules.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust (Local)**: The local CipherTrust Manager.
 - **CipherTrust (External)**: The external CipherTrust Manager.
 - **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
 - **Native**: Native cloud, Google.
3. Select an **Algorithm**. For the **Native** Key Origin, the Algorithm field is unavailable.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY and KEY SCHEDULES sections becomes Complete and the Key ID link is displayed, the key is created successfully.

-
3. Click **Close**. The Add Google Key wizard is closed.

The newly created key is displayed in the list of Google Cloud keys.

Cloning Vormetric DSM Key Material

Clone and upload the DSM key material using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

1. Select **Vormetric DSM**.
2. Click **Next**. The Select DSM Key screen is displayed.

Select DSM Key

1. Select an **Algorithm**. The options are **AES** and **RSA**.
2. Select the **Key Size** based on the key type:
 - For an **AES** key, the option is 256.
 - For an **RSA** key, the options are 2048, 3072, and 4096.
3. Select the desired **DSM Key** from the list.
4. Click **Next**. The Configure Destination (Google) Key screen is displayed.

Configure Destination (Google) Key

1. Enter a user-friendly alias as the **Key Name**. This will be the key name on Google Cloud. This name helps uniquely identify a Google Cloud key. By default, the DSM Key Name you specified on the previous screen is populated.

The key name can only contain alphanumeric characters and dashes.

2. (Optional) Provide a basic **Description** of the key.
3. Select the desired **Key Ring**. The drop-down list shows the key rings in the project location linked with the configured Google Cloud connection.
4. (Optional) Select **Restrict key versions to import only** to specify that the newer versions of the key can only be imported. After key creation, this field will become non-editable. Google native key rotation will not be allowed for them. That is, the **Google Automatic Rotation** check box will be disabled for them.
5. (Optional) Configure the duration for which the key can remain in the soft-deleted state before it is destroyed from CCKM. The default duration is 30 days.
 - a. Select **Soft Delete Wait Time**.
 - b. Specify **Soft Delete Wait Time (Days)**. The duration can range from 1 to 120 days. For import-only keys,

the minimum duration is 0 days.

Note

- The duration applies to all versions of the key created in the future.
- The soft-delete wait time can't be modified after the key is created.
- The soft-deleted key will be removed from CCKM after the number of days specified in **Soft Delete Wait Time (Days)**.

6. (Optional) Select **Google Automatic Rotation** and specify the following:

- a. Select **Rotation Period** from the drop-down list. This is the time period (in days) after which the key will rotate automatically, after the date specified in **Starting on** (see below). The options are 30, 90, 180, and 365 days.

To specify a custom frequency, select **Custom** and specify the rotation frequency in the **Rotate key every (days)** field that appears. The default value is **1** day.

- b. Click the **Starting on** field, and select the key expiration date and time from the on-screen calendar.

This is the time when the automatic key rotation becomes effective. Next automatic key rotation will happen after the specified **Rotation Period**.

For example, if you set **Rotation Period** as 30 days and set **Starting on** as today, the key will automatically rotate after 30 days from today.

7. Select a **Protection Level** for the key. A protection level specifies how cryptographic operations are performed. The options are:

- **Software**: Crypto operations are performed in software.
- **HSM**: Crypto operations are performed in an HSM.

8. (Optional) Enter **Tags**. A tag is a label assigned to the key that consists of a user-defined key and a value.

Note

For Google label requirements, refer to [Labeling keys | Cloud KMS Documentation | Google Cloud](#).

To add a tag:

-
- a. Specify a tag name.
 - b. Specify the tag value.
 - c. Click the + button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

9. Click **Next**. The Add to Schedule screen is displayed.

Add to Schedule

1. **Select Schedule to Apply**. The drop-down list shows the available key rotation schedules.
2. Select the **Key Origin** from the available options. The key origin can be:
 - **CipherTrust (Local)**: The local CipherTrust Manager.
 - **CipherTrust (External)**: The external CipherTrust Manager.
 - **Luna**: HSM Luna. Also select a partition from the **Select Partition** drop-down list.
 - **DSM**: Vormetric Data Security Manager. Also select a domain from the **Select Domain** drop-down list.
 - **Native**: Native cloud, Google.
3. Select an **Algorithm**. For the **Native** Key Origin, the Algorithm field is unavailable.
4. Click **Next**.

The Review and Add Key screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, DESTINATION KEY, and KEY SCHEDULES sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY**, **DESTINATION KEY**, and **KEY SCHEDULES** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY and KEY SCHEDULES sections becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Google Key wizard is closed.

The newly created key is displayed in the list of Google Cloud keys.

Viewing Google Cloud Keys

The Google Keys page shows the list of Google Cloud keys available on the CipherTrust Manager.

To view the Google Cloud keys:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**. The list of available Google Cloud keys is displayed. The Google Keys page displays the following details:

Field	Description
Name	Unique, user-friendly name of the Google Cloud key. Click the link to view additional details of the key or edit the key. Refer to Viewing or Editing Details of Google Cloud Keys . This name is useful in searching for specific keys.
Status	State of the Google Cloud key. The status can be: • Available • Not Available • Deleted
Purpose	Purpose of the Google Cloud key. The purpose can be: •Encrypt Decrypt •Asymmetric Sign •Asymmetric Decrypt
Protection	Protection level of the Google Cloud key. The protection level can be Software or HSM.
Algorithm	Algorithm of the Google Cloud key. A number of algorithms are supported.
Key Ring	Name of the Google Cloud key ring where the key resides.
Location	Location where the Google Cloud key is created.
Project	Project where the Google Cloud key is created.
Organization	Organization where the Google Cloud key is created.
Creation Date	Date and time when the Google Cloud key is created.

Field	Description
Next Google Rotation	Date and time of the next Google Cloud key rotation.

Sometimes, you might notice certain Google Cloud keys are displayed as grayed out. This happens when the keys are no longer accessible. For example, when:

- Any cloud permissions on the keys are changed. The keys are no longer accessible from the Google Cloud connection.
- Connection is changed in KMS. The new connection does not have permissions to access the keys.
- When Google Cloud locations are changed or removed. The keys from the configured location are no longer accessible.

Refreshing Google Cloud Keys

Refreshing is the process of downloading keys created in Google Cloud key rings to CCKM. You can refresh keys from all Google Cloud key rings at once.

Refreshing All Keys

To refresh all keys:

1. Open the Cloud Key Manager application.
2. In the left pane, click **Cloud Keys > Google**. The Google Keys page is displayed. This page displays the list of Google Cloud keys.
3. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all keys is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message Refresh started... is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Google > Google Keys** page.

Refreshing Specific Keys

To refresh a key:

1. Open the Cloud Key Manager application.

2. In the left pane, click **Cloud Keys > Google**. The Google Keys page is displayed. This page displays the list of Google Cloud keys.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Refresh**.

A message Key refreshed successfully.

The refreshed key is listed on the **Cloud Keys > Google > Google Keys** page.

Viewing Versions of a Key

To view the versions of a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the expand icon () to the left of the desired key. The key version details are displayed:

Field	Description
Version	Version number of the key.
State	State of the key version. The state can be Aborted, Enabled, Disabled, Destroy Scheduled, or Destroyed.
Algorithm	Algorithm used for the key version.
Origin	Source of the key material used for the version. The origin can be: • CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud. • External (Unknown): Source of the key material is unknown. It is different than CCKM and the native cloud. Refer to Key Creation Methods and Sources for details.
Creation Date	Date and time when the key version is created.
Destroy Date	Date and time when the key version is destroyed.
Version Resource URL	URL of the key version resource on the CipherTrust Manager.

Disabling a Key Version

If required, you can disable an enabled version of a key. An aborted, destroy scheduled, or destroyed key version cannot be disabled.

A disabled version cannot operate on data. After the key version is disabled, there is a delay of up to a few hours during which it can still operate on data. You can again [enable](#) the version later. All versions of a key can also be disabled at once. Refer to [Disabling All Versions of a Key](#).

To disable a key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the expand icon () to the left of the desired key.
4. Click the overflow icon () corresponding to the desired version.
5. Click **Disable Version**. The Disable Key Version dialog box is displayed.
6. Click **Yes, Disable** to confirm the action.

The state of the key version changes to Disabled.

Enabling a Key Version

If required, you can enable a disabled version of a key. An aborted, destroy scheduled, or destroyed key version cannot be enabled.

To enable a key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the expand icon () to the left of the desired key.
4. Click the overflow icon () corresponding to the desired version.
5. Click **Enable Version**. The Enable Key Version dialog box is displayed.
6. Click **Yes, Enable** to confirm the action.

The state of the key version changes to Enabled.

Downloading Public Key of an Asymmetric Version

The public key of *asymmetric* versions of a Google Cloud key can be downloaded.

To download the public key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the expand icon () to the left of the desired key.
4. Click the overflow icon () corresponding to the desired *asymmetric* version.
5. Click **Get Public Key**.

The public key of (`public-key.pem`) the asymmetric key version is downloaded to your machine.

Scheduling Destruction of a Key Version

With CCKM, you can schedule destruction of a Google Cloud key version. The version will be destroyed from GCP after the configured soft-delete wait time. However, you can [cancel the schedule destruction](#) before the scheduled destruction time.

Caution

When a key version is destroyed from GCP, the version is immediately disabled. Any data encrypted or signed by the version cannot be decrypted or verified. However, the destroyed key can be recovered using the `post /v1/cckm/google/keys/{id}/versions/{versionID}/re-import` API. Refer to [Re-importing a Google Cloud Key Version](#) for details.

To schedule destruction of a key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the expand icon () to the left of the desired key.
4. Click the overflow icon () corresponding to the desired version.
5. Click **Schedule Destruction**. The Schedule Key Version Destruction dialog box is displayed.

Note

If you confirm scheduled destruction, the version will be destroyed from GCP after the configured soft-delete wait time.

6. Click **Yes, Schedule Destruction** to confirm the action.

The state of the key version changes to **Destroy Scheduled**. After the configured soft-delete wait time, the state will become **Destroyed**.

Canceling Scheduled Destruction of a Key Version

A scheduled destruction of a key version can be canceled before the destruction time arrives (that is, within the configured soft-delete wait time, after you scheduled destruction).

To cancel the schedule destruction of a key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the expand icon () to the left of the desired key.
4. Click the overflow icon () corresponding to the desired version with the **Destroy Scheduled** state.
5. Click **Cancel Destruction**. The Cancel Key Version Destruction dialog box is displayed.
6. Click **Yes, Cancel Destruction** to confirm the action.

The state of the key version changes to Disabled. The disabled key version can be enabled, if required. Refer to [Enabling a Key Version](#).

Enabling All Versions of a Key

With CCKM, you can enable all disabled versions of a key at once. Aborted, destroy scheduled, or destroyed key versions cannot be enabled.

To enable all disabled versions of a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Enable All Key Versions**. The Enable All Versions dialog box is displayed.
5. Click **Yes, Enable**.

All the key versions disabled earlier are now enabled. The status of the key versions changes to Enabled.

Disabling All Versions of a Key

With CCKM, you can disable all enabled versions of a key at once. Aborted, destroy scheduled, or destroyed key versions cannot be disabled.

A disabled version cannot operate on data. After the key version is disabled, there is a delay of up to a few hours during which it can still operate on data. You can again [enable](#) the version later. Individual versions of a key can also be disabled. Refer to [Disabling a Key Version](#).

To disable all enabled versions of a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Disable All Key Versions**. The Disable All Versions dialog box is displayed.
5. Click **Yes, Disable**.

All the key versions enabled earlier are now disabled. The status of the key versions changes to Disabled.

Scheduling Destruction of All Versions of a Key

With CCKM, you can schedule destruction of all versions of key at once. The versions will be destroyed from GCP after the configured soft-delete wait time. However, you can [cancel the schedule destruction](#) before the scheduled destruction time.

Caution

When a key version is destroyed from GCP, the version is immediately disabled. Any data encrypted or signed by the version cannot be decrypted or verified. However, the destroyed key can be recovered using the `post /v1/cckm/google/keys/{id}/versions/{versionID}/re-import` API. Refer to [Re-importing a Google Cloud Key Version](#) for details.

To schedule destruction of all versions of a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Destroy All Versions**. The Schedule Destruction dialog box is displayed.
5. Click **Yes, Schedule Destruction**.

All the key versions are scheduled for destruction. The status of the key versions changes to Destroy Scheduled.

Canceling Scheduled Destruction of All Key Versions

A scheduled destruction of all key versions can be canceled before the destruction time arrives (that is, within the configured soft-delete wait time, after you scheduled destruction).

To cancel the schedule destruction of all key versions:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.

3. Click the overflow icon () corresponding to the desired key.
4. Click **Cancel Destruction**. The Cancel Key Version Destruction dialog box is displayed.
5. Click **Yes, Cancel Destruction** to confirm the action.

The state of all enabled key versions changes to Disabled. The disabled key versions can be enabled, if required. Refer to [Enabling All Versions of a Key](#).

Adding a Key Version

CCKM provides two methods to add a new version to a key. Refer to [Key Creation Methods and Sources](#) for details on key creation methods and key sources.

To add a new key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Add Version**. The Add Version dialog box is displayed.
5. **Select Method**. The options are:
 - **Create/Upload New Key Material**: Refer to [Adding Key Version by Creating/Uploading Key Material](#).
 - **Clone Existing Key Material**: Refer to [Adding Key Version by Cloning Existing Key Material](#).

Adding Key Version by Creating New Key Material

1. Select **Create/Upload New Key Material** as the method.
2. **Select Source**. The options are:
 - **CipherTrust (Local)**: Select this option, select the **Algorithm**, and specify **Key Name** for the new key version.
 - **Google (Native)**: Select this option to create a new native Google Cloud key.
 - **Luna HSM**: Select this option, select the **Algorithm**, select the **Partition ID**, select **Key Attributes**, and specify **Key Name** for the new key version.

The key attributes Modifiable, Extractable, and Sensitive are selected for a **BYOK Compatible** key.

- **Vormetric DSM**: Select this option, select the **Algorithm**, select the **Domain**, and specify **Key Name** for the new key version.
3. Click **Add Version**.

A new version is added to the key.

Adding Key Version by Cloning Existing Key Material

1. Select **Clone Existing Key Material** as the method.
2. Select a **Source**. The options are:
 - **CipherTrust (External)**
 - **CipherTrust (Local)**
 - **Luna HSM**
 - **Vormetric DSM**
3. Select an **Algorithm**.
4. Select a **Key Size**.

Note

Depending on the algorithm of the existing key, the Algorithm and Key Size options for the new key version will vary.

5. Select a **Source Key** from the list.
6. Click **Save**.

A new version is added to the key.

Viewing or Editing Details of Google Cloud Keys

After a key is created, you can add or update tags and key rotation schedules, and also rotate key versions in the edit view. You can perform various operations on key versions.

In the edit view of a key, you can view all the key details such as its purpose, protection level, and location etc.

To view or edit an Google Cloud key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**. The list of available Google Cloud keys is displayed.
3. Click the overflow icon () corresponding to the desired key and click **View/Edit**. Alternatively, you can click the key name link. The edit view of the key is displayed. The edit view is divided into:
 - **LABELS:** View, add, and update labels. Refer to [Adding or Updating Labels](#).

- **ALGORITHM:** Update the default algorithm for the new key version. Refer to [Updating Default Algorithm for New Key Versions](#) for details.
- **ROTATION:** Update a key rotation schedule. Refer to [Updating Key Rotation Schedule](#) for details.
- **KEY VERSIONS:** Perform operations on key versions. Refer to [Performing Key Version Operations](#).
- **PERMISSIONS:** Manage the principles and roles. Refer to [Managing Principals and Roles](#).

Updating Default Algorithm for New Key Versions

To update the default algorithm for the new versions of the asymmetric key:

1. In the **ALGORITHM** section, select the desired algorithm from the **Select Default Algorithm for New Version** drop-down list.
2. Click **Update**.

The default algorithm is updated for the new versions of the key.

The key labels are updated.

Adding or Updating Labels

A label is a tag assigned to the key, which consists of a user-defined key and a value.

Note

For Google label requirements, refer to [Labeling keys | Cloud KMS Documentation | Google Cloud](#).

To add or update key labels, in the **LABELS** section:

1. Under **Label**, specify a tag key.
2. Enter the tag value.
3. Click the **+** button.

Similarly, you can add more tags. To remove a tag, click the close (X) icon in the tag name.

4. Click **Update**.

The key labels are updated.

Updating Key Rotation Schedule

To update a key rotation schedule, in the **ROTATION** section:

1. **Select Rotation Schedule.** The drop-down list shows the available key rotation schedules.

2. Select the **Key Origin**. The options are **CipherTrust (External)**, **CipherTrust (Local)**, **Native (Google)**, and **DSM**. Refer to [Key Creation Methods and Sources](#) for details on sources.
3. Select the **Algorithm** for the key.
4. (Applicable to the CipherTrust (External) and DSM) Select the desired **Domain**.
5. Click **Update**.

The key rotation schedule is updated.

Performing Key Version Operations

With CCKM, you enable/disable, add or rotate, and schedule destruction of key versions. Also, you can download public key of asymmetric versions of a key.

To perform a operation on a key version, under the Key Versions section:

1. Click the overflow icon () corresponding to the desired version. A popup menu appears.
2. Select the desired operation. Depending on the current state, the options can be Disable Version, Enable Version, and Schedule Destruction. An Aborted version cannot be enabled, disabled, or scheduled for destruction.

To download the public key of an asymmetric key version, click **Get Public Key**. The public key of (**public-key . pem**) the asymmetric key version is downloaded to your machine.

3. Confirm the operation.

Note

To rotate all versions of a key, click **Rotate** under the Key Versions section.

Managing Principals and Roles

Assigning Principals and Roles

To assign the principles and roles:

1. Expand **PERMISSIONS**.
2. Click **Assign Principal/Role**. The Assign Principal/Role wizard will open.
3. In the ADD PRINCIPAL section, select a principal **Type** from the drop-down list. The available types are **User**, **Service Account**, **Group**, and **Domain**.
4. **Add Principals**. You can add multiple principals separated by a comma.

To add additional principals, click **Add More Principals**.

5. In the SELECT ROLES section, select a **Role** from the drop-down list.
6. Click **Set Conditions**. The Add Conditions wizard is displayed.
 - a. Enter **Title**.
 - b. (Optional) Enter **Description**.
 - c. (Optional) Enter a **Condition** in **Basic View**.

Note

- Ensure the correctness of the logical expressions (conditions) you add. Neither CCKM nor GCP checks for the logical validity of the added conditions.
- Provide input in valid formats. CCKM performs input data validation similar to GCP.

- a. Select a **Condition Type**.
- b. Select an **Operator**.
- c. Select a **Detail** from the drop-down list for the selected Condition Type. This may be a resource type, resource service, value, or other detail.

To add additional conditions, select **Add Condition**.

To add a group condition, select **Add Group Condition**. A group condition contains more than one condition.

Note

- If you add multiple conditions and group conditions, you need to provide the relation between the conditions and relation between the conditions within the group condition. The relation can be **AND** or **OR**.
- You can also add conditions within the conditions and group conditions and group conditions within the conditions and group conditions.

- d. Click **Save**.

To add additional roles, click **Add More Roles**.

-
7. Click **Save**.

If the added principals already exist, the principal/role will be assigned and you will see a success message.

Editing Roles

To edit the roles:

1. Expand **PERMISSIONS**.
2. Expand the desired **Role**.
3. Under the expanded role, click the overflow icon () corresponding to the desired principal.
4. Click **Edit**. The Edit Roles wizard is displayed.
5. Update the existing Role.
6. Click **Set Conditions**. The Add Conditions wizard is displayed.
 - a. Enter **Title**.
 - b. (Optional) Enter **Description**.
 - c. (Optional) Enter a **Condition** in **Raw View**.

Note

- Ensure the correctness of the logical expressions (conditions) you add. Neither CCKM nor GCP checks for the logical validity of the added conditions.
- Provide input in valid formats. CCKM performs input data validation similar to GCP.

Enter the details in the text area.

- d. Click **Save**.

To add additional roles, click **Add More Roles**.

7. Click **Update**.

Removing Principals

To remove the principals:

1. Expand **PERMISSIONS**.
-

2. Expand the desired **Role**.
3. Under the expanded role, click the overflow icon () corresponding to the desired principal.
4. Click **Remove**. The Remove Principal dialog box is displayed.
5. Click **Remove**.

On successful deletion, a Google key updated successfully message is displayed on the screen.

Note

After all the principals are removed from a role, the role will be deleted automatically.

Re-Importing a Google Cloud Key Version

To re-import a key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**.
3. Click the expand icon () to the left of the desired key.
4. Click the overflow icon () corresponding to the desired version.
5. Click **Re-Import**. The Re-Import Key wizard is displayed.

Note

If you want to re-import the key version, ensure that the state of the version is **Import Failed** or **Destroyed** and the re-import eligibility is **True**. Also, even if these conditions are met, you cannot re-import all the key versions.

6. Under **Select Method**, select the desired method to re-import the key, the options are **Create/Upload New Key Material** and **Clone Existing Key Material**.
7. Click the desired tab to view the instructions.

Create/Upload New Key Material

- a. Select the **Source**. The options are **CipherTrust (Local)**, **Vormetric DSM**, and **Luna HSM**.

Note

For some key materials, the source might already be selected.

- b. Click **Next**. The **Configure Source Key** tab is displayed.

CipherTrust (Local) as a source

- a. Select the Key Type. You can select **Symmetric** or **Asymmetric**.
- b. Enter the **Key Name**.
- c. (Applicable to Asymmetric key type) Select the **Algorithm**. You can select **RSA** or **EC**.

Vormetric DSM as a source

- a. Select the Encryption Type. You can select **Symmetric** or **Asymmetric**.
- b. Enter the **DSM Key Name**.
- c. (Optional) Enter the **Description**.
- d. Select a DSM Domain from the list drop-down list.
- a. (Applicable to Asymmetric Encryption type) Select an **Algorithm**. You can select **RSA-2048**, **RSA-3072**, or **RSA-4096**.

Luna HSM as a source

- a. Enter **Luna HSM Key Label**.
- b. Select **Partition ID**.
- c. Select **Mechanism**.
- d. Select **Key Size**. The options are **2048**, **3072**, and **4096**.
- e. Select the **Key Attributes**.

Clone Existing Key Material

- a. Select the **Source**. The options are **CipherTrust (Local)**, **Vormetric DSM**, and **Luna HSM**.

Note

For some key materials, the source might already be selected.

- b. Click **Next**. The **Configure Source Key** tab is displayed.

Note

By default, the selected Algorithm and Key Size are AES and 256, respectively.

- a. Select a Key from the list.

8. Click **Re-Import**.

Google Cloud Reports

CCKM provides options to generate key visibility reports based on:

- Key-related activities between CCKM and Google Cloud key rings
- Track keys by their expiration dates

Google Cloud reports are categorized as:

- **Key Activity Report:** Inspect individual Google Cloud key histories by operations, for example, when they were refreshed, rotated, edited, or deleted. Also, use this report to compare key activities between CCKM and Google Cloud key ring.
- **Key Aging Report:** Track keys by their expiration dates. Audit a range of dates, from past material deletions to future scheduled deletions, within the selected Google Cloud key ring.

Use the Google Cloud reports to:

- Generate a report
- View the list of existing reports
- View details of a particular report
- Delete a report from CCKM
- Download the content of a report in CSV

Creating Google Cloud Reports

To create a Google Cloud report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Google**. The Google Reports page is displayed.
3. Click **Add Report**. The Choose Report Type and Name screen of the Add Google Report wizard is displayed.
4. Select a report type. The options are:
 - Key Activity Report
 - Key Aging Report
5. Specify a **Report Name**. This is a mandatory field.
6. Click **Next**. The Select Google Parameters screen is displayed.
7. Select a key ring from **Key Rings** drop-down list.
8. Click **Add**. The selected Google Key Ring is displayed under Selected Key Rings Parameters. Add more key rings, if required.
9. Click **Next**. The Set Start and End Dates screen is displayed.
10. In the **Include Entries From** field, specify the start date and time for the report.
 - Click the field and select the date and time on the on-screen calendar.
 - Alternatively, enter the time in **MM/DD/YYYYHH:MM** format.
11. In the **To** field, specify the end date and time for the report.
12. Click **Save**.

A success message **Created report** is displayed. The newly created report is displayed under the Google Cloud reports list. The report status becomes Running Now.

Viewing Google Cloud Reports

The Google Reports page displays the list of existing reports. Filter the reports by their names.

To view existing Google Cloud reports:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Google**. The list of available reports is displayed. The following details are displayed:

Column Name	Description
Report Name	Name of the report.
Run Date	When the report is run.
Type	Type of the report. The type can be: • Key Activity • Key Aging
Start Date	Start date from when the report is generated. If a report is run now, its status becomes Running Now.
End Date	End date for the report.

Viewing Details of a Google Cloud Report

To view the details of a Google Cloud report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Google**. The list of available reports is displayed.
3. Click the desired **Report Name** link.

Alternatively, click the overflow icon () corresponding to the desired report and click **View**.

4. View the report details.

The page is divided into:

- Mini detail view: Shows basic report details such as Cloud Service, Report Type, Start Date, End Date, and Overall Status.
- KEY RINGS: Expand to view the included key rings.
- REPORT DETAILS: Filter the report by Key Name, Key Ring, and Project. The following details are displayed:

Field	Description
Key Name	Name of the Google Cloud report.
Key Ring	Name of the linked Google Cloud key ring.
Project	Name of the Google Cloud project.
Location	Location of the Google Cloud project where the key ring resides.
Expiration Date	(Key Aging reports only) Date and time when the report expires.

Field	Description
Key Activity	(Key Activity reports only) Activity performed on the Google Cloud key.
Modified Date	(Key Activity reports only) Date and time when the key is modified.
User Name	(Key Activity reports only) User who performed the key activity.

Downloading Google Cloud Reports

To download a Google Cloud report as a CSV file:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Google**. The list of available reports is displayed.
3. Click the overflow icon () corresponding to the desired report and click **Download**.

Alternatively, click the desired report under **Report Name**, and click **Download CSV** in the detail view.

The report is downloaded as a CSV file.

Deleting Google Cloud Reports

To delete a Google Cloud report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Google**. The list of available reports is displayed.
3. Click the overflow icon () corresponding to the desired report and click **Delete**. The Confirm Delete Report message is displayed.
4. Click **Delete Report**.

The report is removed from the list of reports.

Scheduling Operations

CCKM allows you to periodically refresh keys, rotate keys, or add KMS containers in the background. To refresh keys, rotate keys, or add KMS containers, you need to create schedule configuration. A schedule configuration defines when to run a refresh, rotation, or KMS container job. You can use either the basic or cron format to specify when the rotation, refresh, or KMS container job will run.

Specify basic format in the following order:

[Repeats, at]

Where,

- **Repeats:** frequency of the scheduler. Possible values are daily, weekly, monthly, and yearly.
- **at:** specific time at which the rotation or refresh operation will be performed. Possible value lies between 12:00-11:00 AM/PM UTC.

Specify cron format in the following order:

" * * * * * "

These five values indicate when the job should be executed. These values are mandatory and must be specified in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

Field	Allowed values	Allowed special characters
Minute	0-59	* / , -
Hour	0-23	* / , -
Day of month	1-31	* / , - ?
Month	1-12 or JAN-DEC	* / , -
Day of week	0-6 or SUN-SAT	* / , - ?

Adding Key Rotation Schedule

A key rotation schedule replaces selected keys with new cryptographic keys at the specified time.

To add key rotation schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. Select **Schedule Type** as **Key Rotation**.
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **Google** from the **Cloud Name** drop-down list.
 - Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.

-
7. On the **Schedule Config** screen, enter or select the following details.
 - a. Configure **DURATION** for the scheduler. Specify the schedule start and end time:
 - **Schedule Starts**: Specify time when the schedule starts.
 - **Schedule Ends**: Unavailable by default, that is, the scheduler never expires.
 - **Never**: Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.

- b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
 - c. Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created schedule is displayed in the schedules list.

Adding Key Refresh Schedule

A key refresh schedule synchronizes keys from linked Google Cloud accounts to Cloud Key Manager.

To add a key refresh schedule:

1. Open the **Cloud Key Manager** application.
 2. In the left pane, click **Schedules**. The list of available schedules is displayed.
 3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
 4. **Select Schedule Type as Key Refresh.**
 5. Click **Next**.
 6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **Google** from the **Cloud Name** drop-down list.
 - Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
 7. On the **"Schedule Config** screen, enter or select the following details.
 - a. Configure **DURATION** for the scheduler. Specify the schedule start and end time:
-

- **Schedule Starts:** Specify time when the schedule starts.
- **Schedule Ends:** Unavailable by default, that is, the scheduler never expires.
- **Never:** Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.

- b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will refresh keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
- c. Click **Next**.

8. On the **Add Key Rings** screen:

- a. Select the desired key ring from the **Refresh Keys from Selected Key Ring(s)** drop-down list.
- b. Click **+** button. Similarly, you can add more key rings. To remove an added key ring, click the close (X) icon in the name of the added key ring.

Alternatively, if you want to refresh all the existing key rings, select the **Refresh All Key Rings** check box. When selected, the **Refresh Keys from Selected Key Ring(s)** field becomes unavailable.

- c. Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created scheduled is displayed in the schedules list.

Automatic Cloud Key Discovery

A KMS container schedule automatically detects the KMS containers based on the Google Cloud connection and adds them to the CipherTrust Cloud Key Manager.

Adding a KMS container (key ring or project) to CCKM consumes one CCKM license count.

To add a KMS container schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Select Schedule Type screen of the Add New Schedule wizard is displayed.

Schedule Type

1. Select **Schedule Type** as **Add KMS Container**.
2. Click **Next**. The General Info screen is displayed.

General Info

1. Enter a unique **Scheduler Name**.

2. Select **Google** from the **Cloud Name** drop-down list.
3. (Optional) Add **Description** for the scheduler in a maximum of 250 characters.
4. (Optional) Select **Enable Schedule** to enable the schedule.
5. Select a **Define Job Type**. You can select either **Discover Only** or **Discover and Add**.

Note

Discover Only lists all the discovered containers. Discover and Add adds all the discovered containers to the CCKM.

6. Click **Next**. The Connection & ACL Info screen is displayed.

Connection & ACL Info

1. Select a **Connection**.
2. (Optional) Enable or disable **Success Audit Event**.
3. In the ACCESS CONTROL section.
 - a. Click **Assign User/Group**. The Assign User/Group dialog box is displayed.
 - b. Select the desired user or group from the **User/Group** drop-down list.
 - c. Click **Save**. The newly added user/group is displayed under **Name** in the ACCESS CONTROL section.
 - d. Grant the permissions to the user/group, as appropriate.

Granting additional permissions

To grant the additional permissions, select the check box under the desired operation corresponding to the desired users or groups. The following permissions can be granted.

- View Keys, Add Key, Edit / Disable/ Enable, Synchronize, Upload
- Schedule Destroy, Cancel Destroy

To remove a permission, clear the check box under the desired operation.

- e. Remove a user or group.
 - a. Under **Unassign**, click **X** corresponding to the desired user/group. The Remove Group dialog box is displayed.
 - b. Click **Remove**.

- Click **Next**. The Schedule Config screen is displayed.

Schedule Config

- Configure **DURATION** for the scheduler. Specify the schedule start and end time:

- **Schedule Starts:** Specify time when the schedule starts.
- **Schedule Ends:** Unavailable by default, that is, the scheduler never expires.
- **Never:** Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.

- Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
- Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created schedule is displayed in the schedules list.

Viewing/Editing Schedules

To view/edit a schedule:

- Open the **Cloud Key Manager** application.
- In the left pane, click **Schedules**. The Schedules page displays following details:

Field	Description
Name	Unique name of the scheduler configuration.
Schedule Type	Type of the schedule. Possible types are: • Key Rotation • Key Refresh • Add KMS Container (GCP and Azure only)
Cloud Name	Name of the cloud.
Last Modified	Time when the schedule is modified.
Frequency	Frequency of the scheduler configuration.
Start Date	Creation time of the scheduler configuration.
End Date	Expiry time of the scheduler configuration. Never is displayed if the schedule is set to never expire.
Status	Status of the scheduler configuration. Possible values are:

Field	Description
	• Enabled • Disabled

- Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.

Alternatively, click the **Name** link corresponding to the desired schedule.

- Edit or configure the following fields:

- Optional, basic description of the schedule.
- Status of the scheduler configuration. Select **Enable Schedule** to enable the schedule, clear to disable.
- (For KMS Container schedule) Define Job Type.
- **CONNECTION & ACL INFO** details, refer to the add KMS container [Connection & ACL Info](#) section.
- Scheduler configuration parameters such as duration, frequency, and expiry time of the key. Refer to the key rotation [Schedule Config](#) screen, key refresh [Schedule Config](#) screen, or KMS container [Schedule Config](#) screen for details, as appropriate.

- Click **Update** to save the changes.

JOB HISTORY

The Schedules page also contains a section named JOB HISTORY. The JOB HISTORY section displays information related to a job such as Run Date, Job ID, Status, and Errors. Every time a schedule is run, a Job is created.

- For the KMS Container schedule, this section allows you to download the CSV report containing the list of Key Rings or Projects.

To download the report, click the overflow icon () corresponding to the desired schedule and click **Download Key Rings Only** or **Download Projects Only**.

- To view the details of the job, click the **Job ID** link corresponding to the desired job. The Job details page is displayed. It shows the list of Key Rings and Projects associated with the job along with their details.

To download the reports of Key Rings and Projects, go to the top left of the Job details page, click **Download**, and select a **report** from the drop-down menu. The options are **Key Rings Only**, **Projects Only**, and **Key Rings and Projects**.

Note

You can download the report after the schedule has run successfully.

Disabling Schedules

To disable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Disable**. The Disable Schedule message is displayed.
4. Click **Disable Schedule**.

A message **Successfully disabled the schedule** is displayed on the screen.

Enabling Schedules

To enable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Enable**. The Enable Schedule message is displayed.
4. Click **Enable Schedule**.

A message **Successfully enabled the schedule** is displayed on the screen.

Manually Running Schedules

To manually run a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Run Now**.

The Run Now schedule is started in the background. A message **Schedule is running now. It will take a few seconds to finish.** is displayed on the screen. After the schedule is run successfully, a message **Successfully finished running the schedule** is displayed.

Deleting Schedules

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.

-
3. Click the overflow icon () corresponding to the desired schedule and click **Delete Schedule**. The Delete Schedule message is displayed.
 4. Click **Delete**.

A message **Successfully deleted schedule** is displayed on the screen.

Downloading Google Metadata

Downloading Google Keys Metadata

CCKM users with permissions to view the keys can export and download the Google keys metadata displayed on the CCKM GUI. Filter the keys metadata to display on the GUI based on your requirements, for example, all keys created in the past 12 months, all keys residing in a specific Google key ring, and all keys based on a specific algorithm. You can then download the filtered metadata as a CSV file.

To download the Google keys metadata visible:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Google**. The Key Rings tab of the Google page is displayed.
3. Filter the keys metadata based on your requirements.

Use the Search box to filter the keys. To further filter the metadata, show/hide the desired columns and use the column-level filters.

4. Click **Download as CSV**.

The filtered Google keys metadata is downloaded as a CSV file.

Downloading Google Key Rings Metadata

CCKM users with permissions to view the Google key rings can export and download the key rings metadata displayed on the CCKM GUI. Filter the key rings metadata to display on the GUI based on your requirements. You can then download the filtered metadata as a CSV file.

Metadata of a maximum of 100 key rings can be exported and downloaded.

To download the Google key rings metadata visible:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Google**. The Key Rings tab of the Google page is displayed.
3. Filter the key rings metadata based on your requirements.

Use the Search box to filter the key rings. To further filter the metadata, show/hide the desired columns and use the column-level filters.

4. Click **Download as CSV**.

The filtered Google key rings metadata is downloaded as a CSV file.

Performance Summary

This section provides metrics and results of GCP performance testing.

Test Setup

Test Runner	
Operating System	Ubuntu 18.4
RAM	8 GB
Core	4

Appliance	
Build	2.11.0.8177
RAM	16 GB

Process

Scripts are written on top of the Locust Framework for load and stress testing. The testing is done for create, read, update, and delete (CRUD) key operations with respect to different numbers of virtual users (VUS) over parallel connections, one for every virtual user.

CRUD Operations

Expand the headings below to view performance numbers.

Response with request coverage in percentile (10 VUS, 180 seconds)

POST

Click the tabs to view performance numbers.

Table 1

Method	Name	# Request s	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/ s
POST	/cckm/ google/ upload- key	207	0	8502	1306	14281	630	1.2	0.0
Aggregated	Aggregated	207	0	8502	1306	14281	630	1.2	0.0

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/cckm/google/upload-key	7200	8700	10000	12000	13000	13000	14000	14000
Aggregated	Aggregated	7200	8700	10000	12000	13000	13000	14000	14000

GET

Click the tabs to view performance numbers.

Table 1

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/cckm/google/keys/[id]	285	0	6097	4156	11275	1034	1.6	0.0
Aggregated	Aggregated	285	0	6097	4156	11275	1034	1.6	0.0

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/cckm/google/keys/[id]	5000	5900	7200	8500	9400	9900	11000	11000
Aggregated	Aggregated	5000	5900	7200	8500	9400	9900	11000	11000

PATCH

Click the tabs to view performance numbers.

Table 1

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
PATCH	/cckm/google/keys/[id]	246	0	7035	4482	11785	969	1.4	0.0
Aggregated	Aggregated	246	0	7035	4482	11785	969	1.4	0.0

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
PATCH	/cckm/google/keys/[id]	5900	6600	8100	9500	11000	11000	12000	12000
Aggregated	Aggregated	5900	6600	8100	9500	11000	11000	12000	12000

DELETE (List and Destroy)

Click the tabs to view performance numbers.

Table 1

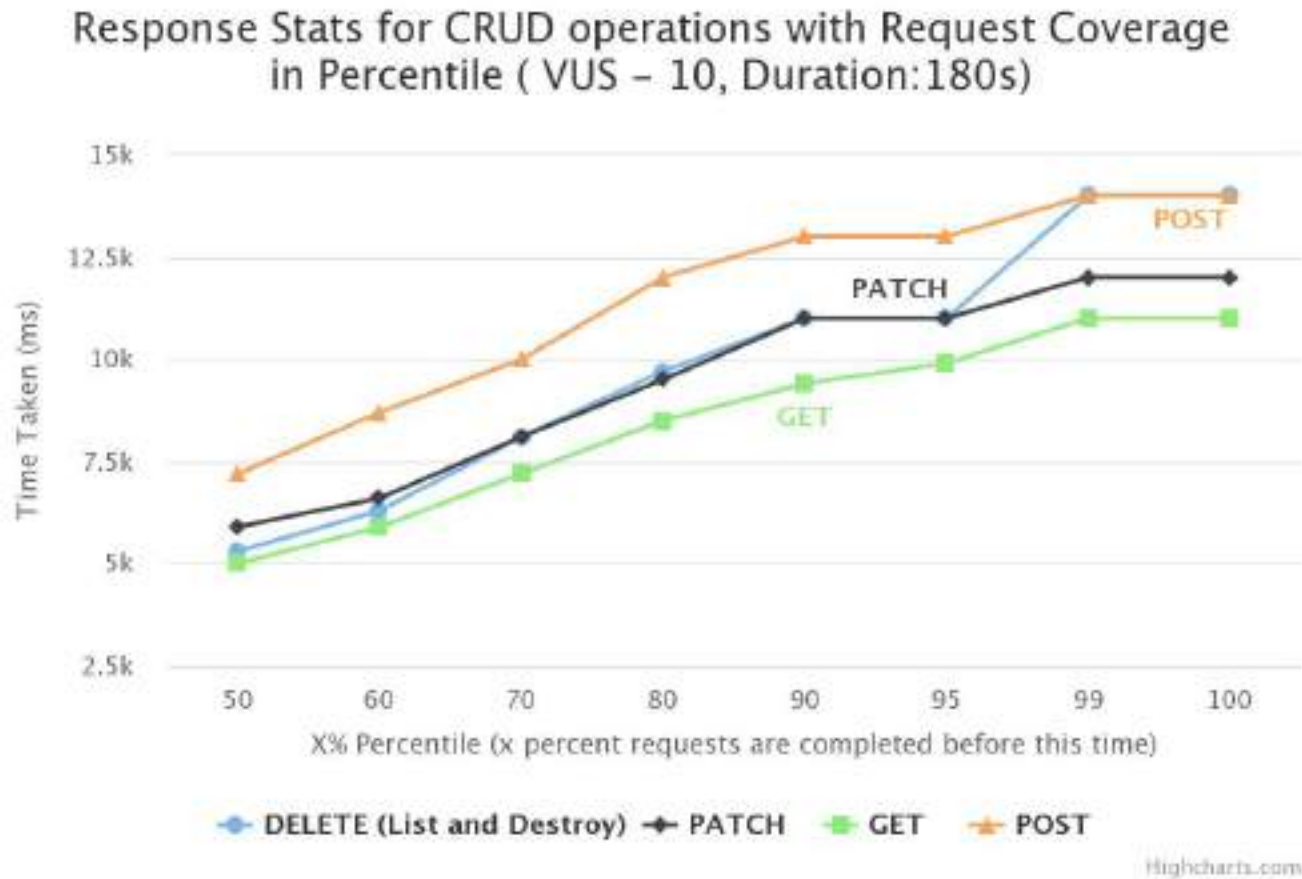
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/cckm/google/keys/[id]/versions	225	0	6728	4080	14456	1481	1.3	0.0
POST	/cckm/google/keys/[id]/versions/[versionID]/schedule-destroy	223	200	1123	978	1601	724	1.2	1.1
Aggregated	Aggregated	448	200	3938	978	14456	1104	2.5	1.1

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/cckm/google/keys/[id]/versions	5300	6300	8100	9700	11000	11000	14000	14000
POST	/cckm/google/keys/[id]/versions/[versionID]/schedule-destroy	1100	1100	1200	1200	1300	1400	1600	1600
Aggregated	Aggregated	4100	4500	4900	6300	9700	11000	12000	14000

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
ed	ed								

Here, X% percentile indicates that X percent requests are completed before this time.



In the graph above:

- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
- Y-axis represents the time taken by a virtual user to perform any CRUD operation.
- All the virtual users are spawned parallelly at the same time to perform each of these CRUD operations for the given duration.

Response with request coverage in percentile (30 VUS, 180 seconds)

POST

Click the tabs to view performance numbers.

Table 1

Method	Name	# Request s	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/ s
POST	/cckm/ google/ upload- key	207	0	23659	2793	29505	630	1.2	0.0
Aggregated	Aggregated	207	0	23659	2793	29505	630	1.2	0.0

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/cckm/ google/ upload- key	25000	27000	28000	29000	29000	29000	29000	30000
Aggregated	Aggregated	25000	27000	28000	29000	29000	29000	29000	30000

GET

Click the tabs to view performance numbers.

Table 1

Method	Name	# Request s	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/ s
GET	/cckm/ google/ keys/[id]	270	0	18602	12907	35524	1034	1.5	0.0
Aggregated	Aggregated	270	0	18602	12907	35524	1034	1.5	0.0

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/cckm/ google/ keys/[id]	19000	20000	22000	23000	23000	25000	34000	36000
Aggregated	Aggregated	19000	20000	22000	23000	23000	25000	34000	36000

PATCH

Click the tabs to view performance numbers.

Table 1

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
PATCH	/cckm/google/keys/[id]	221	0	22675	15652	37918	969	1.2	0.0
Aggregated	Aggregated	221	0	22675	15652	37918	969	1.2	0.0

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
PATCH	/cckm/google/keys/[id]	23000	24000	25000	25000	27000	29000	36000	38000
Aggregated	Aggregated	23000	24000	25000	25000	27000	29000	36000	38000

DELETE (List and Destroy)

Click the tabs to view performance numbers.

Table 1

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/cckm/google/keys/[id]/versions	202	0	24001	16704	39826	1494	1.1	0.0
POST	/cckm/google/keys/[id]/versions/[versionID]/schedule-destroy	201	194	450	300	810	669	1.1	1.1
Aggregated	Aggregated	403	194	12255	300	39826	1082	2.2	1.1

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/cckm/	24000	26000	26000	27000	29000	35000	39000	40000

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
	google/ keys/[id]/ versions								
POST	/cckm/ google/ keys/[id]/ versions/ [versionI D]/ schedule -destroy	390	420	520	550	580	620	740	810
Aggregat ed	Aggregat ed	17000	19000	22000	26000	27000	29000	38000	40000

Here, X% percentile indicates that X percent requests are completed before this time.

Response Stats for CRUD operations with Request Coverage in Percentile (VUS – 30, Duration:180s)



In the graph above:

- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
- Y-axis represents the time taken by a virtual user to perform any CRUD operation.

- All the virtual users are spawned parallelly at the same time to perform each of these CRUD operations for the given duration.

Response with request coverage in percentile (50 VUS, 180 seconds)

POST

Click the tabs to view performance numbers.

Table 1

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/cckm/google/upload-key	203	0	37889	9291	47310	630	1.1	0.0
Aggregated	Aggregated	203	0	37889	9291	47310	630	1.1	0.0

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/cckm/google/upload-key	42000	42000	43000	44000	45000	46000	46000	47000
Aggregated	Aggregated	42000	42000	43000	44000	45000	46000	46000	47000

GET

Click the tabs to view performance numbers.

Table 1

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/cckm/google/keys/[id]	239	0	33494	25306	61685	1034	1.3	0.0
Aggregated	Aggregated	239	0	33494	25306	61685	1034	1.3	0.0

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/cckm/google/keys/[id]	32000	32000	32000	35000	45000	55000	61000	62000
Aggregated	Aggregated	32000	32000	32000	35000	45000	55000	61000	62000

PATCH

Click the tabs to view performance numbers.

Table 1

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
PATCH	/cckm/google/keys/[id]	202	0	38554	33081	65534	969	1.1	0.0
Aggregated	Aggregated	202	0	38554	33081	65534	969	1.1	0.0

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
PATCH	/cckm/google/keys/[id]	36000	37000	37000	39000	47000	54000	63000	66000
Aggregated	Aggregated	36000	37000	37000	39000	47000	54000	63000	66000

DELETE (List and Destroy)

Click the tabs to view performance numbers.

Table 1

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/cckm/google/keys/[id]/versions	197	0	39153	30230	61986	1494	1.1	0.0
POST	/cckm/google/keys/[id]/versions/	196	192	1007	355	1748	657	1.1	1.1

Method	Name	# Request s	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/ s
	[versionID]/ schedule-destroy								
Aggregated	Aggregated	393	192	20128	355	61986	1077	2.2	1.1

Table 2

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/cckm/ google/ keys/[id]/ versions	37000	37000	38000	39000	48000	53000	61000	62000
POST	/cckm/ google/ keys/[id]/ versions/ [versionID]/ schedule-destroy	1000	1100	1100	1200	1300	1500	1600	1700
Aggregated	Aggregated	30000	37000	37000	37000	39000	48000	59000	62000

Here, X% percentile indicates that X percent requests are completed before this time.

Response Stats for CRUD operations with Request Coverage in Percentile (VUS – 50, Duration:180s)



In the graph above:

- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
- Y-axis represents the time taken by a virtual user to perform any CRUD operation.
- All the virtual users are spawned parallelly at the same time to perform each of these CRUD operations for the given duration.

Note

The delete operation includes listing key version(s) and schedule destroy. When multiple virtual users are executing the delete operation, more than 90% **Bad request** failures are observed with the status code 400. As the operation is synchronous, the behavior might be due a bottleneck on the GCP side when the requests are sent.

Google Cloud External Key Manager Resources

Google Cloud External Key Manager (EKM) is a cloud native service that provides access to an external key encryption key (KEK) for use as a wrapping key in Google Cloud Platform (GCP) Projects. CCKM integration with

Google Cloud EKM enables you to:

- **Manage endpoints** for KEKs for keys added to the key ring through GCP EKM
- **Configure and manage policy sets**

The AES256 wrap/unwrap KEK allows users, developers, and organizations to maintain separation between encrypted data at rest and encryption keys.

The benefits of using CCKM Google Cloud EKM Endpoints include:

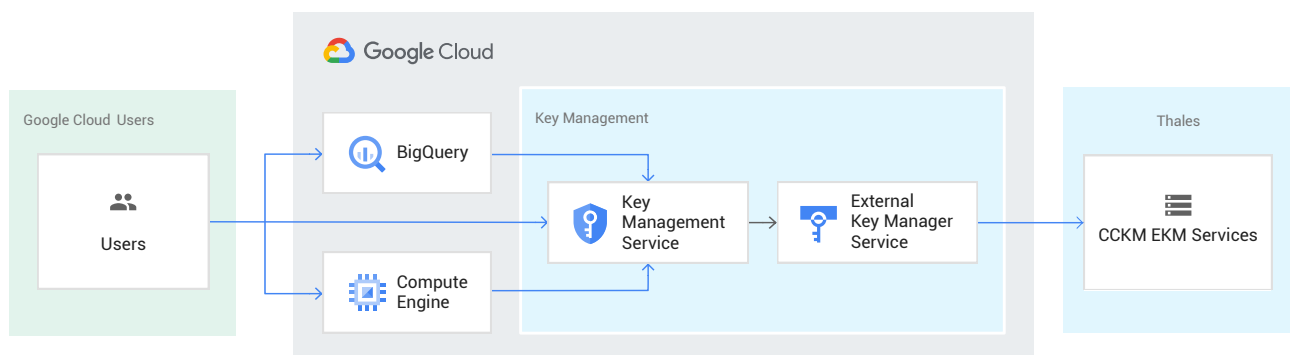
- Secure generation, storage and protection of your KEK.
- Privately maintained key provenance, managed access control, and centralized key management.
- Full life cycle management of your encryption key.
- Visibility for compliance.

GCP allows users to use Cloud External Key Management (EKM) in the Google Cloud Key Management Service (KMS) for Google Projects. CCKM protects your data in the GCP while your **encryption keys are stored in CipherTrust Manager** outside of GCP. Users create a Key Encryption Key (KEK) in CCKM, create a Cloud EKM key in Google Cloud, using the KEK's URI to identify the externally-managed key in Google Cloud KMS, and use the keys to protect data in a Customer-Managed Encryption Key (CMEK) integration service, to encrypt data using a symmetric key, or to sign with an asymmetric key. In this scenario, Google Cloud KMS does not store the external key material.

Note

CCKM doesn't support Google BYOK for Google Trusted Partner Clouds (TPC).

The following diagram shows how the Cloud KMS and CCKM fit into the key management model, using BigQuery and Compute Engine as example services.



Note

If you are [deploying a new CipherTrust Manager instance](#) exclusively or primarily to use the Google Cloud EKM service, we recommend deploying the instance geographically close to one of the [Google Cloud KMS regions](#) where you intend to set up the Google Cloud KMS Key Ring.

We have tested the following Google Customer-Managed Encryption Key (CMEK) integration services for Google Cloud EKM:

- BigQuery
- Compute Engine

All other Google CMEK integration services for Google Cloud EKM are not validated by Thales, but are expected to work and are supported. Consult [Google EKM documentation](#) for the full list of Google CMEK services for EKM. Only CMEK services integrated with Google Cloud EKM are supported with CCKM EKM endpoints.

These are "Hold Your Own Key" (HYOK) integrations, where you manage and control the base KEK inside of CCKM. Google Cloud has additional CMEK services that do not follow the HYOK model and do not integrate with EKM.

The connection between Google Cloud and CCKM can take place over the internet, or mediated through a [Google Cloud Virtual Private Cloud \(VPC\) network](#).

Ubiquitous Data Encryption

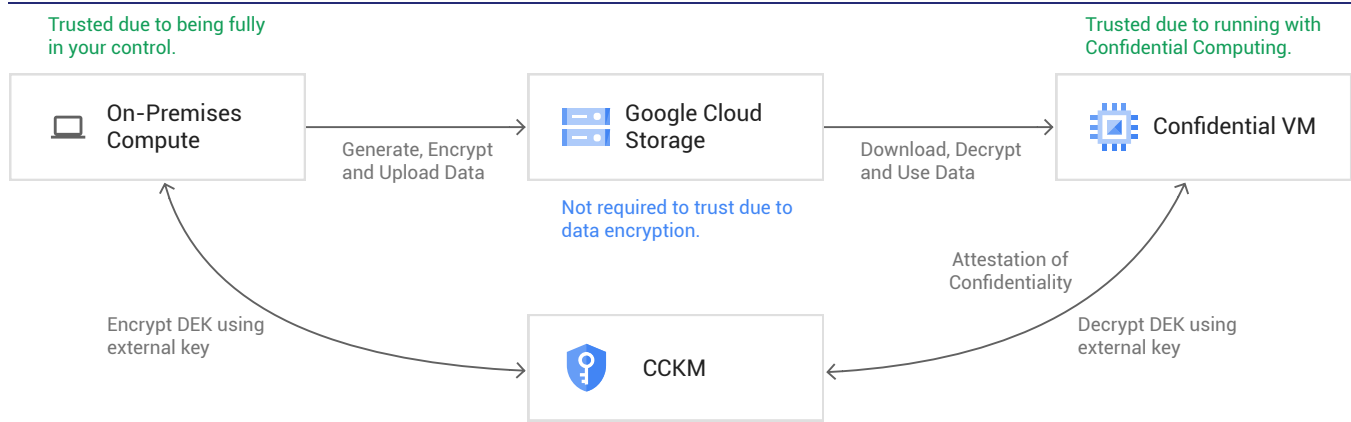
CCKM provides another integration with EKM, called Google Cloud External Key Manager Ubiquitous Data Encryption (EKM UDE). While standard EKM protects data in use for CMEK services in Google Cloud, EKM UDE protects data as it moves between two environments, called workloads, mediated by Google Cloud KMS. The two workloads can be two Google Cloud Confidential VMs, two non-confidential environments (on-premises or cloud), or one Confidential VM and one non-confidential environment.

CCKM provides options in the EKM UI menus to [manage endpoints](#), [configure and manage policy sets](#), and configure confidential computing requirements for access to KEKs.

The UDE version of EKM provides additional security, access control and auditability guarantees, namely:

- the end-to-end encryption of Data Encryption Keys (DEKs) between workloads and the external key manager
- the leveraging of Confidential VMs to provide strong guarantees of the runtime privacy of customer data (data-in-use security)
- the use of independently-verifiable attestations of the runtime environment, allowing the EKM to strongly differentiate between protected and unprotected environments

An example architecture is shown below, showing a potential interaction between CCKM, Google Cloud Storage, and a Confidential VM. For any type of workload, CCKM holds the KEK needed to wrap and unwrap DEKs. Communications between CCKM and a Confidential VM require an Attestation of Confidentiality sequence for an additional guarantee that only the intended workload can access the KEK.



These CipherTrust Cloud Key Manager keys can be used in four main use cases within GCP:

1. A DEK is generated within a GCP confidential VM, then is wrapped by the CCKM KEK. The KEK is configured such that unwrapping of the wrapped key is only possible by an attested, verified confidential VM. You can place additional restrictions on instance ID, project ID, and zones, which limit KEK use to *specific* confidential VMs.
2. A DEK is generated on-premise, in a regular (non-confidential computing) environment, then is wrapped by the CCKM KEK. The data is uploaded to Google Cloud Storage (GCS) and the KEK is configured such that unwrapping of the wrapped key (and hence the protected data) is only possible by an attested, verified confidential VM. You can place additional restrictions on instance ID, project ID, and zones, which limit KEK use to *specific* confidential VMs.
3. A DEK is generated within a GCP confidential VM, then is wrapped by the CCKM KEK. The KEK is configured such that wrapping of the wrapped key is only possible in an attested, verified confidential VM, but that unwrapping is possible in a regular (non-confidential computing) environment. You can place additional restrictions on instance ID, project ID, and zones, which limit KEK use to *specific* confidential VMs.
4. A DEK is generated on-premise in a regular environment, then is wrapped by CipherTrust-managed KEK. The data is moved to another regular environment (on cloud or on-premise). The KEK is configured such that unwrapping of the wrapped data is possible in a second regular environment.

These four cases, respectively, give the following guarantees:

1. In case 1, the guarantee that the protected DEK/data is only accessible by a confidential VM.
2. In case 2, the guarantee that data encrypted on-premise and migrated to the cloud will only be accessible by a confidential VM.
3. In case 3, the guarantee that data retrieved from the cloud and decrypted, was originated in a confidential VM.
4. In case 4, the guarantee is that the data is only decryptable when the KEK is accessible.

Related Pages

- [CCKM Overview](#)

-
- [Google Cloud External Key Manager Guide](#)
 - [Protecting BigQuery data with Cloud KMS keys](#)
 - [Protecting Compute Engine resources with Cloud KMS keys](#)
 - [Facilitate ubiquitous data encryption with the Split-Trust Encryption Tool](#)

Managing Google EKM Endpoints

Google Cloud External Key Manager (EKM) Service can access CipherTrust Cloud Key Manager (CCKM) after you meet some prerequisites.

Once this access is established, users in the '[CCKM Admins](#)' group and '[Key Users](#)' group can create and manage an endpoint in CCKM for Google Cloud EKM service to access a Key Encryption Key (KEK) using CCKM's GUI, CLI and REST API.

You can create a regular [EKM Endpoint](#) or an [External Key Manager Ubiquitous Data Encryption \(EKM UDE\) Endpoint](#). The architecture and use cases for the two endpoint types are described on the [Google Cloud External Key Manager Resources](#) page.

After you have created an endpoint, you can:

- [Edit the policies](#)
- [Rotate the KEK](#)
- [Change the base hostname](#)
- [Enable or disable](#) the wrapping and unwrapping operation. This allows you to avoid deleting the endpoint, and temporarily suspend Google Cloud EKM's ability to use the KEK.
- [Delete the endpoint](#)
- [View activity for a wrap or unwrap endpoint](#). Google Cloud Key Management Service (KMS) consumes these endpoints.
- [View attestation activity for an EKM UDE endpoint](#)

Prerequisites

To allow a connection between CipherTrust Manager and Google Cloud External Key Manager Service, some network and security configuration must be in place in both entities.

Google Cloud Platform Prerequisites

- A Google Cloud project using Cloud KMS must exist for the CCKM EKM integration's use. You can create a new project or choose an existing project in your Google Cloud account. You need to provide either a project ID or the private key file associated with the service account to add a Google project to CipherTrust Manager. The project ID is the simplest path to add the Google Cloud project just for EKM usage. The private key file is

required if you want add a Google Cloud connection to CipherTrust Manager.

- If you are using confidential computing environments for EKM UDE, an additional Google Cloud project needs to be associated with the confidential computing environments. A Google service account with the Identity and Access Management (IAM) permission `compute.instances.getShieldedInstanceIdentity` must have access to the project. This permission is required to carry out full verification of attestation evidence.
- If you wish to require [Key Access Justifications \(KAJ\)](#) you need to configure KAJ on Google Cloud.

CipherTrust Manager Prerequisites

- The CipherTrust Manager must have a public IP address with the web interface HTTPS port (default of 443) open. See [Planning Network Configuration](#) for details.
- The web interface port within CipherTrust Manager can be changed from the default port of 443 to another port. If you plan to change the default port for the CipherTrust Manager web interface, ensure to change it BEFORE configuring the Google EKM service on CCKM. Also reflect this port change when creating an EKM endpoint on CCKM as part of the prerequisites for deploying EKM. **Changing the default port AFTER configuring Google EKM is not supported.**
- The CipherTrust Manager must be reachable through a Fully Qualified Domain Name (FQDN).
- The web interface must have a [TLS certificate signed by an external Certificate Authority \(CA\) trusted by Google Cloud Platform](#). Google Cloud trusts certificates issued by well-known public CAs such as Verisign. Alternatively, you can create a certificate chain with Google's [Certificate Authority Service](#) and upload the chain to CipherTrust Manager.

For correct operation of UDE in a cluster of CipherTrust Manager nodes, all of the nodes must be configured with the same TLS certificate. This requirement is a consequence of termination of TLS sessions for UDE occurring at the individual CipherTrust Manager nodes rather than at the cluster load balancer.

- A Google Cloud project using Cloud KMS must be added to the CipherTrust Cloud Key Manager before creating an endpoint.
- For EKM UDE endpoints using confidential computing environments, the Google Cloud project associated with the confidential computing environments must be added to CCKM through [connection manager](#) for CCKM to completely verify attestation evidence.

Prepare an EKM endpoint

Below, we provide steps to create a Google endpoint in the CCKM GUI and to make the endpoint available to Google Cloud EKM.

1. Add the Google Cloud project using Google Cloud KMS to CCKM.

This action consumes one [CCKM license cloud unit](#).

Note

These steps demonstrate the simplest configuration to add a Google project ID to CCKM just for EKM endpoints, without using an active connection to Google Cloud. If you want to monitor and manage a Google Cloud account connection and associated resources on CipherTrust Manager, you can also [add a Google Cloud connection using connection manager](#), and then use the connection to retrieve project IDs.

- a. On Google Cloud, create a new project or choose an existing one.
 - b. Copy the project ID.
 - c. In CipherTrust Manager, open the **Cloud Key Manager** application.
 - d. Navigate to **KMS Containers > Google** and open the **Projects** tab.
 - e. Click **Add Existing Project**
 - f. In the popup window, under **Select Method**, switch to **Manually Enter Project ID**.
 - g. In **Project ID**, paste in the project ID.
 - h. (Optional) Use the **Enable success audit events** toggle to enable or disable audit recording of successful operations within the given Google cloud project. This toggle is set to enable, by default.
 - i. Click **Add Project**.
2. Login as a user in the 'CCKM Admins' group to create the endpoint in the CCKM GUI.

Note

You can also create the endpoint with the REST API or CLI to to associate meta information with the endpoint. Use the `/v1/cckm/ekm/endpoints` endpoint in the [REST API](#), or `ksctl cckm ekm endpoints create --ekm-endpoint-create-jsonfile <meta_information_filename>` in the CLI.

3. Navigate to **Cloud Key Manager > Services > Google Cloud EKM Endpoints**.
4. Click **Create Endpoint**
5. Provide a **Name** and a **Key URI Hostname** for your endpoint. For the Key URI Hostname, enter the the FQDN of the CipherTrust Manager instance.

Note

The Key URI Hostname will be used to construct the URL Google will use to send requests to CipherTrust Manager and for JWT validations.

The web interface of the CipherTrust Manager must have a valid TLS certificate signed by a trusted CA.

If the SSL termination takes place at the CipherTrust Manager, the Key URI Hostname must match the FQDN of the CipherTrust Manager. If the SSL termination takes place at the load balancer in front of the CipherTrust Manager, the Key URI Hostname must match the FQDN of the public-facing load balancer. Wildcard certificates are supported.

Note

If you changed the web interface port within CipherTrust Manager from the default port of 443 to another port, ensure to add that same port number when configuring Key URI Hostname by specifying the new port number along with the Key URI Hostname. For example, example.com:8443.

6. Select **EKM endpoint** as the Endpoint type.

Note

Ubiquitous Data Encryption endpoint steps are described [below](#).

7. Select either **Symmetric** or **Asymmetric** for **Key Type** of the KEK. If selecting **Symmetric**, click **Next** thereafter. If selecting **Asymmetric**, select key algorithm from the **Key Algorithm** drop down and click **Next**.

Endpoints with symmetric keys can be used for wrapping and unwrapping operations, whereas endpoints with asymmetric keys can be used for signing and fetching the public key.

Symmetric key type is always an AES-256 symmetric key. Symmetric keys for EKM endpoints must have the Encrypt, Decrypt, Wrap, and Unwrap usage masks. In addition, this key type must not be exportable or deletable.

Asymmetric key type has several RSA and EC algorithm options. Asymmetric keys for EKM endpoints must have the Sign and Verify usage masks. In addition, this key type must not be exportable or deletable. The available key algorithms for asymmetric keys are:

- RSA_SIGN_PSS_2048_SHA256

- RSA_SIGN_PSS_3072_SHA256
- RSA_SIGN_PSS_4096_SHA256
- RSA_SIGN_PSS_4096_SHA512
- RSA_SIGN_PKCS1_2048_SHA256
- RSA_SIGN_PKCS1_3072_SHA256
- RSA_SIGN_PKCS1_4096_SHA256
- RSA_SIGN_PKCS1_4096_SHA512
- EC_SIGN_P256_SHA256
- EC_SIGN_P384_SHA384

8. Choose to either create a new key or copy an existing key.

If creating a new key, select **Create New Key**, enter the name of the key, and click **Next**. If copying an existing key, select **Copy Existing Key**, select a key from the **Key** drop down, and click **Next**.

Caution

Although it is possible to copy a key that is associated with an existing EKM or EKM UDE endpoint, we recommend that you do not do this.

Note

The **Copy Existing Key** option is applicable to the migration of an EKM endpoint from one CipherTrust Manager deployment to another. When creating a new EKM endpoint in the new deployment using this option, CCKM uses the key name you select from the **Key** drop down and associates all of the existing versions of a CipherTrust Manager key to the new endpoint.

9. Edit the Endpoint Policy.

- a. Provide a comma-separated list of Google service accounts in the format **service-PROJECT_NUMBER@gcp-sa-ekms.iam.gserviceaccount.com** that have access to the endpoints specified in the **Clients** field.
- b. Edit the Key Access Justification settings. You can disable the requirement for Key Access Justifications, or specify which justification reasons are needed for the EKM endpoint to initiate a wrap or unwrap

request.

- c. Click **Next**.

Note

You can edit these [endpoint policies](#) later, if desired.

10. Review your settings and click **Add Endpoint**.

The EKM endpoint is created.

Caution

Modify endpoints only through the CipherTrust Cloud Key Management (CCKM) GUI, API, or CLI. Do not modify the KEK using the **Keys** menu, the `ksctl keys` command, or the `/v1/vault/keys2` endpoint. Such modifications might make the KEK unavailable to the Google Cloud EKM service.

11. Enable [Key Access Justifications](#) in your Google Account, if you require this feature, and you have not already enabled this feature.
12. Proceed to [establish a connection between Google Cloud and the endpoint](#).

Prepare an EKM UDE endpoint

1. Add the Google Cloud project using Google Cloud KMS to CCKM.

This action consumes one [CCKM license cloud unit](#).

Note

These steps demonstrate the simplest configuration to add a Google project ID to CCKM just for EKM endpoints, without using an active connection to Google Cloud. If you want to monitor and manage a Google Cloud account connection and associated resources on CipherTrust Manager, you can also [add a Google Cloud connection using connection manager](#), and then use the connection to retrieve project IDs.

- a. On Google Cloud, create a new project or choose an existing one.

- b. Copy the project ID.
 - c. In CipherTrust Manager, open the **Cloud Key Manager** application.
 - d. Navigate to **KMS Containers > Google** and open the **Projects** tab.
 - e. Click **Add Existing Project**
 - f. In the popup window, under **Select Method**, switch to **Manually Enter Project ID**.
 - g. In **Project ID**, paste in the project ID.
 - h. (Optional) Use the **Enable success audit events** toggle to enable or disable audit recording of successful operations within the given Google cloud project. This toggle is set to enable, by default.
 - i. Click **Add Project**.
2. Login as a user in the 'CCKM Admins' group to create the endpoint in the CCKM GUI.

Note

You can also create the endpoint with the REST API or CLI to to associate meta information with the endpoint. Use the `/v1/cckm/ekm/endpoints` endpoint in the [REST API](#), or `ksctl cckm ekm endpoints create --ekm-endpoint-create-jsonfile <meta_information_filename>` in the CLI.

3. Navigate to **Cloud Key Manager > Services > Google Cloud EKM Endpoints**.
4. Click **Create Endpoint**
5. Provide a **Name** and a **Key URI Hostname** for your endpoint. For the Key URI Hostname, enter the the FQDN of the CipherTrust Manager instance.

Note

The Key URI Hostname will be used to construct the URL Google will use to send requests to CipherTrust Manager and for JWT validations.

The web interface of the CipherTrust Manager must have a valid TLS certificate signed by a trusted CA.

If the SSL termination takes place at the CipherTrust Manager, the Key URI Hostname must match the FQDN of the CipherTrust Manager. If the SSL termination takes place at the load balancer in front of the CipherTrust Manager, the Key URI Hostname must match the FQDN of the public-facing load

balancer. Wildcard certificates are supported.

Note

If you changed the web interface port within CipherTrust Manager from the default port of 443 to another port, ensure to add that same port number when configuring Key URI Hostname by specifying the new port number along with the Key URI Hostname. For example, example.com:8443.

6. Select **Ubiquitous Data Encryption Endpoint** as the Endpoint type. Click **Next**.
7. Select the requirement for Confidential VMs to originate wrap or unwrap requests. You can select **Not Required**, required **For Wrap and Unwrap**, required **For Wrap** only, or required **For Unwrap** only.
8. Choose to either create a new key or copy an existing key.

If creating a new key, select **Create New Key**, enter the name of the key, and click **Next**. If copying an existing key, select **Copy Existing Key**, select a key from the **Key** drop down, and click **Next**.

Caution

Although it is possible to copy a key that is associated with an existing EKM or EKM UDE endpoint, we recommend that you do not do this.

Note

The **Copy Existing Key** option is applicable to the migration of an EKM endpoint from one CipherTrust Manager deployment to another. When creating a new EKM endpoint in the new deployment using this option, CCKM uses the key name you select from the **Key** drop down and associates all of the existing versions of a CipherTrust Manager key to the new endpoint.

The supported key type is AES-256. Valid keys must have the Encrypt, Decrypt, Wrap, and Unwrap usage masks and cannot be exportable or deletable.

9. Specify the endpoint policy.
 - a. Provide a comma-separated list of Google service accounts in the format **service-**

`PROJECT_NUMBER@gcp-sa-ekms.iam.gserviceaccount.com` that have access to the endpoints specified in the `Clients` field.

- b. Edit the Key Access Justification settings. You can disable the requirement for Key Access Justifications, or specify which justification reasons are needed for the EKM endpoint to initiate a wrap or unwrap request. Click **Next**.

Note

You can edit these [endpoint policies](#) later, if desired.

- c. Specify policies for **UDE Attestation**. The values for **Zones**, **Project IDs** and **Instance Names** constrain which workloads can use the EKM UDE Endpoint, and are enforced whenever a Confidential VM originates a request.

- d. Click **Next**.

10. Review your settings and click **Add Endpoint**. Click **Close** to close.

A unique URI is created, associated with the KEK that acts as the Google EKM UDE endpoint key. The hostname is applied to the URI, to create a path that Google Cloud can access.

Caution

Do not edit the KEK through the general CipherTrust Manager key management functions. Do not modify the KEK through the **Keys** menu in the GUI, **ksctl keys** commands in the CLI, or the `/v1/vault/keys2` endpoint in the REST API. This can result in the KEK becoming unavailable to the Google Cloud EKM service unexpectedly.

11. If you intend to use one or more confidential computing environments, you must add an additional Google Cloud project associated with the confidential computing VMs. [Add a Google Cloud connection on CipherTrust Manager through Connection Manager](#). You need to provide the private key file associated with the service account.

Note

For this case, the Google Cloud project needs to be associated with a Google service account with the Identity and Access Management (IAM) permission `compute.instances.getShieldedInstanceIdentity`. This ensures CCKM can fully verify

attestation evidence.

12. Enable [Key Access Justifications](#) in your Google Account, if you require this feature, and you have not already.
13. Proceed to [establish a connection between Google Cloud and the endpoint](#).

Establish a Connection Between Google Cloud and the Endpoint

Google Cloud configuration to connect to the endpoint depends on whether the connection is over the internet or Google Cloud VPC.

EKM over internet

If you want to set up EKM communication over the internet, follow [Using Cloud EKM via the Internet Google documentation](#) to create the necessary Google Cloud EKM key. You need the following values related to CipherTrust Manager:

- The full Key URI, including hostname, of the EKM endpoint.
- The geographical region where the CipherTrust Manager instance is deployed. In Google Cloud, choose a location for the Key Ring that is as close as possible to reduce network latencies.

EKM over VPC

If you want to set up EKM communication over VPC, follow [Using Cloud EKM with VPC Google documentation](#) to create the necessary Google Cloud entities and connections. On Google Cloud, you need to:

- prepare a VPC network.
- create a Service Directory service endpoint using the private IP address of CipherTrust Manager and port 443.
- authorize CloudEKM to access the VPC.
- create a connection using Location, Service Directory, EKM Hostname (CipherTrust Manager hostname), and CipherTrust Manager web certificate. This connection is created in your key project using KMS infrastructure.
- create a Cloud EKM key using the EKM or EKM UDE endpoint Key URI path.

Do the following to obtain CipherTrust Manager-related values needed by Google Cloud:

1. Note the fully qualified domain name of the CipherTrust Manager. This value and the port number **443** are needed to create the Service Directory service endpoint.
2. Download the web interface certificate.

```
ksctl interfaces certificate get --name web --icertfile <desired-filename>.pem
```

On the web console, this certificate is available at **Admin Settings > Interfaces**.

3. Convert the downloaded certificate to DER format. This is needed to create the EKM connection on Google Cloud.

```
openssl x509 -outform der -in <filename>.pem -out <filename>.der
```

4. Make a note of where the CipherTrust Manager is geographically deployed. In Google Cloud, choose a location for the EKM connection that is as close as possible to reduce network latencies.
5. Copy the Key URI path of the EKM or EKM UDE endpoint. When you create the corresponding Cloud EKM key, provide the Key path in the format `api/v1/cckm/ekm/endpoints/<endpoint_id>`. Do **not** include the CipherTrust Manager hostname in the path.

Once a connection is established, consult documentation for your desired [Google CMEK service integrated with Cloud EKM](#) to grant permissions for the CMEK service to use the KEK in Cloud KMS. Consult this documentation as well for particular encryption and decryption scenarios for the CMEK service.

For example:

- Protecting BigQuery data with Cloud KMS keys
- Protecting Compute Engine resources with Cloud KMS keys

Note

Only CMEK services integrated with EKM are supported with CCKM EKM endpoints. These are "Hold Your Own Key" (HYOK) integrations, where you manage and control the base KEK inside of CCKM. Google Cloud has additional CMEK services that do not follow the HYOK model and do not integrate with EKM. Consult [Google Cloud EKM documentation](#) for a list of supported CMEK services.

For using the EKM UDE functionality either on premise or in a confidential VM, consult the [Google Cloud Platform open source repository for the Split Trust Encryption Tool](#).

Rotate the Key Encryption Key

For EKM UDE endpoints, and EKM endpoints using symmetric key type, you can create a new version of the KEK which will be applied to future wrap operations. This can be done even if the endpoint is disabled. Rotating the KEK regularly is a security best practice.

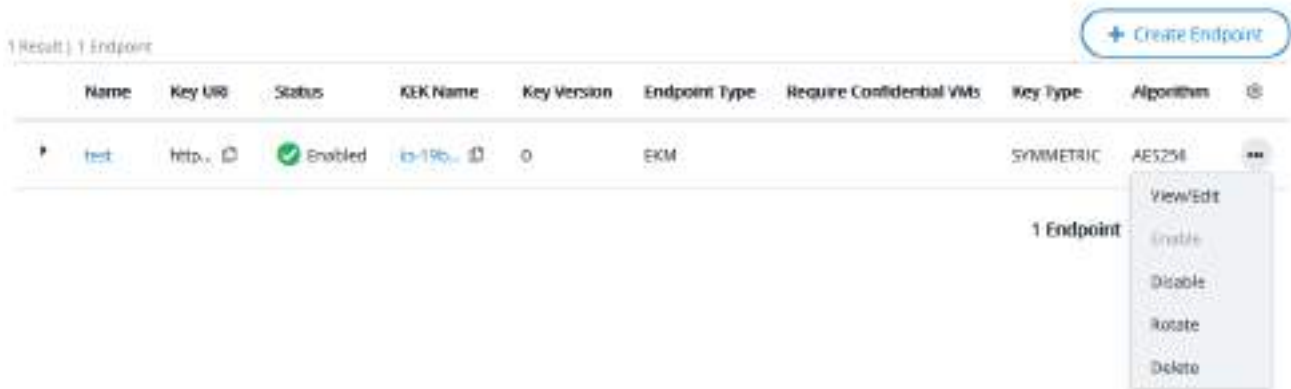
Note

There is no extra configuration required on Google Cloud. The next time Google Cloud tries a wrap operation with the KEK, the new key material is automatically used to wrap.

You can also post to the `/v1/cckm/ekm/endpoints/{id}/rotate` REST API endpoint, as described in the API Guide, or use `ksctl cckm ekm endpoints rotate --id <endpoint_id>` in the CLI.

In the GUI:

1. Login to the CipherTrust Manager products page as a user in the 'CCKM Admins' group.
2. Navigate to **Cloud Key Manager>Services>Google Cloud EKM**.
3. Find the endpoint in the list, and click the ellipsis icon (...) at the far right for options.



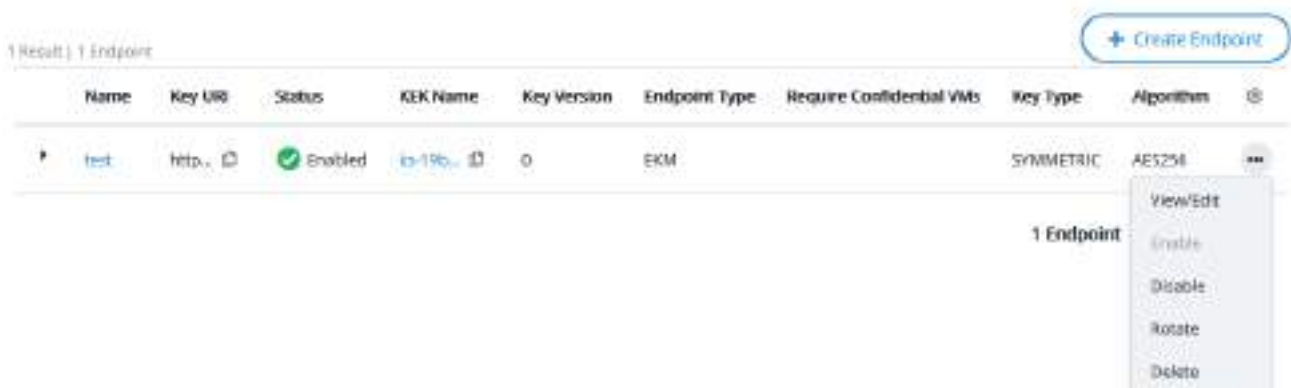
4. Select **Rotate**.

Change the base hostname

You can also patch the `/v1/cckm/ekm/endpoints/{id}` REST API endpoint, as described in the API Guide, or use `ksctl cckm ekm endpoints update --id <endpoint-id> --hostName <new-base-url-hostname>` in the CLI.

In the GUI:

1. Login to the CipherTrust Manager products page as a user in the 'CCKM Admins' group.
2. Navigate to **Cloud Key Manager>Services>Google Cloud EKM**
3. Find the endpoint in the list, and click the ellipsis icon (...) at the far right for options.



- Click **View/Edit**.
- In the **General Info** section, enter a new **Key URI hostname** and click **Update**.

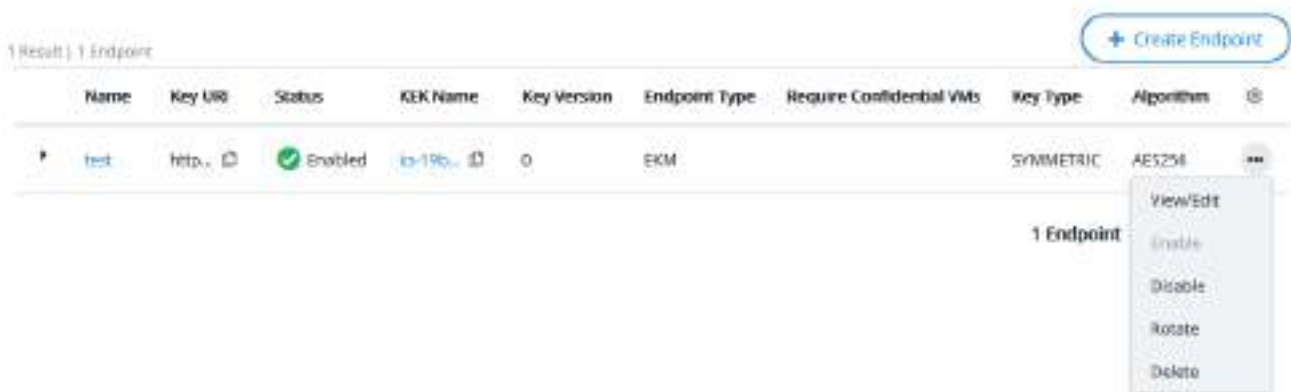
Enable or Disable Key Wrapping

You can post to the `/v1/cckm/ekm/endpoints/{id}/enable` and `/v1/cckm/ekm/endpoints/{id}/disable` REST API endpoints, as described in the [API Guide](#). In the CLI, you can use `ksctl cckm ekm endpoints enable --id <endpoint-id>` and `ksctl cckm ekm endpoints disable --id <endpoint-id>`.

This is a way to temporarily remove and restore client access to an endpoint without permanently deleting the endpoint and its KEK.

In the GUI:

- Login to the CipherTrust Manager products page as a user in the 'CCKM Admins' group.
- Navigate to **Cloud Key Manager>Services>Google Cloud EKM**
- Find the endpoint in the list, and click the ellipsis icon(...) at the far right for options.



- Click **Enable** or **Disable**.

Delete the Endpoint

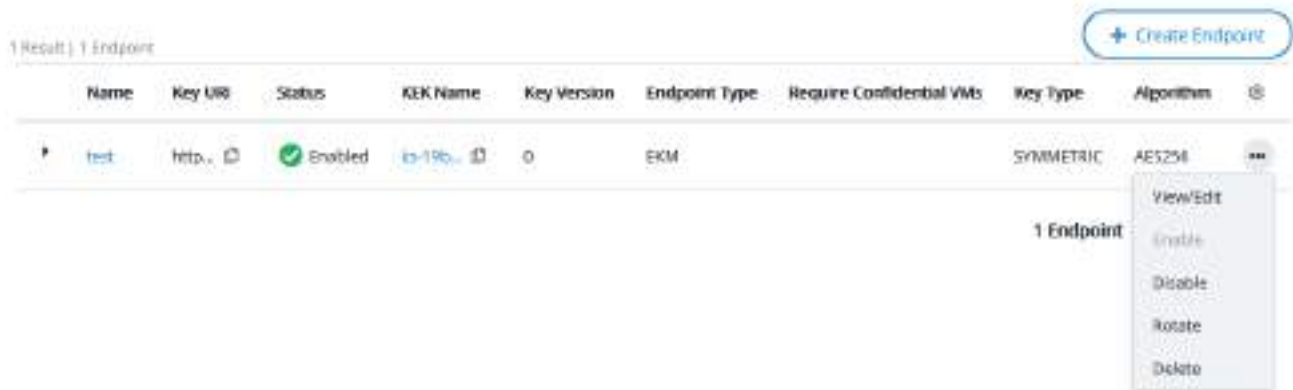
You can delete the `/v1/cckm/ekm/endpoints/{id}/REST API endpoint`, as described in the [API Guide](#). In the CLI, you can use `ksctl cckm ekm endpoints delete --id <endpoint-id>`. This permanently deletes the KEK.

Caution

Once the endpoint has been deleted, the key cannot be restored.

In the GUI:

1. Login to the CipherTrust Manager products page as a user in the 'CCKM Admins' group.
2. Navigate to **Cloud Key Manager>Services>Google Cloud EKM**
3. Find the endpoint in the list, and click the ellipsis icon(...) at the far right for options.



4. Click **Delete**.

A confirmation window appears, as this operation is irreversible.

5. Enable the **I wish to delete this endpoint** checkbox and click **Delete**.

View Activity for a Wrap or Unwrap Endpoint

Google Cloud KMS calls EKM endpoints to perform a wrap or unwrap operation on a base64 blob, and therefore protect some data on behalf of a CMEK service integrated with Cloud EKM such as BigQuery or Compute Engine. EKM UDE endpoints are called from workloads via Google-provided integration tools.

Standard EKM wrapping functionality is performed with `/v1/cckm/ekm/endpoints/{id}:wrap` and `/v1/cckm/ekm/endpoints/{id}:unwrap` REST API endpoints, as described in the [REST API endpoint](#). Google provided tools can find and make calls to wrap and unwrap endpoints without user intervention, if Google Cloud KMS has correctly configured the Cloud EKM key, and the CMEK service is correctly configured to access the key on Google Cloud KMS. This configuration happens as part of [preparing a new endpoint](#).

EKM UDE endpoints also allow for confidential wrapping and unwrapping, which requires the use of a secure session and a policy enforcement check with an authorized confidential VM, through `/v1/cckm/ekm/endpoints/{id}:confidentialwrap` and `/v1/cckm/ekm/endpoints/{id}:confidentialunwrap`.

Requests to these endpoints generate a record under **Records> Server Records** in the GUI, and the `/v1/audit/records` endpoint in the API. These records can be helpful to monitor EKM activity or troubleshoot EKM problems.

View Attestation Activity for an EKM UDE Endpoint

In EKMS, using an EKM UDE endpoint for wrap or unwrap operations requires a three-step process:

1. The establishment of a secure TLS 1.3 session between the Google-provided integration component and CipherTrust Manager.

2. The optional provision of attestation information/proofs over this secure session, allowing CCKM to assess the data-in-use protection capabilities of the requester.

Note

During this process, CCKM attempts to fetch a public key. This operation requires a Google service account with the IAM permission `compute.instances.getShieldedInstanceIdentity` registered with CCKM as a connection. If the public key cannot be fetched, an error is logged, `unable to check validity of instance key...` Wrap and unwrap operations are allowed to proceed.

3. The submission and handling of (confidential) wrap and unwrap requests, over this secure session.

The TLS sessions which are established have a session lifetime of 600 seconds, allowing multiple confidential wrap/unwrap requests to be handled by any given session. Thus, any confidential wrap or unwrap request can be associated with a secure channel and any secure channel can be associated with a set of attestation proofs (or none in the case of an on premise, non-confidential environment).

All relevant details are captured in **Records> Server Records** in the GUI, and the `/v1/audit/records` endpoint in the API, allowing cross-referencing of confidential wrap/unwrap requests with sessions with attestations.

Note

The EKM UDE API contains a set of 'Session' endpoints, namely: `session/begin session`; `session/handshake`; `session/negotiate attestation`; `session/finalize` and `session/end session`.

These endpoints allow the establishment of a secure TLS 1.3 session between the Google-provided integration component and CipherTrust Manager. Further details on these endpoints is beyond the scope of this document.

Managing Google EKM Endpoint Policies

When you create a Google External Key Manager (EKM) or EKM Ubiquitous Data Encryption (EKM UDE) endpoint, policies are associated with it to control which requests are processed. Users in the 'CCKM Admins' group can edit the EKM or EKM UDE endpoint policies. While the CipherTrust Manager web console and the REST API provide a basic format for editing policies, the **Raw** editing options requires understanding of the [Open Policy Agent \(OPA\)](#) policy engine.

If you wish to edit the Key Access Justification Reasons, this feature is described in detail in [Google Cloud Platform documentation](#).

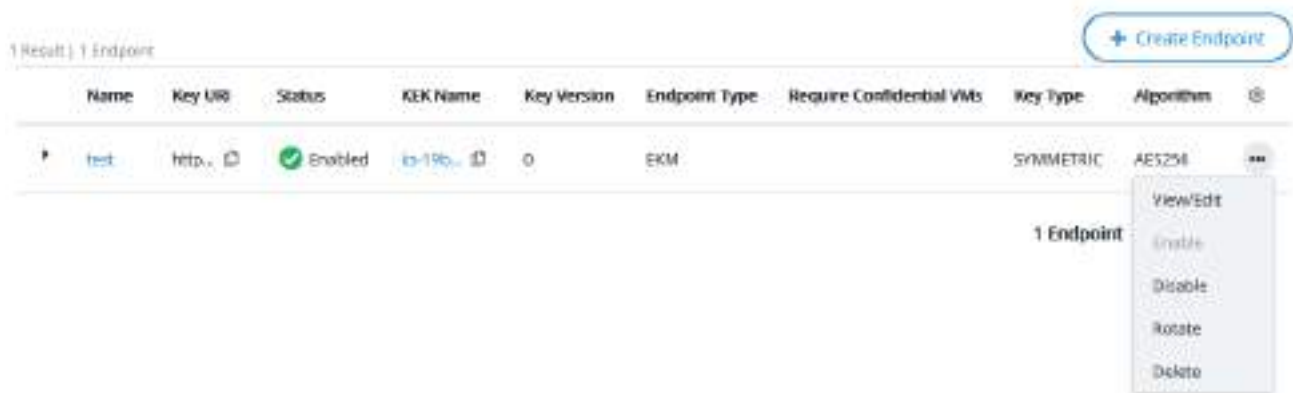
Edit the Policies for EKM and EKM UDE Endpoints

To edit the policies, you can also patch the `/v1/cckm/ekm/endpoints/{id}/policies` [REST API](#)

`endpoint`, or use `ksctl cckm ekm endpoints policy update --id <policy-id> --ekm-endpoint-update-policy-file <filename>` to pass in a new policy file.

In the GUI:

1. Login to the CipherTrust Manager products page as a user in the 'CCKM Admins' group.
2. Navigate to **Cloud Key Manager>Services>Google Cloud EKM**
3. Find the endpoint in the list, and click the ellipsis icon(...) at the far right for options.



4. Click **View/Edit**.

The policy options are visible in the **Endpoint Policy** section. The **Basic View** is selected by default.

5. Skip to step 6 for editing in **Raw View**. In **Basic View**, you can edit the following policies:
 - Clients allowed to use the endpoint, specified as a list separated by commas. Google service accounts are in the format of an email address, `<service-account-name>@<project-id>.iam.gserviceaccount.com`.
 - Whether **Key Access Justifications** are required or not.
 - If **Key Access Justifications** are required, which justification reasons are accepted.
 - EKM UDE endpoints have additional fields for **Zones**, **Project IDs**, and **Instance Names**. These settings constrain which workloads can use the EKM UDE Endpoint, and are enforced whenever a Confidential VM originates a request.
6. In **Raw View**, you can edit the following parameters:
 - `package` controls the rego policy package name.
 - `default allow` is a mandatory line to declare whether the rego policy is enforced. Setting this to `false` enables the policy and setting it to `true` disables the policy.
 - `input.clients` controls which clients are allowed to access the endpoint. This should match the clients on Google Cloud Service Accounts that are allowed to perform wrap or unwrap operations.

- `default allowedJustification` controls whether [Key Access Justifications](#) are required or not.
- `input.justificationReason` controls what justification reason needs to be provided for Google Cloud EKM to initiate a wrap or unwrap operation. You can remove this line, or comment it out with `#` at the start of the line if you do not require the feature.
- EKM UDE endpoints have additional lines to set Zones, Project IDs, and Instance Names, which are `input.attestationZones`, `input.attestationProjectIDs`, and `input.instanceNames` respectively. These settings constrain which workloads can use the EKM UDE Endpoint, and are enforced whenever a Confidential VM originates a request.

7. Click **Update**.

Key Access Justifications

Justification reasons are used by the [Key Access Justifications](#) feature in Google Cloud. This feature is optional for EKM and EKM UDE. When justification reasons are set, they need to be provided for Google Cloud EKM to initiate a wrap or unwrap operation.

The supported justification reasons are:

- REASON_UNSPECIFIED
- CUSTOMER_INITIATED_SUPPORT
- GOOGLE_INITIATED_SERVICE
- THIRD_PARTY_DATA_REQUEST
- GOOGLE_INITIATED_REVIEW
- CUSTOMER_INITIATED_ACCESS
- GOOGLE_INITIATED_SYSTEM_OPERATION
- REASON_NOT_EXPECTED
- MODIFIED_CUSTOMER_INITIATED_ACCESS
- GOOGLE_RESPONSE_TO_PRODUCTION_ALERT
- MODIFIED_GOOGLE_INITIATED_SYSTEM_OPERATION
- CUSTOMER_AUTHORIZED_WORKFLOW_SERVICING

Note

These justification reasons appear with spaces instead of underscores and with lowercase letters in the

CipherTrust Manager GUI Basic View. For example, `GOOGLE_INITIATED_REVIEW` appears as **Google Initiated Review** in the Basic View.

In OPA format, this is expressed as:

```
package example
default allow = false
allow {
  # Uncomment and add specific clients in below line to allow wrap/unwrap from Google services
  # input.clients == {"abc@yahoo.com", "abc@google.com", "abc@msn.com"}[_]
  input.justificationReason == {"REASON_UNSPECIFIED", "CUSTOMER_INITIATED_SUPPORT", "GOOGLE_INITIATED_SERVICE", "THIRD_PARTY_DATA_REQUEST", "GOOGLE_INITIATED_REVIEW", "CUSTOMER_INITIATED_ACCESS", "GOOGLE_INITIATED_SYSTEM_OPERATION", "REASON_NOT_EXPECTED", "MODIFIED_CUSTOMER_INITIATED_ACCESS"}[_]
}
```

Google EKM Performance Summary

We have tested six environments to capture performance metrics for the CipherTrust Cloud Key Manager (CCKM) Google External Key Manager (EKM) integration. The results provided in this document demonstrate the effects of deployment choices on throughput for [EKM endpoint wrap operations](#). The results can help you plan your CipherTrust Manager deployment to meet your performance needs for Google EKM.

There is a Google requirement that wrap and unwrap requests must be completed within 150 ms and so results are presented to show the throughput possible before meeting that threshold.

Note

Actual performance numbers in your environment can be different. The results can vary based on factors such as how and where the CipherTrust Manager is deployed, CipherTrust Manager resources, the location of clients, the network connectivity, and how the traffic is load-balanced.

Tested Environments

All environments used an open source [k6.io tool](#) as the REST client, run from a Debian virtual machine. The Virtual Machine was hosted on Google Cloud Platform (GCP) in `us central 1a zone` with 8 vCPUs of 64 GB memory, and 50 GB memory disk.

The following CipherTrust Manager deployments were tested:

- Google Cloud Platform deployments:
 - [GCP Single Node with 4 CPUs 16 GB RAM](#)

- GCP Single Node with 8 CPUs 64 GB RAM
- GCP Two Node Cluster with 4 CPUs 16 GB RAM each
- GCP Two Node Cluster with 8 CPUs 64 GB RAM each
- AWS Cloud deployments:
 - AWS Cloud Single Node with 4 CPUs 16 GB RAM
 - AWS Cloud Single Node with 8 CPUs 64 GB RAM

CipherTrust Managers were deployed as geographically close to the `k6.io` client as possible, to avoid potential network latencies which can occur when crossing geographic regions. In your EKM deployment, we similarly recommend deploying the Virtual CipherTrust Manager instance geographically close to one of the [Google Cloud KMS regions](#) where you intend to set up the Google Cloud KMS key ring.

Network Requirements

The following ports were opened to ensure CipherTrust Manager/CCKM communication:

Type	Protocol	Port Range
SSH	TCP	22
HTTPS	TCP	443
PostgreSQL (for cluster)	TCP	5432

Test Process

The test consisted of starting a given number of virtual users to perform wrap operations on the EKM endpoint. Each user simulated a separate thread.

Total test duration was 40 seconds for each reading. The test was divided into the following increments:

- Ramp-up time was 5 seconds. Virtual users were started.
- Test duration was 30 seconds for each reading. Virtual users make wrap requests during that time.
- Ramp-down time was 5 seconds. Virtual users were stopped until there were zero active virtual users.

Repeating the Test in Your Environment

We have published the scripts used with `k6.io` on [Github](#), for you to repeat the tests in your own environments as desired.

Google Cloud Deployment Results

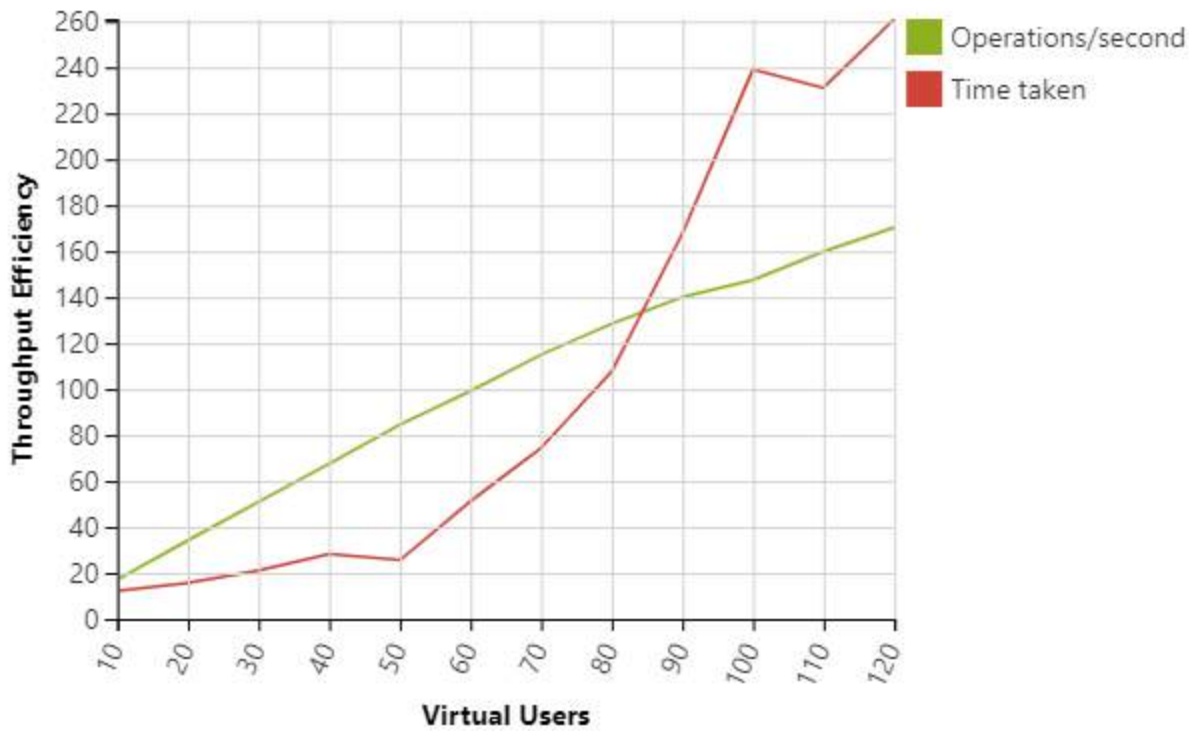
For each environment, operations per second are charted against response time (within 90% of the operations). The Google threshold of 150 ms is shown.

GCP Single Node with 4 CPUs 16 GB RAM

This deployment includes a single node with 4 CPUs and 16 GB memory. Both the CCKM instance and k6 client are on GCP. Here are details of this deployment:

Setup	System Volume	CPUs	Memory	CM Location
k6 Client	50 GB	8	64 GB	us-central1-a
CipherTrust Manager-2.11.1-9650	50 GB	4	16 GB	us-central1-a

Virtual Users	Run time	operations/second	Time taken (ms)
10	40	17.128677	12.28
20	40	34.131589	15.65
30	40	51.078121	21.17
40	40	67.495063	28.18
50	40	84.489887	25.68
60	40	99.229655	51.26
70	40	115.031462	74.41
80	40	128.49213	107.74
90	40	140.081857	167.61
100	40	147.564764	239.16
110	40	159.864781	231.13
120	40	170.39233	261.07



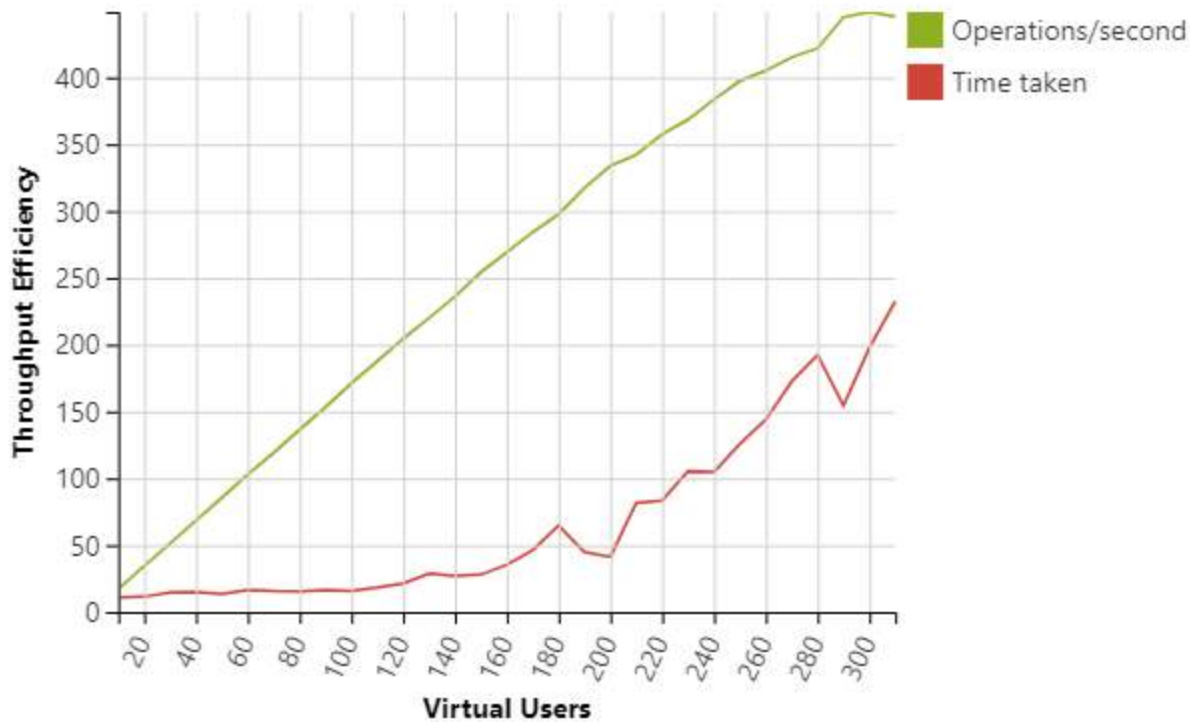
GCP Single Node with 8 CPUs 64 GB RAM

This deployment includes a single node with 8 CPUs and 64 GB memory. Both the CCKM instance and k6 client are on GCP. Here are details of this deployment:

Setup	System Volume	CPUs	Memory	CM Location
k6 Client	50 GB	8	64 GB	us-central1-a
CipherTrust Manager-2.11.1-9650	50 GB	8	64 GB	us-central1-a

Virtual Users	Run time	operations/second	Time taken (ms)
10	40	17.195846	10.71
20	40	34.483597	11.42
30	40	51.238013	14.41
40	40	68.537726	14.78
50	40	85.408548	13.61
60	40	102.824163	16.55
70	40	119.151866	15.62
80	40	136.375668	15.51
90	40	153.182252	16.48
100	40	171.242637	15.70

Virtual Users	Run time	operations/second	Time taken (ms)
110	40	187.627874	18.24
120	40	204.609053	21.36
130	40	220.006562	28.71
140	40	236.131067	27.01
150	40	254.164297	27.95
160	40	269.268795	35.34
170	40	284.350237	46.32
180	40	297.665916	64.83
190	40	317.407079	44.94
200	40	333.846892	41.24
210	40	342.163071	81.55
220	40	357.614392	83.49
230	40	368.557695	105.36
240	40	383.538218	104.80
250	40	397.327756	125.62
260	40	405.167935	143.98
270	40	415.166388	172.50
280	40	421.803097	192.47
290	40	445.06695	154.35
300	40	449.011222	197.75
310	40	445.623924	232.46



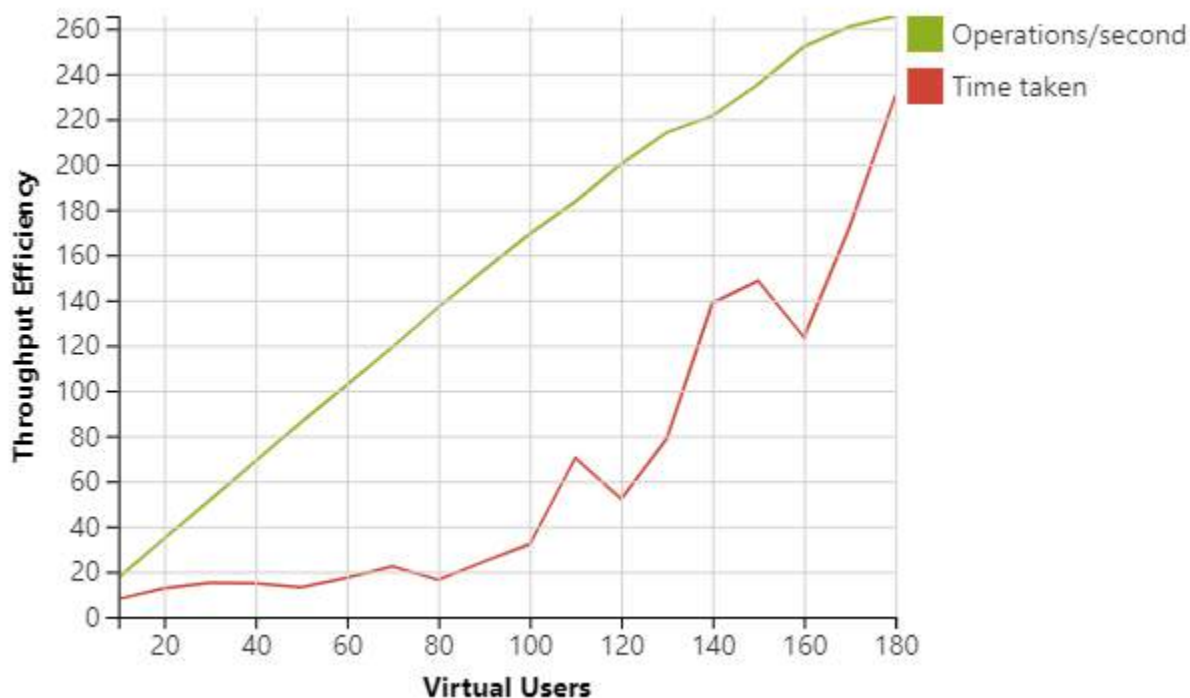
GCP Two Node Cluster with 4 CPUs 16 GB RAM each

This deployment includes two clustered CipherTrust Manager nodes and a load balancer. Each node has 4 CPUs and 16 GB RAM. The CCKM instances, load balancer, and k6 client are on GCP. Here are details of this deployment:

Setup	System Volume	CPUs	Memory	CM Location
k6 Client	50 GB	8	64 GB	us-central1-a
CipherTrust Manager-2.11.1-9650	50 GB	4	16 GB	us-central1-a
CipherTrust Manager-2.11.1-9650	50 GB	4	16 GB	us-central1-a

Virtual Users	Run time	operations/second	Time taken
10	40	16.933648	25.36
20	40	33.919138	27.43
30	40	50.790014	24.65
40	40	67.988872	25.38
50	40	84.994217	24.36
60	40	101.052646	25.73
70	40	117.893556	27.03

Virtual Users	Run time	operations/second	Time taken
80	40	134.42209	29.42
90	40	151.622632	26.88
100	40	167.016231	34.71
110	40	183.838216	31.55
120	40	199.663086	40.23
130	40	215.480264	52.79
140	40	231.106931	47.81
150	40	246.236306	56.34
160	40	260.515979	69.89
170	40	276.584743	71.87
180	40	285.872371	103.50
190	40	298.094234	127.96
200	40	312.950692	133.10
210	40	323.618858	174.34
220	40	320.245097	249.39



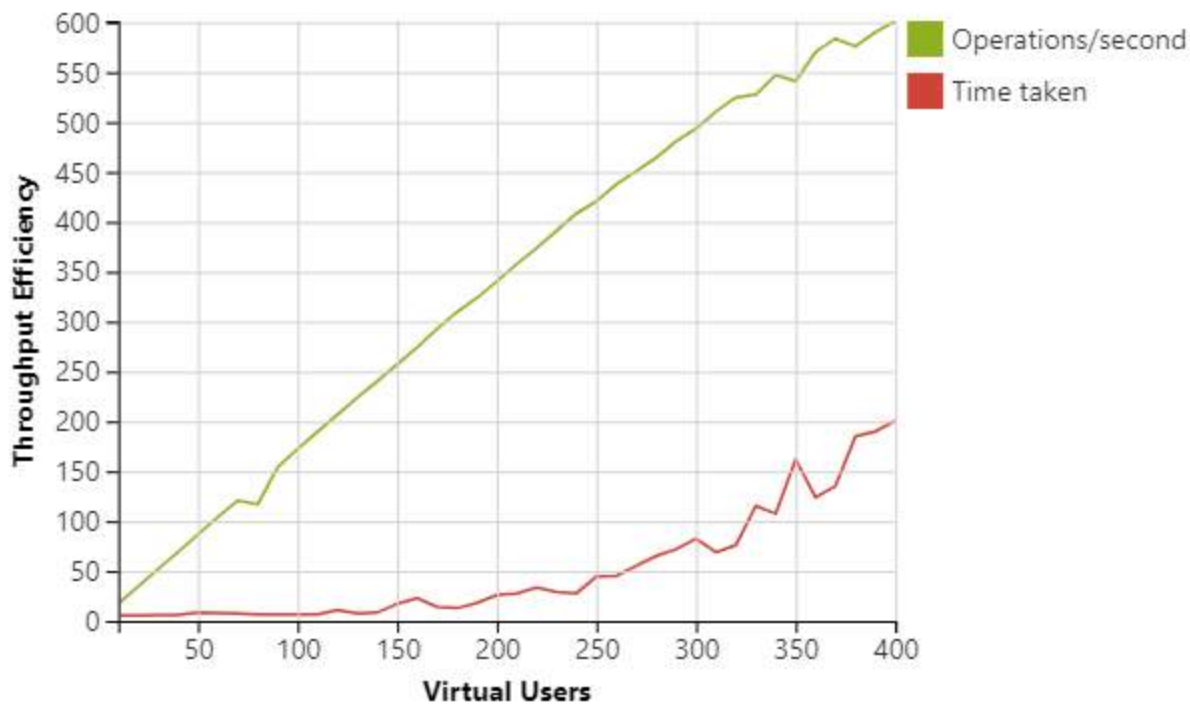
GCP Two Node Cluster with 8 CPUs 64 GB RAM each

This deployment includes two clustered CipherTrust Manager nodes and a load balancer. Each node has 8 CPUs and 64 GB RAM. The CCKM instances, load balancer, and k6 client are on GCP. Here are details of this

deployment:

Setup	System Volume	CPUs	Memory	CM Location
k6 Client	50 GB	8	64 GB	us-central1-a
CipherTrust Manager-2.11.1-9650	50 GB	8	64 GB	us-central1-a
CipherTrust Manager-2.11.1-9650	50 GB	8	64 GB	us-central1-a

Virtual Users	Run time	operations/second	Time taken
50	40	85.517417	14.46
100	40	169.774173	16.39
150	40	255.329062	18.37
200	40	340.257077	17.98
250	40	425.298916	19.19
300	40	508.189417	21.05
350	40	589.549419	28.54
400	40	672.584463	29.66
450	40	744.992112	51.04
500	40	811.565106	76.48
550	40	868.288425	122.31
600	40	916.005949	170.52
650	40	941.168986	225.58
700	40	966.152862	309.11
750	40	1030.021185	441.66
800	40	1036.617214	517.10



AWS Cloud Deployment Results

For each environment, operations per second are charted against response time (within 90% of the operations).

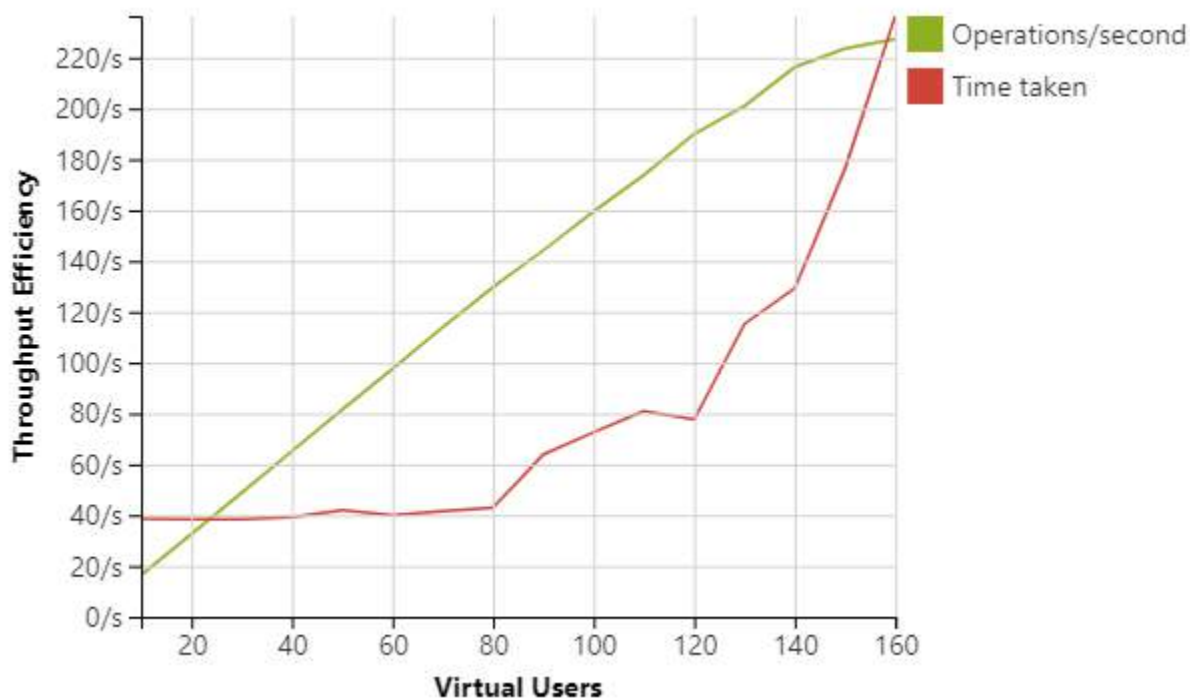
AWS Cloud Single Node with 4 CPUs 16 GB RAM

This deployment includes a single node with 4 CPUs and 16 GB memory. The CCKM instance is on AWS cloud and the K6 client is on GCP. Here are details of this deployment:

Setup	System Volume	CPUs	Memory	CM Location
k6 Client	50 GB	8	64 GB	us-central1-a
CipherTrust Manager-2.11.1-9650	50 GB	4	16 GB	us-east-1 (N. Virginia)

Virtual Users	Run time	operations/second	Time taken (ms)
10	40	16.463051	38.64
20	40	32.609432	38.45
30	40	48.793795	38.33
40	40	65.174326	39.23
50	40	81.490359	41.86
60	40	97.587339	39.93
70	40	113.92342	41.60

Virtual Users	Run time	operations/second	Time taken (ms)
80	40	129.689105	43.01
90	40	144.200816	63.91
100	40	159.4371	72.62
110	40	173.811015	80.92
120	40	189.979549	77.67
130	40	201.09059	115.29
140	40	216.295868	129.43
150	40	223.625394	176.25
160	40	227.379309	236.40



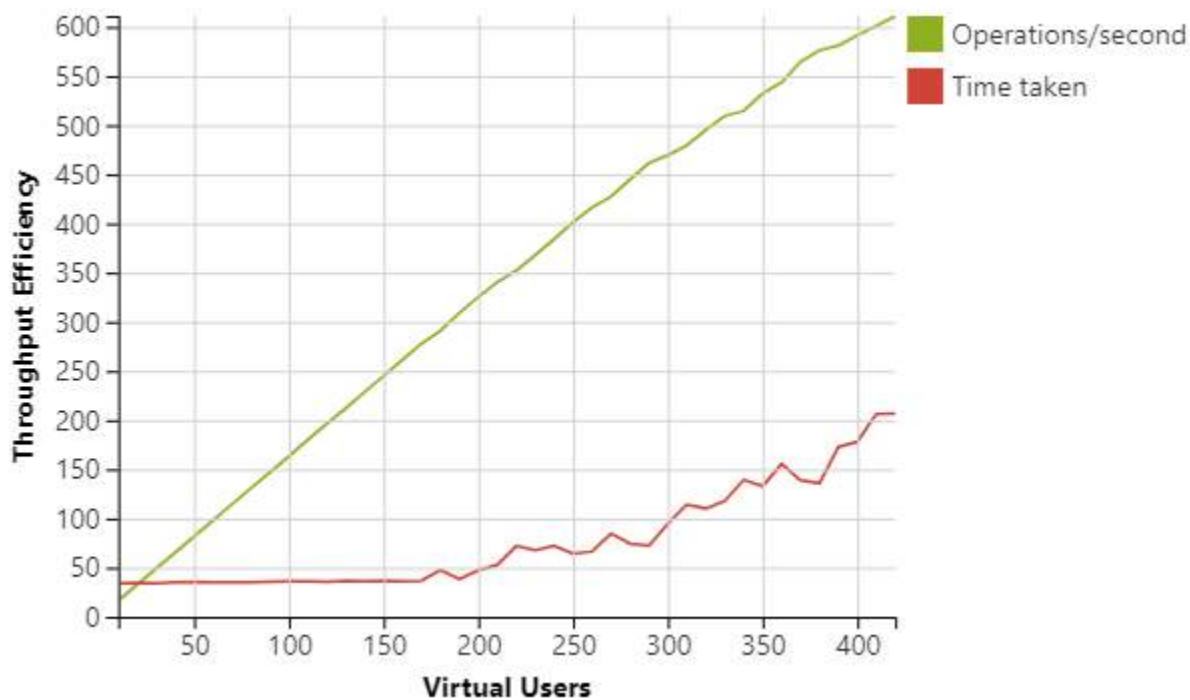
AWS Cloud Single Node with 8 CPUs 64 GB RAM

This deployment includes a single node with 8 CPUs and 64 GB memory. The CCKM instance is on AWS cloud and the k6 client is on GCP. Here are details of this deployment:

Setup	System Volume	CPUs	Memory	CM Location
k6 Client	50 GB	8	64 GB	us-central1-a
CipherTrust Manager-2.11.1-9650	50 GB	8	64 GB	us-east-1 (N. Virginia)

Virtual Users	Run time	operations/second	Time taken (ms)
10	40	16.492342	33.99
20	40	32.90742	34.79
30	40	49.043675	34.04
40	40	65.427584	35.19
50	40	81.827261	35.74
60	40	98.073539	35.01
70	40	114.684917	35.28
80	40	130.841442	34.95
90	40	146.943168	35.58
100	40	163.183103	36.01
110	40	179.882543	36.39
120	40	196.185992	35.59
130	40	212.209226	36.52
140	40	228.54225	36.32
150	40	244.991603	36.47
160	40	261.448303	35.99
170	40	277.951132	36.75
180	40	291.084292	47.64
190	40	309.021605	38.30
200	40	325.403116	47.22
210	40	340.395326	52.90
220	40	352.025967	72.09
230	40	367.930293	67.80
240	40	384.138939	72.09
250	40	401.465902	64.28
260	40	416.115064	66.41
270	40	427.417691	84.61
280	40	444.777548	74.32
290	40	461.519153	72.49
300	40	469.440098	94.77
310	40	479.391403	113.95
320	40	495.338039	110.28
330	40	509.461413	117.88

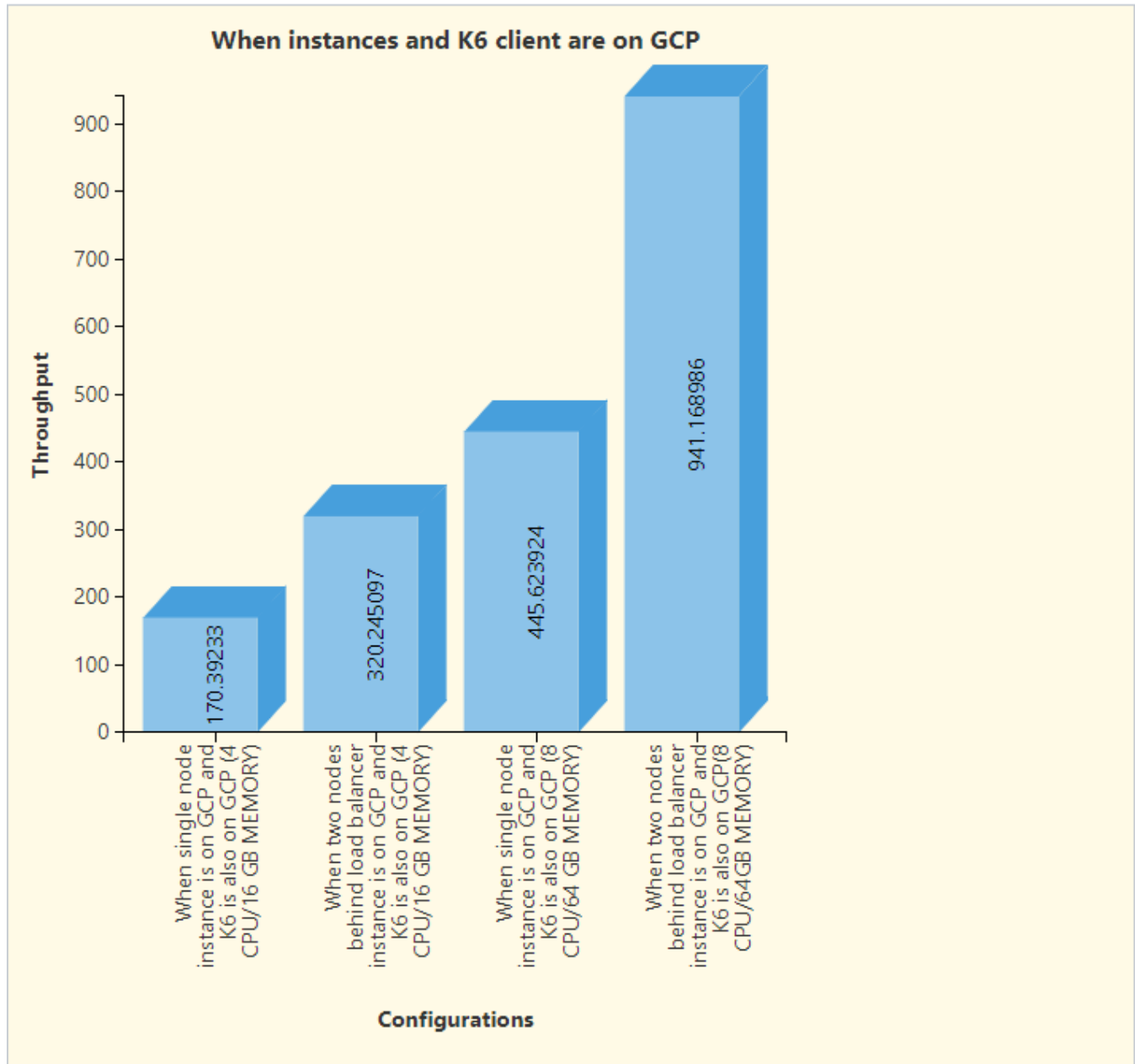
Virtual Users	Run time	operations/second	Time taken (ms)
340	40	514.51503	139.38
350	40	532.468021	132.88
360	40	543.400011	155.49
370	40	564.752063	139.05
380	40	576.253478	135.73
390	40	581.077997	172.77
400	40	591.480818	178.26
410	40	600.98002	206.44
420	40	611.056859	206.72



Conclusion

- Adding a second load balanced node scales performance approximately linearly. A larger performance increase was observed from adding CPU and RAM in each cloud environment.

Throughput comparison between single node and two node instances



Managing Google EKM Cryptospaces

GCP also allows users to use EKM in the Google Cloud Key Management Service (KMS) for the creation and management of external keys through VPC connections that support Cloud KMS EKM management mode. Using the Cloud KMS EKM management mode in the VPC connection to the EKM, you create and manage your external keys from the Google Cloud KMS. The key material of these keys are generated from the EKM (CipherTrust Manager). These external keys that are created and managed using this type of VPC connection are also referred to as coordinated keys.

In support of the VPC connection type of Cloud KMS EKM management mode, CCKM provides cryptospaces. A cryptospace is a logical workspace only available in CCKM in which a group of keys resides. It is within a CCKM cryptospace that coordinated keys are created, rotated, and destroyed through a VPC connection. The EKM cryptospace keys (also referred to as endpoints) can only be managed through the Google Cloud KMS (and not

through CipherTrust Manager or CCKM).

As part of creating a cryptospace in CCKM, grant your Cloud KMS service account access to your cryptospace and the keys to be created in it. In addition, set up a Key Access Justifications (KAJ) policy to define which access justifications should be allowed or denied. Keys created in a given cryptospace inherit the cryptospace's default KAJ policy. For more information about the required parameters, see [Create an EKM Cryptospace](#).

Note

CCKM doesn't support Google BYOK for Google Trusted Partner Clouds (TPC).

This page provides the prerequisites to allow a connection between a CipherTrust Manager and the Google Cloud External Key Manager Service. It also provides steps to create and update a cryptospace in CCKM and to manage the identity providers. After an EKM cryptospace is created, you can:

- Refresh EKM cryptospaces
- View EKM cryptospaces
- Change the base hostname
- Block or unblock a cryptospace
- Delete a cryptospace
- Update Confidential VM requirement
- Update service account
- Update cryptospace policy

To view cryptospace endpoints, refer to [Managing Google EKM cryptospace endpoints](#).

Note

This document assumes that you are familiar with Google Cloud EKM and Cloud EKM keys to protect your data.

Prerequisites

To allow a connection between CipherTrust Manager and Google Cloud External Key Manager Service, some network and security configuration must be in place in both entities.

Google Cloud platform prerequisites

- A Google Cloud project using Cloud KMS must exist for the CCKM EKM integration's use. You can create a

new project or choose an existing project in your Google Cloud account. You need to provide either a project ID or the private key file associated with the service account to add a Google project to CipherTrust Manager. The project ID is the simplest path to add the Google Cloud project just for EKM usage. The private key file is required if you want [add a Google Cloud connection to CipherTrust Manager](#).

- For the confidential computing environments for EKM UDE, an additional Google Cloud project needs to be associated with the confidential computing environments. A Google service account with the Identity and Access Management (IAM) permission `compute.instances.getShieldedInstanceIdentity` must have access to the project. This permission is required to carry out full verification of attestation evidence.
- If you wish to require [Key Access Justifications \(KAJ\)](#) you need to configure KAJ on Google Cloud.
- Refer to [Create an EKM connection via VPC](#) for information on how to create an EKM connection via VPC from the Google console. Note there are prerequisite steps for creating an EKM connection via VPC that are provided within this Google documentation, including preparing a VPC network, creating a Service Directory service endpoint, and authorizing Cloud EKM to access your VPC. Be sure to perform all required prerequisite steps. Also included in the prerequisite steps is high-level information about how to set up your EKM.
- Refer to [Create an external key](#) for information on how to create Cloud EKM keys on an existing key ring in Cloud KMS. Refer to [Create a coordinated external key](#) to create the coordinated external keys.

CipherTrust Manager Prerequisites

- The CipherTrust Manager must have an IP address that is accessible from Google KMS. This is done by setting up a VPC. See [VPC networks](#) for more information.
- You can change the default port of 443 for the CipherTrust Manager web interface. One would have to ensure to change it **BEFORE** configuring the given cloud service on CCKM. However, in the case of Google Cryptospace, Google only accepts and sends requests on the default port of 443. If you wish to use a different port for the web interface other than the default, add a network component, such as a load balancer or a firewall, in front of CipherTrust Manager. Then configure port mapping on this component by mapping port 443 to the non-default port on the CipherTrust Manager you plan to use. For more information, refer to [Support for Changing the Default Port of Web Interface Setting](#).
- The web interface must have a [TLS certificate signed by an external Certificate Authority \(CA\) trusted by Google Cloud Platform](#). Google Cloud trusts certificates issued by well-known public CAs such as Verisign. Alternatively, you can create a certificate chain with Google's [Certificate Authority Service](#) and upload the chain to CipherTrust Manager.

For correct operation of UDE in a cluster of CipherTrust Manager nodes, all of the nodes must be configured with the same TLS certificate. This requirement is a consequence of termination of TLS sessions for UDE occurring at the individual CipherTrust Manager nodes rather than at the cluster load balancer.

EKM UDE uses the same TLS certificate the CCKM web interface uses. If the TLS certificate for the CCKM web interface is updated, a restart of the CCKM service is required for the CCKM for UDE to use the updated certificate.

- A Google Cloud project using Cloud KMS must be added to the CCKM before creating a cryptospace.
- For EKM UDE endpoints using confidential computing environments, the Google Cloud project associated with the confidential computing environments must be added to CCKM through [connection manager](#) for CCKM to completely verify attestation evidence.

Create an EKM cryptospace

To create a Google EKM cryptospace in the CCKM GUI and to make it available to Google Cloud EKM.

1. Add the Google Cloud project using Google Cloud KMS to CCKM.

This action consumes one [CCKM license cloud unit](#).

Note

These steps demonstrate the simplest configuration to add a Google project ID to CCKM just for EKM cryptospaces, without using an active connection to Google Cloud. If you want to monitor and manage a Google Cloud account connection and associated resources on CipherTrust Manager, you can also [add a Google Cloud connection using connection manager](#), and then use the connection to retrieve project IDs.

- a. On Google Cloud, create a new project or choose an existing one.
- b. Copy the project ID.
- c. In CipherTrust Manager, open the **Cloud Key Manager** application.
- d. Navigate to **KMS Containers > Google** and open the **Projects** tab.
- e. Click **Add Existing Project**. The **Add Existing Google Project** screen displays.
- f. Under **Select Method**, select **Manually Enter Project ID**. This option allows you manually enter a project ID to connect to a project without credentials. This option is only applicable to creating and managing Key Encryption Keys (KEKs) for EKM usage.

The **Select From List** option allows you to select an existing project from a list of connections stored in CipherTrust Connection Manager. This option is applicable to creating and managing Google Cloud keys.

- g. In **Project ID**, paste in the project ID.
- h. (Optional) Use the **Enable success audit events** toggle to enable or disable audit recording of successful operations within the given Google cloud project. This toggle is set to enable, by default.
- i. Click **Add Project**.

2. Login as a user in the 'CCKM Admins' group to create the cryptospace in the CCKM GUI.

Note

You can also create the cryptospace with the REST API or CLI to to associate meta information with the endpoint. Use the `/v1/cckm/ekm/cryptospaces` endpoint in the [REST API](#), or `ksctl cckm ekm cryptospaces create --ekm-cryptospace-create-jsonfile <meta_information_filename>` in the CLI.

3. Navigate to **Cloud Key Manager > Services > Google Cloud EKM**.
4. Select the **CryptoSpace(s)** tab.
5. Select **Create CryptoSpace**. The General Info screen of the Create CryptoSpace wizard is displayed.

General Info

1. Specify a unique **Name** for the cryptospace.
2. (Optional) Enter a **Description** for the cryptospace.
3. Click **Next**. The CryptoSpace Settings screen is displayed.

CryptoSpace Settings

1. Select a **Google Cloud Type**. The options are **Google Cloud Platform** and **Trusted Partner Cloud**.
2. Specify a **Hostname**. This is the base url hostname for Ciphertrust Manager.
3. In the **Project ID** drop-down list, select the GCP project to be associated with this cryptospace in CCKM.
4. Add the location. The options are:
 - a. **Select Location From List**: Select the desired location of the cryptospace from the drop-down list.
 - b. **Manually Type Location**: Enter the desired Location.
5. Click **Next**. The CryptoSpace Type screen is displayed.

CryptoSpace Type

1. Select either **EKM CryptoSpace** or **Ubiquitous Data Encryption CryptoSpace** from the available CryptoSpace types. The default is **EKM CryptoSpace**.

The Ubiquitous Data Encryption CryptoSpace type allows you to create an EKM cryptospace that supports UDE. For this, the default requirement for Confidential VMs is **Not Required**. If selecting **Ubiquitous Data Encryption CryptoSpace**, select the requirement for Confidential VMs to originate wrap or unwrap requests under **Require Confidential VMs**. You can select **Not Required**, required **For Wrap and Unwrap**, required

For Wrap only, or required **For Unwrap** only.

2. Click **Next**. The Service Accounts screen is displayed.

Service Accounts

1. In **Service Account**, add the service account that is to be granted access to this cryptospace. Google service accounts are identified by an email in the form `service-account-name@project-id.iam.gserviceaccount.com`.
2. Click **Add Service Account**. The account is added to the Service Accounts section. Expand the caret icon to view the permissions assigned to this service account. By default, all the check boxes for the available permissions for the service account are selected. At least one permission must be selected for the given service account.

You can add multiple service accounts.

3. (Optional) Clear any permissions you wish to not use for the service account.
4. (Optional and applicable to Trusted Partner Cloud (TPC) only). Select the **Identity Providers** from the list.
5. Click **Update**.
6. Click **Next**. The CryptoSpace Policy screen is displayed.

CryptoSpace Policy

1. Provide the list of Google service accounts that are granted access to the endpoints in the **Clients** field, separating the clients with commas. Google service accounts are in the format of an email address, `<service-account-name>@<project-id>.iam.gserviceaccount.com`.
2. (Applicable to Google Cloud Platform only) Edit the Key Access Justification settings. You can disable the requirement for Key Access Justifications, or specify which justification reasons are needed for the EKM endpoint to initiate a wrap or unwrap request.
3. Click **Next**. The Review and Add screen is displayed.

Review and Add

1. Review your settings. To modify any of the sections (**GENERAL INFO**, **CRYPTOSPACE SETTINGS**, **CRYPTOSPACE TYPE**, **SERVICE ACCOUNTS**, or **CRYPTOSPACE POLICY**), click **Edit** for the given section.
2. Click **Create CryptoSpace**.
3. Click **Close**.

The EKM cryptospace is created.

After the EKM cryptospace is created, [create an EKM connection via VPC](#) from the Google console. As part of this step, select **Cloud KMS** as the **EKM management mode** and provide the cryptospace URL in the **Crypto Space path**. To obtain the cryptospace URL from CCKM, copy the path within the **Path** column of the **Cryptospace(s)** tab in the **Google Cloud External Key Manager** page. Note that there are prerequisite steps

provided within this Google documentation including preparing a VPC network, creating a Service Directory service endpoint, and authorizing Cloud EKM to access your VPC.

After you have created an EKM connection via VPC, proceed to [creating an external key](#) from the Google console. As part of creating an external key, select **External** under **Protection Level** and **via VPC** under **External key manager (EKM) connection type**. Select the EKM via VCP connection that you created from the drop-down list. Once the key is created, it displays on the **Key ring details** page within the Google console. It also then displays within **Cryptospace Endpoint(s)** tab in the **Google Cloud External Key Manager** page of CCKM. Thereafter, you can begin using these Cloud EKM keys to protect your data.

Refreshing EKM cryptospaces

To refresh all EKM cryptospaces:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > Google Cloud EKM**. The Google Cloud External Key Manager page displays.
3. Click the **CryptoSpace(s)** tab. The list of cryptospace added to CCKM is displayed.
4. Click **Refresh**.

The refreshed cryptospaces are listed on the **CryptoSpace(s)** page.

Viewing EKM cryptospaces

The **CryptoSpace(s)** tab shows the list of existing cryptospaces within CCKM. Search for cryptospace by Name or Project.

To view the list of Google cryptospace endpoints available on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > Google Cloud EKM**. The Google Cloud External Key Manager page displays.
3. Click the **CryptoSpace(s)** tab. The list of cryptospaces added to CCKM is displayed. The tab displays the following details:

Field	Description
Name	Name of the cryptospace.
Block	Indicates whether the cryptospace is blocked or unblocked.
CryptoSpace Type	Cryptospace type. There are two types of cryptospaces. One for EKM endpoints (ekm) and another for EKM UDE endpoints (ekm-ude).
Require Confidential VMs	Indicates whether a wrap or unwrap request is

Field	Description
	required to originate from a confidential VM.
Hostname	Base url hostname for Ciphertrust Manager.
Project	Name of the project.
Path	External-KMS-defined resource identifier for the cryptospace.
Endpoints	Number of endpoints within the cryptospace.
Location	Location of the cryptospace.
Google Cloud Type	Type of the Google Cloud - Google Cloud Platform or Trusted Partner Cloud (TPC).
Creation Date	Time when the cryptospace was created.

To view the custom columns, click the Customize View () icon, select the desired option, and click **OK** to display the column.

To view the Cryptospace details page, click on the name of the cryptospace under the **Name** column.

To view cryptospace endpoints, refer to [Managing Google EKM cryptospace endpoints](#).

Update the Description

In the GUI:

1. Login to the CipherTrust Manager GUI as a user of the 'CCKM Admins' group.
2. Navigate to **Cloud Key Manager > Services > Google Cloud EKM > CryptoSpace(s)** tab.
3. Find the cryptospace in the list.
4. Click the overflow icon () corresponding to the cryptospace.
5. Click **View/Edit**.
6. In the **GENERAL INFO** section, enter a new **Description** and click **Update**.

Changing the base hostname

You can patch the `/v1/cckm/ekm/cryptospaces/{id}` REST API endpoint, as described in the API Guide, or use `ksctl cckm ekm cryptospaces update --id <cryptospace-id> --ekm-cryptospace-hostname <new-base-url-hostname>` in the CLI.

In the GUI:

1. Login to the CipherTrust Manager products page as a user in the 'CCKM Admins' group.

2. Navigate to **Cloud Key Manager > Services > Google Cloud EKM > CryptoSpace(s)** tab.
3. Find the cryptospace in the list.
4. Click the overflow icon () corresponding to the cryptospace.
5. Click **View/Edit**.
6. In the **CRYPTOSPACE SETTINGS** section, enter a new **Hostname** and click **Update**.

Block or unblock a cryptospace

Blocking and unblocking a cryptospace is a way to temporarily suspend and restore access to the cryptospace and its keys.

You can block the `/v1/cckm/ekm/cryptospaces/{id}/block` REST API endpoint or unblock the `/v1/cckm/ekm/cryptospaces/{id}/unblock` REST API endpoint, as described in the API Guide. In the CLI, you can use `ksctl cckm ekm cryptospaces block --id <cryptospace-id>` to block a cryptospace and `ksctl cckm ekm cryptospaces unblock --id <cryptospace-id>` to unblock a cryptospace.

In the GUI:

1. Login to the CipherTrust Manager GUI as a user of the 'CCKM Admins' group.
2. Navigate to **Cloud Key Manager > Services > Google Cloud EKM > CryptoSpace(s)** tab.
3. Find the cryptospace in the list.
4. Click the overflow icon () corresponding to the cryptospace.
5. Click **Block** or **Unblock**, depending on whether the cryptospace is currently in a blocked or unblocked state.
6. You are asked to confirm the action. Click **Block** or **Unblock** again.

The status in the **Blocked** column updates on the table.

Delete a cryptospace

You can delete the `/v1/cckm/ekm/cryptospaces/{id}/` REST API endpoint, as described in the API Guide. In the CLI, you can use `ksctl cckm ekm cryptospaces delete --id <cryptospace-id>`. This permanently deletes the cryptospace.

Caution

Once the cryptospace has been deleted, it cannot be restored.

In the GUI:

1. Login to the CipherTrust Manager GUI as a user of the 'CCKM Admins' group.
2. Navigate to **Cloud Key Manager > Services > Google Cloud EKM > CryptoSpace(s)** tab.
3. Find the cryptospace in the list.
4. Click the overflow icon () corresponding to the cryptospace.
5. Click **Delete**.

A confirmation window appears, as this operation is irreversible.

6. Enable the **I wish to delete this cryptospace** checkbox and click **Delete**.

Update a Confidential VM requirement

You can patch the `/v1/cckm/ekm/cryptospaces/{id}` [REST API endpoint](#), as described in the API Guide, or use `ksctl cckm ekm cryptospaces update --id <cryptospace-id> --cvm-required-for-encrypt=true` in the CLI.

In the GUI:

1. Login to the CipherTrust Manager products page as a user in the 'CCKM Admins' group.
2. Navigate to **Cloud Key Manager > Services > Google Cloud EKM > CryptoSpace(s)** tab.
3. Find the cryptospace in the list.
4. Click the overflow icon () corresponding to the cryptospace.
5. Click **View/Edit**.
6. Under **CryptoSpace Type**, select/enter the following details:

- a. From the available cryptospace types, select **Ubiquitous Data Encryption CryptoSpace**.

This option allows you to create an EKM cryptospace that supports UDE.

- a. Under **Require Confidential VMs**, select the requirement for Confidential VMs to originate wrap or unwrap requests. You can select **Not Required**, required **For Wrap and Unwrap**, required **For Wrap** only, or required **For Unwrap** only.
- b. Click **Update**.

Update a service account

In the GUI:

1. Login to the CipherTrust Manager products page as a user in the 'CCKM Admins' group.

2. Navigate to **Cloud Key Manager > Services > Google Cloud EKM > CryptoSpace(s)** tab.
3. Find the cryptospace in the list.
4. Click the overflow icon () corresponding to the cryptospace.
5. Click **View/Edit**.
6. Under **Service Accounts**, select/enter the following details:
 - a. In **Input Service Account(s)**, add the service account that is to be granted access to this cryptoSpace. Google service accounts are identified by an email in the form `service-account-name@project-id.iam.gserviceaccount.com`.
 - b. Click **Add Service Account**.

The account is added to the Service Accounts section. Expand the caret icon to view the assigned permissions and identity providers. By default, all the checkboxes for the available permissions for the service account are selected. At least one permission must be selected for the given service account.

 - a. Clear the permissions you don't wish to use.
 - b. (Optional) Select the identity providers you wish to use.
 - c. Click **Update**.
7. Click **Update**.

Update a cryptospace policy

A cryptospace endpoint inherits its policy from the cryptospace in which it resides. Changing Cryptospace policy affects all the endpoints that reside in this cryptospace.

In the GUI:

1. Login to the CipherTrust Manager products page as a user in the 'CCKM Admins' group.
2. Navigate to **Cloud Key Manager > Services > Google Cloud EKM > CryptoSpace(s)** tab.
3. Find the cryptospace in the list.
4. Click the overflow icon () corresponding to the cryptospace.
5. Click **View/Edit**

The policy options are visible in the **CryptoSpace Policy** section. The **Basic View** is selected by default.

6. Skip to step 6 for editing in **Raw View**. In **Basic View**, you can edit the following policies:
 - Clients allowed to use the cryptospace, specified as a list separated by commas. Google service accounts are in the format of an email address, `<service-account-name>@<project-`

```
id>.iam.gserviceaccount.com.
```

- Whether [Key Access Justifications](#) are required or not.
- If [Key Access Justifications](#) are required, which justification reasons are accepted.
- EKM UDE endpoints have additional fields for **Zones**, **Project IDs**, and **Instance Names**. These settings constrain which workloads can use the EKM UDE Endpoint, and are enforced whenever a Confidential VM originates a request.

7. In **Raw View**, you can edit the following parameters:

- `package` controls the rego policy package name
- `default allow` is a mandatory line to declare whether the rego policy is enforced. Setting this to `false` enables the policy and setting it to `true` disables the policy.
- `input.clients` controls which clients are allowed to access the endpoint. This should match the clients on Google Cloud Service Accounts that are allowed to perform wrap or unwrap operations.
- `default allowedJustification` controls whether [Key Access Justifications](#) are required or not.
- `input.justificationReason` controls what justification reason needs to be provided for Google Cloud EKM to initiate a wrap or unwrap operation. You can remove this line, or comment it out with `#` at the start of the line if you do not require the feature.
- EKM UDE endpoints have additional lines to set Zones, Project IDs, and Instance Names, which are `input.attestationZones`, `input.attestationProjectIDs`, and `input.instanceNames` respectively. These settings constrain which workloads can use the EKM UDE Endpoint, and are enforced whenever a Confidential VM originates a request.

8. Click **Update**.

Key Access Justifications

Justification reasons are used by the [Key Access Justifications](#) feature in Google Cloud. This feature is optional for EKM and EKM UDE. When justification reasons are set, they need to be provided for Google Cloud EKM to initiate a wrap or unwrap operation.

The supported justification reasons are:

- `REASON_UNSPECIFIED`
 - `CUSTOMER_INITIATED_SUPPORT`
 - `GOOGLE_INITIATED_SERVICE`
 - `THIRD_PARTY_DATA_REQUEST`
 - `GOOGLE_INITIATED_REVIEW`
-

- CUSTOMER_INITIATED_ACCESS
- GOOGLE_INITIATED_SYSTEM_OPERATION
- REASON_NOT_EXPECTED
- MODIFIED_CUSTOMER_INITIATED_ACCESS
- GOOGLE_RESPONSE_TO_PRODUCTION_ALERT
- MODIFIED_GOOGLE_INITIATED_SYSTEM_OPERATION
- CUSTOMER_AUTHORIZED_WORKFLOW_SERVICING

Note

These justification reasons appear with spaces instead of underscores and with lowercase letters in the CipherTrust Manager GUI Basic View. For example, `GOOGLE_INITIATED_REVIEW` appears as **Google Initiated Review** in the Basic View.

In OPA format, this is expressed as:

```
package example
default allow = false
allow {
# Uncomment and add specific clients in below line to allow wrap/unwrap from Google services
# input.clients == {"abc@yahoo.com", "abc@google.com", "abc@msn.com"}[_]
input.justificationReason == {"REASON_UNSPECIFIED", "CUSTOMER_INITIATED_SUPPORT", "GOOGLE_INITIATED_SERVICE", "THIRD_PARTY_DATA_REQUEST", "GOOGLE_INITIATED_REVIEW", "CUSTOMER_INITIATED_ACCESS", "GOOGLE_INITIATED_SYSTEM_OPERATION", "REASON_NOT_EXPECTED", "MODIFIED_CUSTOMER_INITIATED_ACCESS"}[_]
}
```

Managing Identity Providers

Google Trusted Partner Cloud (TPC) is an isolated environment with a unique identity from Google EKM. If you are configuring CCKM for a Google Trusted Partner Cloud (TPC) environment, you can add a specific identity provider to authenticate the Service Account.

Creating Identity Providers

An issuer or an openId configuration URL is required to create an identity provider.

To create an identity provider:

1. Open the **Cloud Key Manager** Application.

2. In the left pane, click **Services > Google Cloud EKM**. The Google Cloud External Key Manager page displays.
3. Under **IDENTITY PROVIDERS**, click **Add Identity Provider**. The Add Identity Provider screen is displayed.
4. Specify a unique **Name** for the provider.
5. Select the **Provider Verifier**.

The verifier can be:

- **Issuer:** Select and specify the valid **Issuer** and **jwtURL**.
- **OpenID Configuration URL:** Select and specify the **OpenId Configuration URL**. The URL must be valid. For example, when using SafeNet Trusted Access (STA) as an identity provider, the URL is represented by **WELL KNOWN CONFIGURATION URL** on the STA Management Console.

6. Click **Add**.
7. Click **Close**.

The newly created identity provider appears in the list of identity providers.

Viewing Identity Providers

The Google Cloud External Key Manager page shows the available identity providers. The **IDENTITY PROVIDERS** section shows the Name, Issuer, OpenID Configuration URL, and jwtURL.

To view the identity providers:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Cloud EKM**. The Google Cloud External Key Manager page displays.
3. Navigate to **IDENTITY PROVIDERS**. The list of identity providers is displayed.

The following details are shown:

Column	Description
Name	Name of the identity provider.
Issuer	Issuer string from the identity provider JWT.
OpenID Configuration URL	URL of the OpenID configuration.
jwtURL	URL of JWKS.

Updating Identity Providers

To update an identify provider:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Cloud EKM**. The Google Cloud External Key Manager page displays.
3. Under **IDENTITY PROVIDERS**, click the overflow icon () corresponding to the provider you want to edit.
4. Click **Edit**. The Edit Identity Provider screen is displayed.
5. Edit **Name**.
6. Click **Update**.

Deleting Identity Providers

When an identity provider is no longer needed, delete it from the CipherTrust Manager. Before deleting the provider, ensure that it is not in use by any endpoints.

To delete an identify provider:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Cloud EKM**. The Google Cloud External Key Manager page displays.
3. Under **IDENTITY PROVIDERS**, click the overflow icon () corresponding to the provider you want to delete.
4. Click **Delete**. The Delete Identity Provider wizard is displayed.
5. Click **Delete Identity Provider**.

The identity provider is removed from the list of identity providers.

Managing Google EKM Cryptospace Endpoints

For an overview of EKM cryptospaces, refer to [Managing Google EKM cryptospaces](#)

After coordinated keys are created in a CCKM cryptospace from Google Cloud KMS through a VPC connection, these keys (or endpoints) are displayed within the **Cryptospaces Endpoints** tab of the **Google Cloud External Key Manager** page in CCKM.

Caution

Modify endpoints only through the CipherTrust Cloud Key Management (CCKM) GUI, API, or CLI. Do not

modify the KEK using the **Keys** menu, the `ksctl keys` command, or the `/v1/vault/keys2` endpoint. Such modifications might make the KEK unavailable to the Google Cloud EKM service.

Steps to refresh, view, and update a Google EKM cryptospace endpoint using the CCKM GUI are provided on this page:

- Refreshing EKM cryptospace endpoints
- View EKM cryptospace endpoints
- View details of an EKM cryptospace endpoint
- Enable or disable a key version of an EKM cryptospace endpoint

Refreshing EKM cryptospace endpoints

To refresh all EKM cryptospace endpoints:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > Google Cloud EKM**. The Google Cloud External Key Manager page displays.
3. Click the **Cryptospaces Endpoints** tab. The list of cryptospace endpoints added to CCKM from Google Cloud KMS through a VPC connection is displayed.
4. Click **Refresh*.

The refreshed cryptospace endpoints are listed on the **Cryptospaces Endpoints** page.

Viewing EKM cryptospace endpoints

The **Cryptospaces Endpoints** tab shows the list of existing cryptospace endpoints residing within a given cryptospace. Search for endpoints by Google Key Name, Google Resource Name, Project, or Cryptospace.

To view the list of Google cryptospace endpoints available on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > Google Cloud EKM**. The Google Cloud External Key Manager page displays.
3. Click the **Cryptospaces Endpoints** tab. The list of cryptospace endpoints added to CCKM from Google Cloud KMS through a VPC connection is displayed. The tab displays the following details:

Field	Description
Google Key Name	Name of the Google key.
Project	Name of the project.
Location	Location of a cryptospace.
Key Type	EKM Key type. Options are either symmetric or asymmetric.
CryptoSpace	Cryptospace name.
CryptoSpace Type	Cryptospace type. There are two types of cryptospaces. One for EKM endpoints (ekm) and another for EKM UDE endpoints (ekm-ude).
Versions	Identifier of the latest version of the key. The latest version number will be the number of versions that the key contains.
Key Ring	The GCP Cloud KMS Key ring name corresponding to this cryptospace endpoint.
Google Resource Name	The name of the key that includes the full path to the resource.
Algorithm	EKM Key Algorithm. Supported values are EXTERNAL_SYMMETRIC_ENCRYPTION(AES256) , RSA_SIGN_PSS_2048_SHA256, RSA_SIGN_PSS_3072_SHA256, RSA_SIGN_PSS_4096_SHA256, RSA_SIGN_PSS_4096_SHA512, RSA_SIGN_PKCS1_2048_SHA256, RSA_SIGN_PKCS1_3072_SHA256, RSA_SIGN_PKCS1_4096_SHA256, RSA_SIGN_PKCS1_4096_SHA512, EC_SIGN_P256_SHA256,EC_SIGN_P384_SHA38

To view the custom columns, click the Customize View () icon, select the desired option, and click **OK** to display the column.

Viewing details of an EKM cryptospace endpoint

To view the details of a Google EKM cryptospace endpoints on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > Google Cloud EKM**. The Google Cloud External Key Manager page displays.

3. Click the **Cryptospaces Endpoints** tab. The list of cryptospace endpoints added to CCKM from Google Cloud KMS through a VPC connection is displayed.
4. On the **Cryptospaces Endpoints** tab, click the **Name** link of the desired cryptospace endpoint.

Alternatively, click the overflow icon () corresponding to the desired cryptospace endpoint, and click **View/Edit**.

The edit view of the CryptoSpace Endpoints page shows additional details of the selected endpoint under the **CRYPTOSPACE ENDPOINT POLICY** and **KEY VERSIONS** sections. Note that **CRYPTOSPACE ENDPOINT POLICY** is uneditable.

Enabling or disabling a key version of an EKM cryptospace endpoint

From the edit view CryptoSpace Endpoints page, you can enable or disable a key version of a cryptospace endpoint in the **KEY VERSIONS** section.

To enable or disable a key version of a cryptospace endpoint on CCKM:

1. Under the **KEY VERSIONS** section, click the overflow icon () corresponding to the desired key version.
2. Select **Enable** or **Disable** depending on the current configuration of the key version.

Google Workspace CSE Resources

Client Side Encryption (CSE) in Google Workspace enables encryption operations within the client, storing only encrypted data and encrypted keys in the cloud. AES symmetric keys are used to encrypt the user content.

Google Workspace CSE allows the users to secure:

- Calls over Google Meet
- Docs, Sheets, and Slides data inside a Drive
- Google Calendar events
- Gmail (Google email) messages

Content is secured with an external encryption key that Google servers cannot access. File and call contents are encrypted on the browser before being sent to Google servers for storage.

With Google Workspace CSE:

- Key ACL Service (KACLS) controls the top-level encryption keys that protect users' Gmail messages, Meet Calls, Calendar event data, and Google Drive (Docs, Sheets, and Slides data). KACLS is also referred to as external key management service in this document.
- The authorization to encrypt or decrypt an object is created by Google Workspace.
- End-user authentication is provided by a third-party identity provider.

- Only encrypted data and encrypted keys are sent to Google Workspace servers.

Note

X25519MLKEM768 key exchange algorithm is supported for TLS 1.3, as a quantum resilience cipher for Google Workspace CSE connections between Chrome and CipherTrust Manager/CCKM. You need to enable the PQC algorithm in Chrome flags and CipherTrust Manager. Refer to [TLS Key Exchange Groups](#) for more details.

Access Policies

Access to encryption keys is controlled using Google Workspace [perimeters](#). These are optional additional checks performed on the authentication and authorization tokens within the KACLS.

Access control can be used to:

- Only allow users denoted in the permitted list of domains to decrypt keys.
- Deny users, for example, customers' Google Workspace administrators.
- Provide advanced restrictions, for example:
 - Time-based restrictions: Is the employee on call, on vacation, and so on.
 - Geolocation restrictions: Prevent access from specific locations or networks.
 - User role/type based access as asserted by the third-party identity providers.

Refer to [Sample Custom Policies](#) for different scenarios.

High Level Architecture

The Google Workspace CSE library enables encryption operations within the client, storing only encrypted data and encrypted keys on the Google Workspace servers. End user authentication is provided by a third-party identity provider.

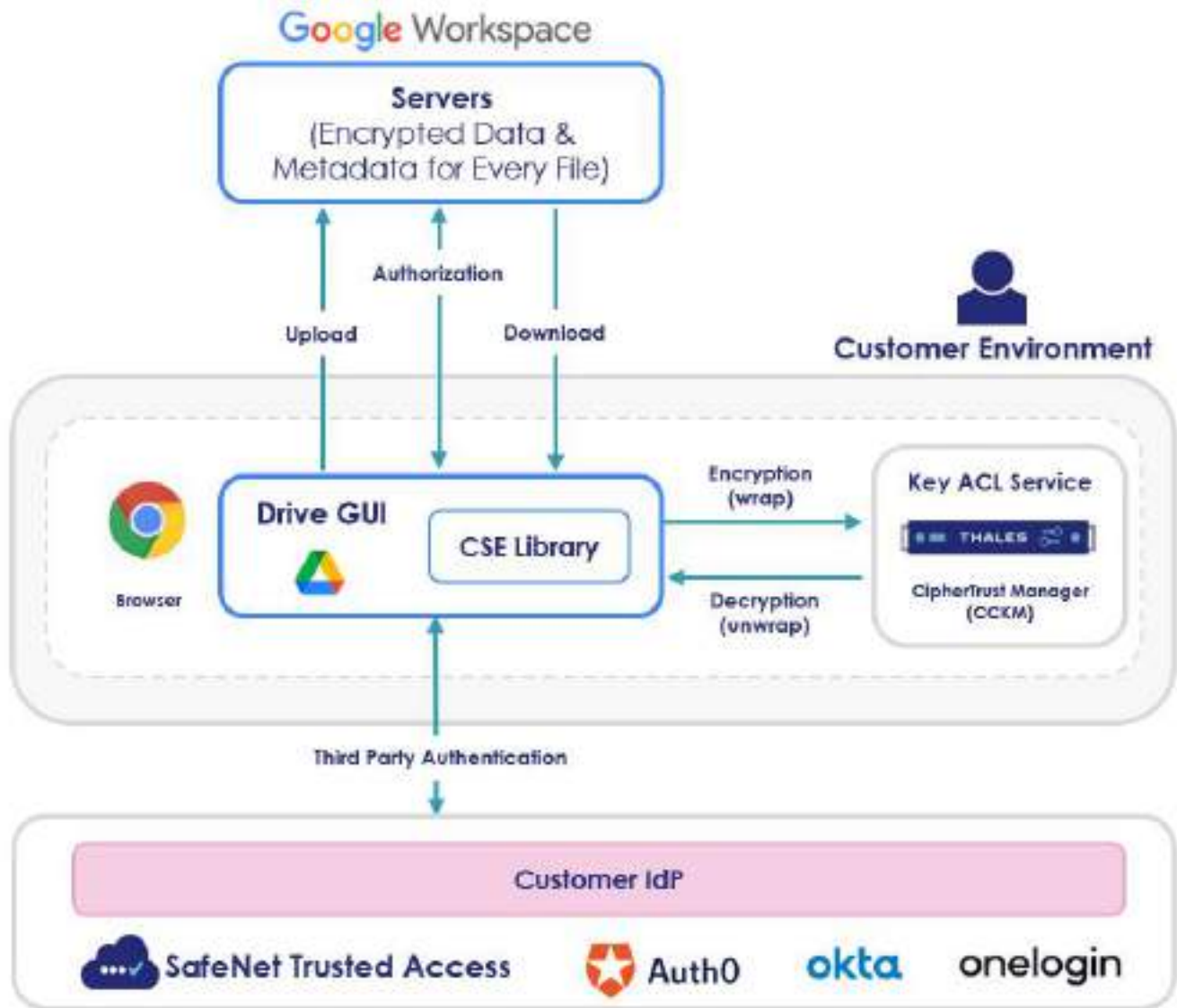
Google Workspace CSE allows the users to secure:

- Calls over Google Meet
- Docs, Sheets, and Slides data inside a Drive
- Google Calendar events
- Gmail (Google email) messages

Content is secured with an external encryption key maintained by the KACLS.

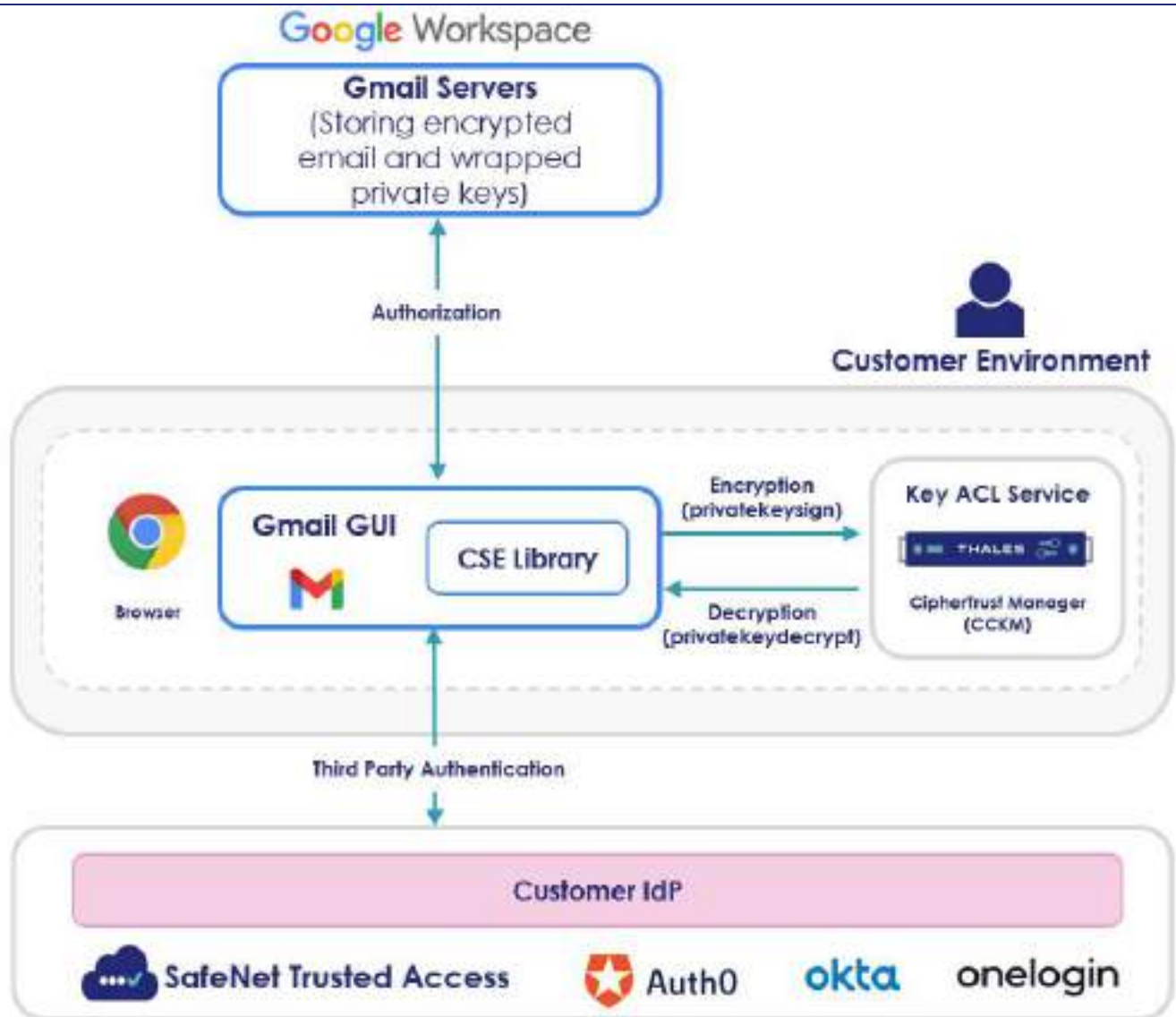
High Level Architecture

The following diagram shows the high level architecture:



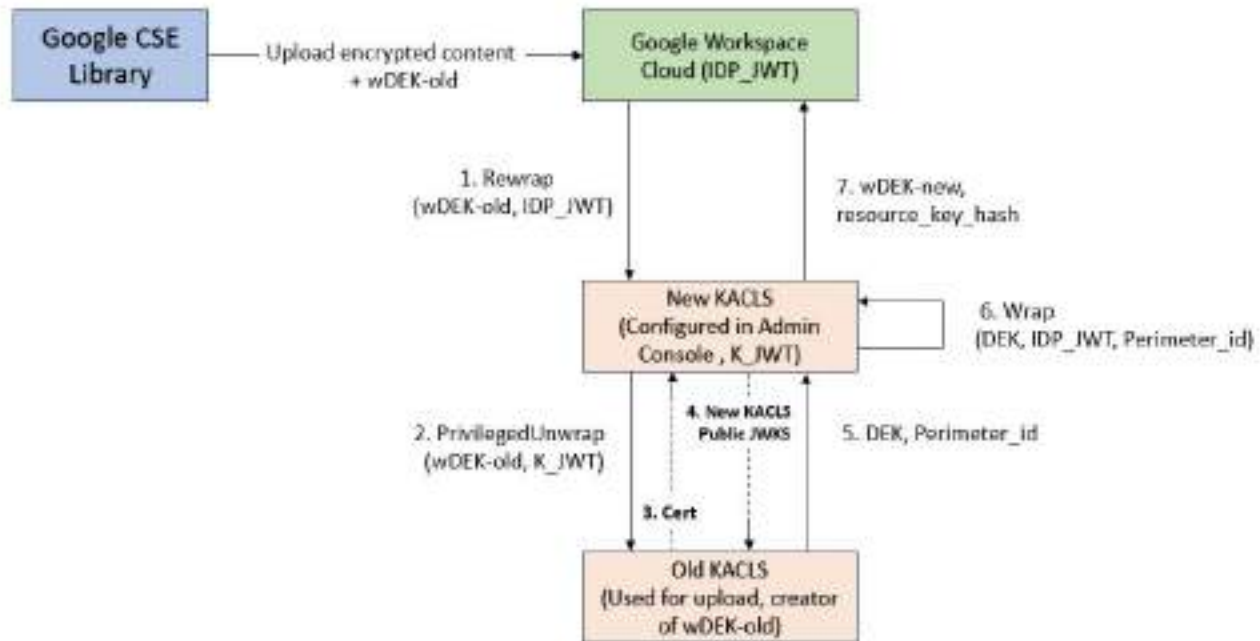
Architecture for Gmail

The following diagram shows the high level architecture for Google Workspace CSE for **Gmail**:



Architecture for Key Migration

The following diagram shows the Google CSE key migration workflow:



Licensing

Thales provides encryption of Google Workspace CSE endpoints through CCKM licensing models. Google Workspace consumes one CCKM license (cloud unit) per endpoint.

The number of Google Workspace endpoints that can be added to the CipherTrust Manager is limited by the number of licenses purchased. For example, if a 1-year license is purchased for 10 cloud units, then only 10 Google Workspace endpoints can be added to the CipherTrust Manager.

An archived or deleted Google Workspace endpoint releases one cloud unit of the CCKM license. It means, you can add another licensed CCKM cloud unit to the CipherTrust Manager. A recovered, enabled, or disabled Google Workspace endpoint consumes one cloud unit of the CCKM license.

Refer to [CCKM Licensing](#) for details.

Prerequisites

Common Prerequisites

Before you can perform client side encryption:

1. Make sure you have a stable release of Google Chrome or Microsoft Edge installed and running.
2. Make sure your access to KACLs is not blocked by web filters, for example, Zscaler.
3. Make sure you have the **CCKM Admins** rights to perform Google Workspace CSE operations on the CipherTrust Manager.

4. Make sure that an identity provider system is set up correctly. For this, the identity provider admin uses either of following methods:

Using .well-known File Configuration

- a. Host the `cse-configuration` JSON file on a Web server. A sample JSON file looks like this:

```
{
  "name": "CSE IDP",
  "client_id": "<authenticationAud>",
  "discovery_uri": "<openidConfigurationURL>",
  "audience": "cse-test"
}
```

Here,

- `<authenticationAud>` is the ID of the third-party identity provider. For example, for Auth0, it is represented by the `Client ID`.
 - `<openidConfigurationURL>` is the identity provider configuration URL. For example, for Auth0, it can be `https://demo.auth0.com/.well-known/openid-configuration`.
- b. Create a sub-domain on the Google domain portal for the hosted Web server. Navigate to the **Google domain > DNS > Customer resource records** and add the IP address of the Web server with the name `cse`.

Using IdP Fallback Settings

- a. On the Google Admin Console, set identity provider configuration.
 - b. Specify the following fields:
 - **Name:** Name for the identity provider.
 - **Client ID:** The ID of the third-party identity provider. For example, for Auth0 and STA, it is represented by the `Client ID`.
 - **Discovery URI:** The identity provider configuration URL. For example, for Auth0, it can be `https://demo.auth0.com/.well-known/openid-configuration`, and for STA, it is represented by **WELL KNOWN CONFIGURATION URL**.
 - **Grant type:** Set as **Implicit**.
 - c. Test and save the settings.
5. (Applicable when a valid public DNS for the CipherTrust Manager is unavailable.) Create a subdomain for Thales key service (KACLS) on Google domain. Refer to [Creating a Subdomain on Google Domain](#).

6. Create a URL to access the KACLS. This URL is referred to as KACLS Endpoint URL in this document. Refer to [Creating a KACLS Endpoint URL](#) below.
7. Configure Google Workspace connection to KACLS. Refer to [Configure Google Workspace Connection to KACLS](#) for details.

Creating a Subdomain on Google Domain

Note

This section is applicable when a valid public DNS for the CipherTrust Manager is unavailable.

1. Log on to [Google domain](#) as a super admin for the user domain.
2. Navigate to **DNS > Custom resource records**.
3. Create a subdomain for the KACLS. Specify a name for your subdomain and the IP address or hostname of the KACLS.

Tip

If you are working in a clustered CipherTrust Manager environment, you need to create a clustered subdomain on Google domain. Refer to [Create a clustered subdomain on Google domain](#) for details.

Note

It is recommended to [add the Google NTP server](#) to the CipherTrust Manager (**Admin Settings > NTPs**).

Creating a KACLS Endpoint URL

A KACLS URL is needed to access the Thales key service. Google Workspace administrators use this URL to configure Google Workspace to communicate with the KACLS. Creating a KACLS URL requires an identity provider and a KACLS endpoint.

To create a KACLS endpoint URL:

1. **Create an identity provider.**
 - **GUI:** Refer to [Creating Identity Providers](#).
 - **API:** Refer to [Creating Identity Providers](#).

2. Create a KACLS endpoint.

- **GUI:** Refer to [Creating KACLS Endpoints](#).
- **API:** Refer to [Creating KACLS Endpoints](#).

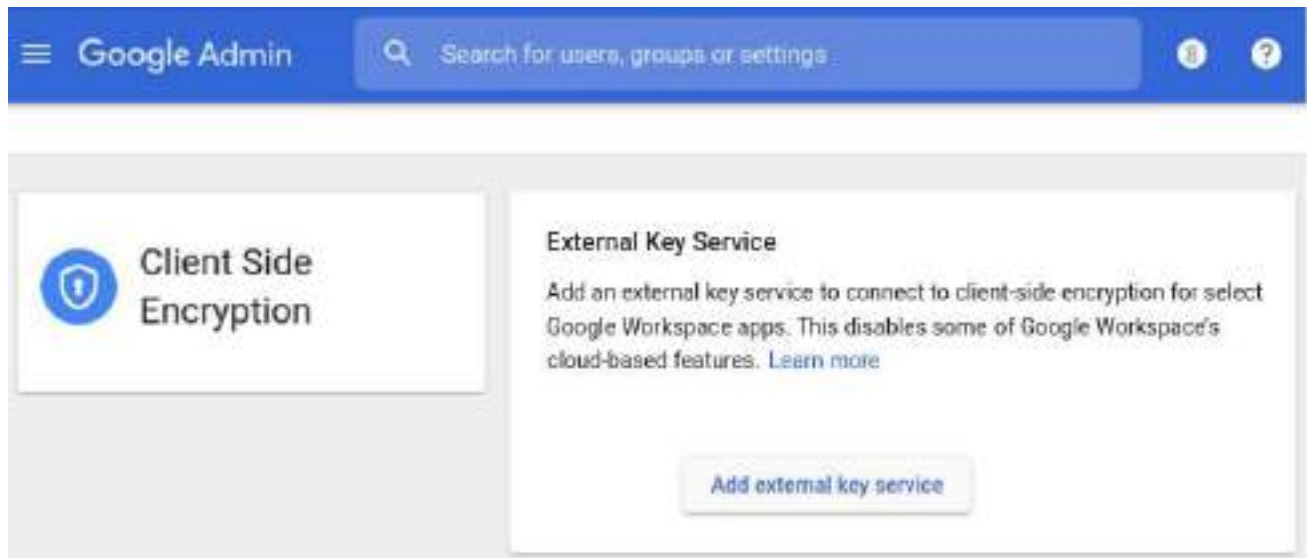
Note

Before proceeding, make sure that the KACLS endpoint URL is accessible from the internet and KACLS is running.

Configure Google Workspace Connection to KACLS

To configure the Google Workspace connection to KACLS:

1. Open the **Google Admin** console.
2. Log on to the user domain as a super admin.
3. Navigate to **Data > Compliance > Client-Side Encryption**.



4. Click **Add external key service**.
5. Specify **Name of external key service**. This name will appear in error messages if Google Workspace cannot contact the key service.
6. Enter **URL of external key service**. This URL was created in [Creating a KACLS Endpoint URL](#).

Edit external key service

External Key Service

Add an external key service to connect to client-side encryption for select Google Workspace apps. This disables some of Google Workspace's cloud-based features. [Learn more](#)

Name of external key service
Thales

URL of external key service
https://beta-demo.thalesgsuiteintegration.net/api/v1/cckm/GoogleWorkspaceCSE/endpoints/077a6404

TEST CONNECTION

CANCEL CONTINUE

7. Click **TEST CONNECTION** to test that Google Workspace can communicate with the KACLS.
If the connection fails, correct the KACLS endpoint URL, ensure the Internet connectivity, and retry.
8. Click **CONTINUE**.
9. Click **SAVE**. The Google Workspace connection to KACLS is configured.

Configure KACLS at the Domain and/or Organization Units Level

To configure KACLS at the domain and/or organization units level:

Note

The Google Workspace admin can manage organizational units by navigating to **Directory > Organizational Units**.

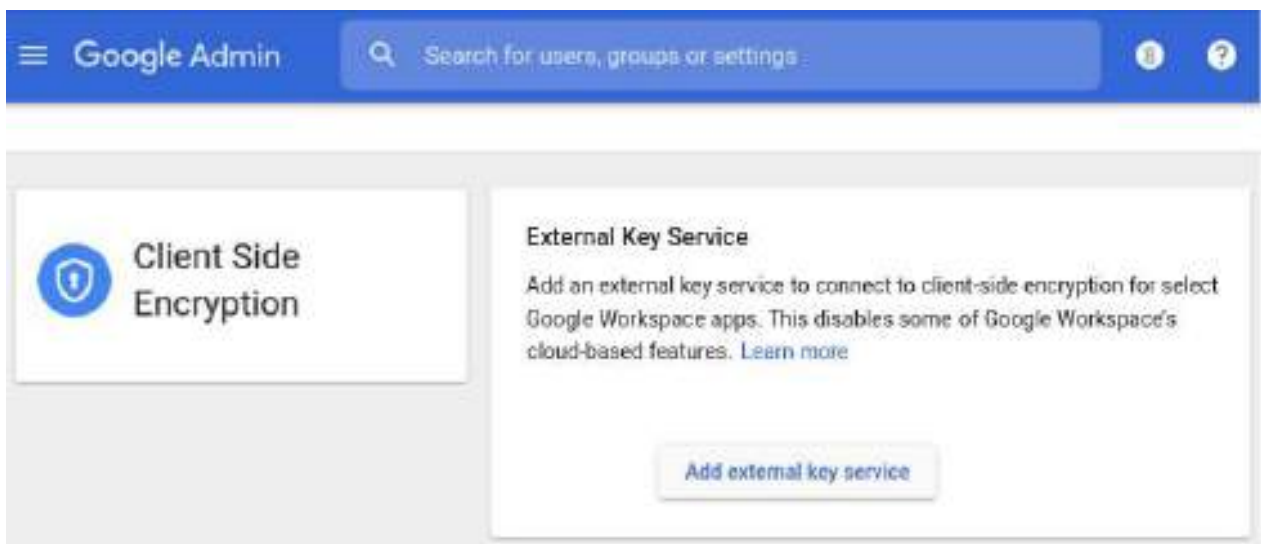
1. Open the [Google Admin](#) console.
2. Log on to the user domain as a super admin.
3. Navigate to **Data > Compliance > Client-Side Encryption**.
4. Click **Assign**.

5. Select the **Domain/OU** that you want to assign the KACLS to.
6. Select the key service from the list.
7. Click **Save**.

Additional Prerequisites for Gmail

Apart from the prerequisites described above, perform these additional steps for Gmail:

1. Create an environment and enroll it with Google.
2. Enable Google Workspace CSE for intended Gmail users (senders and recipients).
 - a. Open the **Google Admin** console, <http://admin.google.com>.
 - b. Log on to the user domain as a super admin.
 - c. Navigate to **CSE settings: Data > Compliance > Client-Side Encryption**.



- d. Scroll down to the Apps section and click the **Gmail** link.
 - e. Select an organizational unit or group for which you want to enable Gmail CSE.
 - f. Under **User access**, select **ON**.
 - g. Save the settings.
3. Prepare your certificates.
 - a. Generate S/MIME certificates. Adhere to the Google-specified [certificate chain rules](#).
 - b. Wrap associated private keys using your KACLS endpoint URLs. Refer to [Encrypting Private Keys \(wrapprivatekey\)](#) for details.

-
- c. Upload the wrapped private keys and certificates to Google.
 - a. **Using Google Workspace Admin Console:** Use this method if you want Google to trust your own root CA.
 - a. Open the **Google Admin** console, <http://admin.google.com>.
 - b. Log on to the user domain as a super admin.
 - c. Navigate to **CSE settings: Apps > Google Workspace > Gmail**.
 - d. On the right, click **User settings**.
 - e. Click **S/MIME** and select **Enable S/MIME encryption for sending and receiving emails** and **Allow users to upload their own certificates**.
 - f. Click **ADD** to upload the root certificate. This root CA is the chain of intermediate and root certificates.
 - g. Save the changes.
 - b. **Using Gmail API client libraries:** Use this method to upload each user's S/MIME certificates and wrapped private keys using the API client libraries provided by Google.

Additional Prerequisites for Key Migration

Browser Requirement

The latest stable version of Google Chrome is installed and running.

Configure External Key Management (EKM) Service

To configure the external key management service.

1. Create a sub-domain for KACLS in the Google domain. Refer to [Creating a subdomain on Google domain](#) for details.
2. Create a KACLS Endpoint URL, refer to [Creating a KACLS Endpoint URL](#).

Configure Migration on KACLS

Using the API, you need to update the rewrap and privilegedUnwrap configurations on the following.

1. Old KACL
2. New KACL

Refer to the [Update a Rewrap Configuration](#) and [Update a Privileged Unwrap Configuration](#) APIs.

Configure Google Workspace Connection to KACLS

Refer to [Configure Google Workspace Connection to KACLS](#) for details.

Note

- To enable the key migration feature, you need to add an EKM service with backup enabled.
- You can only have one backup EKM service enabled at a time.

External key service

Name
test-migration-new

URL
https://test-migration-new.cckm-thalescpl.com/api/v1/cckm/GoogleWorkspaceCSE/endpoints/e9ca22f1-cc7e-4efa-86a0-9cb3d30790a7

Backup external key service
test-migration-old

EDIT **DISABLE & USE BACKUP** **REMOVE BACKUP**

- Ensure to turn on the "migration" toggle.

Additional Prerequisites for CSE Guest Access

Browser Requirement

The latest stable version of Google Chrome is installed and running.

Configure External Key Management (EKM) Service

To configure the external key management service.

1. Create a sub-domain for KACLS in the Google domain. Refer to [Creating a subdomain on Google Domain](#) for details.
2. Create an internal Identity Provider and Guest access Identity Provider. Refer to [Creating an Issuer](#).
3. Create a KACLS endpoint by adding both the Identity providers (internal and Guest) and enabling guest access. Refer to [Creating a KACLS Endpoint URL](#).

4. Validate KACLS Endpoint URL is accessible from the internet and KACLS is running.
5. Users will be given an Endpoint URL to access our key service (KACLS). The users will need this URL to configure Google Workspace to contact KACLS (CipherTrust Manager).

Additional Configurations for Guest Access on Google Admin Console

- The users need to configure Google Workspace Connection to KACLS. Refer to [Configure Google Workspace Connection to KACLS](#) for details.
- Configure a Guest access Identity Provider.

Note

Guest access configuration will require two client IDs per application(AUTHN and AUTHN2 above), used by the KACLS and Meet/Drive/Gmail backend respectively.

Additional Prerequisites for Bulk Import

To perform a bulk import, you must:

- Create a Google Cloud project to use Google Workspace APIs. For more information, see [Create a Google Cloud Project](#) for details.
- Enable the Google Drive API to view the list of active APIs. For more information, see [View & turn off Google Workspace APIs](#).
- Configure Domain Wide Delegation with your service account details. For more information, see [Enable Domain Wide Delegation](#).
- Upload your local files to Google Drive as client-side encrypted (CSE) files.

For more information on prerequisites for this feature, see [Drive CSE Upload GitHub folder](#).

Communication with KACLS

Google Workspace can communicate with KACLS for encryption and decryption of data, as described below.

Google Drive

Google Workspace can communicate with KACLS for encryption and decryption of files on Google Drive. Whenever a new file (**Blank encrypted document**) is created or updated, its data is encrypted automatically.

Also, when a new file is uploaded (using **Drive > File upload > Encrypt and upload**), the file is encrypted and uploaded. When an existing encrypted file is opened, its data is decrypted for authenticated end users.

Google Meet

Google Workspace can communicate with KACLS for encryption and decryption of calls over Google Meet (using **Meet > New meeting > Video call options > Security > Add encryption**).

Whenever an authenticated host initiates an encrypted call, the call data is encrypted automatically. The call data is automatically decrypted for authenticated participants.

Google Calendar

Google Workspace can communicate with KACLS for encryption and decryption of Google Calendar event data such as description, attachments, etc. (using **Calendar > Create**). Turn on the encryption toggle.

Whenever an authenticated host creates a Calendar event, the event description and attachments are encrypted automatically. The event data is automatically decrypted for authenticated recipients.

While creating the event, you can also add an encrypted Google Meet call by clicking **Add Google Meet video conferencing**.

Gmail

Google Workspace can communicate with KACLS for encryption and decryption of Gmail messages. (using **Gmail > Compose**). Click the **Message security** icon and **Turn on** the encryption.

Whenever an authenticated user composes an email message, the message details are encrypted automatically. The encrypted Gmail message is automatically decrypted for authenticated recipients.

If a key pair already exists, a new key pair cannot be inserted. To insert a new key pair, the existing key pair must first be deleted. The existing key pair can only be deleted if it is disabled for more than 30 days.

CSE Guest Access Support for Google Meet and Google Drive

Client-side encryption in Google Workspace enables encryption operations within the client, storing only encrypted files and encrypted keys in the cloud. AES symmetric keys are used to encrypt the user content. Google Workspace Client-side encryption (CSE) allows the user to secure Gmail messages, Meet Calls, Calendar event data, and Google Drive (Docs, Sheets, and Slides data) with an external encryption key that Google servers cannot access.

File contents are encrypted on the browser before being sent to Google servers for storage. With Google Workspace CSE, key ACL service (KACLS) controls the top-level encryption keys that protect users' Drive, Docs, Sheets, and Slides data.

Google Meet CSE provides a way to have end-to-end encrypted meetings by allowing enterprise workspace customers to use their own Identity Provider (IdP) and KACLS. Using their own IdP and KACLS infrastructure, users from a customer domain can participate in end-to-end encrypted meetings. Unfortunately, external accounts that do not belong to the customer domain, and cannot be authenticated using the internal IdP used by the customer, don't have a way to join CSE meetings which limits the usefulness of the CSE feature in its current form.

This problem will be addressed by allowing guest accounts that are authenticated using additional IdPs like Microsoft Azure ID, Google's Identity platforms or an alternative guest IdP provided by the host domain. The solution for Guest Access uses two ID tokens. This allows for the simultaneous operation of a primary IdP service while using a separate service to authenticate external guests.

Eligibility for Guest Accounts

In this feature, you will add support for external guests (outside the host's domain) to join Google Meet CSE calls. In this context, guest accounts are external participants that do not have:

- Any Google Account, any account registered with Google's identity service, regardless of actual domain or account license type (free or paid).
- A Google consumer account.
- A Google account that doesn't belong to the organizer's domain.

These accounts will instead rely on external IdP to access Google Meet to join calls such as "Sign in with Microsoft" and "Sign in with Google".

CSE Guest Access Workflow

A New CSE Join Flow for CSE Guest Accounts

This feature introduces a new join flow for anonymous CSE guest accounts. It allows an initially anonymous user to authenticate with a guest IdP and thereby establish an identity that Google can use to get access to the KACLs server (to encrypt/decrypt media) and authorize access to the meeting. Authenticated and unauthenticated Google users not logged in to the host domain's IdP are treated equally.

Allowed IdP Configurations are Provided when Loading the Meet page

Currently, an anonymous or non-invited user is directly rejected when loading a CSE meeting. This will now change to allow the user to authenticate with the guest IdP.

Upon loading the green room a guest account is presented with the IdP selection screen. In it, the user is challenged to authenticate using a guest IdP as configured by the meeting host organization.

The host organization name is displayed to the guest, and the guest is warned that they will be redirected to a third-party login page where they need to authenticate with the same email address as they were invited.

New Field Introduced in the Authorization Token

The Authorization token (AuthZ) generated by Google Drive (docs/slides/sheets) and meet is extended with a new claim "email_type". The possible values for the claim are:

- **google**: This email belongs to a Google account.
- **google-visitor**: This email does not belong to a Google account, but was PIN-code verified by Google. It is associated with guest access for Google Drive.
- **customer-idp**: This email does not belong to a Google Account, but the user's email was extracted using a customer-configured IdP. It is associated with guest access for Google Meet.

This claim may be unset. In this case, the default value is **google**.

Token Matching Logic

The email_type claim is guaranteed to be present when Google issues an AuthZ token for an email that does not correspond to a Google Account. This is introduced as a part of the "Guest Access" feature.

If a customer does not have Guest Access configured, all requests where email_type is set to `google-visitor` or `customer-idp` will be rejected. Requests with an email type of `google` or with an unset email type should continue to be accepted.

Google CSE support for Google's end-to-end encrypted email

Google Workspace can use the Key ACL Service (KACLS) to encrypt and decrypt Gmail messages. Google's end-to-end encrypted email ("Send to anyone") allows enterprise Gmail users to send end-to-end encrypted messages to any email address, without needing to configure S/MIME certificates for users.

Note

Google's end-to-end encrypted email is generally available (GA) as of October 2025. The GA release includes full support for the following scenarios:

- Sending emails to other Google Workspace domains
- Sending emails to @gmail.com consumer accounts
- Setting up a guest Identity Provider (IdP) for external recipients
- Sending emails to guest recipients

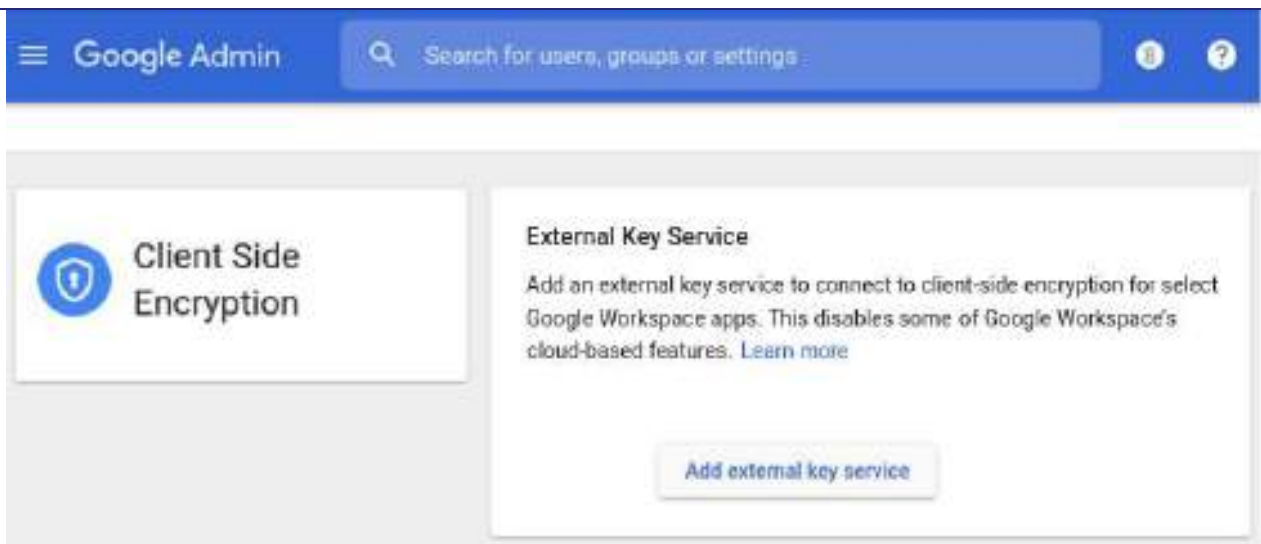
CipherTrust Manager version 2.22 and later will fully support all features introduced in Google's GA release.

Before enabling this client-side encryption, complete the prerequisites.

Prerequisites

The prerequisites include the following.

1. Create an environment and enroll it with Google.
2. Enable Google Workspace CSE for intended Gmail users (senders and recipients).
 - a. Open the **Google Admin** console, <http://admin.google.com>.
 - b. Log on to the user domain as a super admin.
 - c. Navigate to **CSE settings: Data > Compliance > Client-Side Encryption**.



- d. Scroll down to the Apps section and click the **Gmail** link.
 - e. Select an organizational unit or group for which you want to enable Gmail CSE.
 - f. Under **User access**, select **ON**.
 - g. Save the settings.
3. The Assured Controls or Assured Controls Plus add-on is required. End-to-end encrypted email is only available when hardware key encryption is not used.

Enable Send to anyone

To enable "Send to anyone" from the Google Admin console:

1. Open the [Google Admin](#) console.
2. Log on to the user domain as a super admin.
3. Navigate to **Data > Compliance > Client-Side Encryption > Gmail**.
4. Under **Encryption with guest accounts**, click the **Edit** icon.
5. Enable **Allow users to send client-side encrypted messages to recipients who aren't using S/MIME**.
6. Click **Save**.

Note

External recipients are granted guest accounts. These accounts:

- Reside within a dedicated organizational unit (OU) or group.
- Are fully owned by the customer's organization.
- Must adhere to the customer's organization's policies.

Admin controls allow Gmail users to access their accounts.

To enable "Send to anyone" for external Gmail domains and guests on CCKM:

1. When creating a new KACLS endpoint or updating an existing one, ensure to set the value of `allow_guest_access` to true. Refer to [Creating KACLS Endpoints](#) and [Updating a KACLS Endpoint](#) for details.
2. Set the value of the `authenticationAud` parameter to the Client ID of the Guest Identity Provider that has been configured in your Google Admin Console. Refer to [Additional Configurations for Guest Access on Google Admin Console](#) for details.

Clustering for Google Workspace

The CipherTrust Manager can be configured to support High Availability. When clustered, all CipherTrust Manager appliances in the cluster (called cluster nodes or nodes) will continuously synchronize their databases with each other. Any appliance in the cluster may be contacted for Google Workspace operations.

1. Create a CipherTrust Manager cluster. Refer to [Clusters and Nodes](#) for information on creating a CipherTrust Manager Cluster.
2. Create a clustered subdomain on Google domain:
 - a. Log on to [Google domain](#) as a super admin for the user domain.
 - b. Navigate to **DNS > Custom resource records**.
 - c. Create a subdomain for the KACLS cluster. Specify a name for your subdomain and the IP addresses or hostnames of all the clustered KACLS nodes.

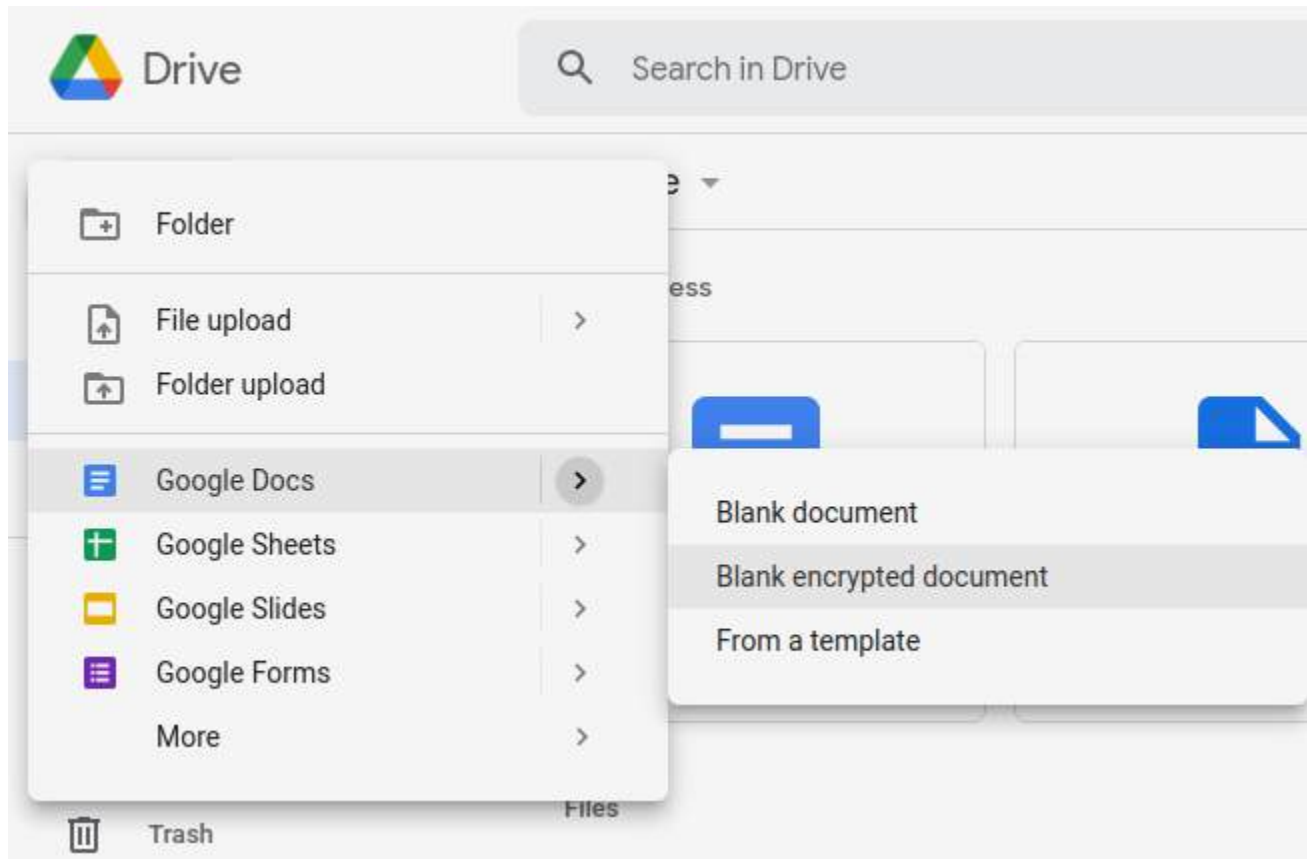
Encrypting Data Encryption Keys

When a user opens a blank encrypted document, initiates an encrypted call, or creates an encrypted Calendar event, a random Data Encryption Key (DEK) is generated and the data is encrypted with it. After the third-party identity provider user authentication is successful, an authentication token (3P JWT) is generated. The user is authenticated using a Google JWT and third-party JWT. KACLS wraps the DEK with the key associated with the KACLS endpoint URL and returns a wrapped key. The encrypted data with its wrapped key is uploaded to the Google Workspace server.

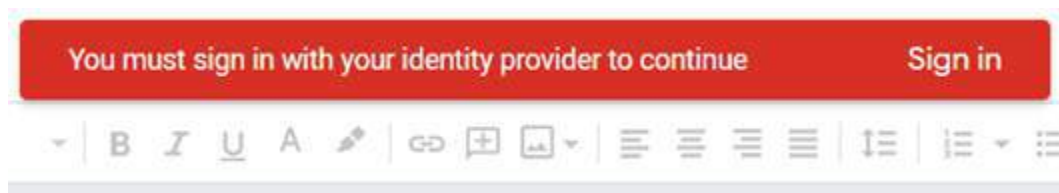
Encrypting Files on Google Drive

To create an encrypted Google Docs document:

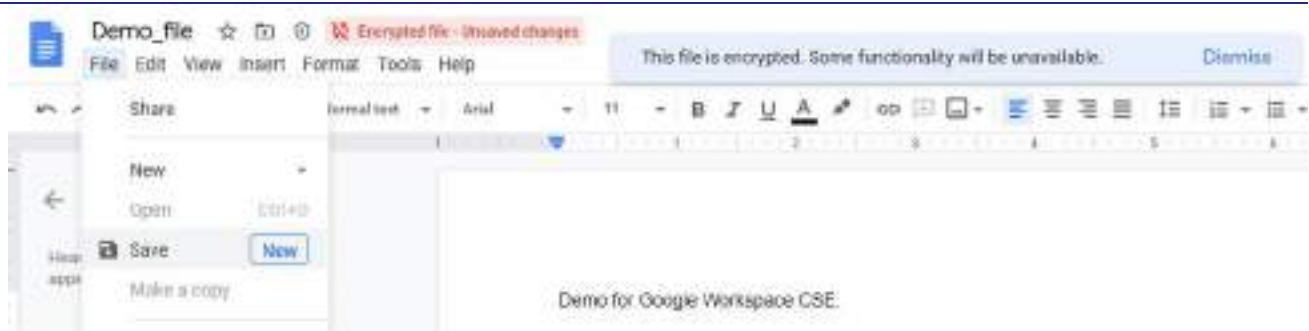
1. Open the **Google Drive** console, <https://drive.google.com>.
2. Log on as an end user.
3. Click **New > Google Docs > Blank encrypted document**.



A message prompting to sign in with your identity provider is displayed, as shown below.



4. Sign in with the configured third-party identity provider's user credentials.
5. Make your changes and save the document, as shown below.



The document is encrypted. KACLS' `wrap` API is called to encrypt the document. The `wrap` requests are logged at KACLS (under **Records > Server Records**), as shown below.

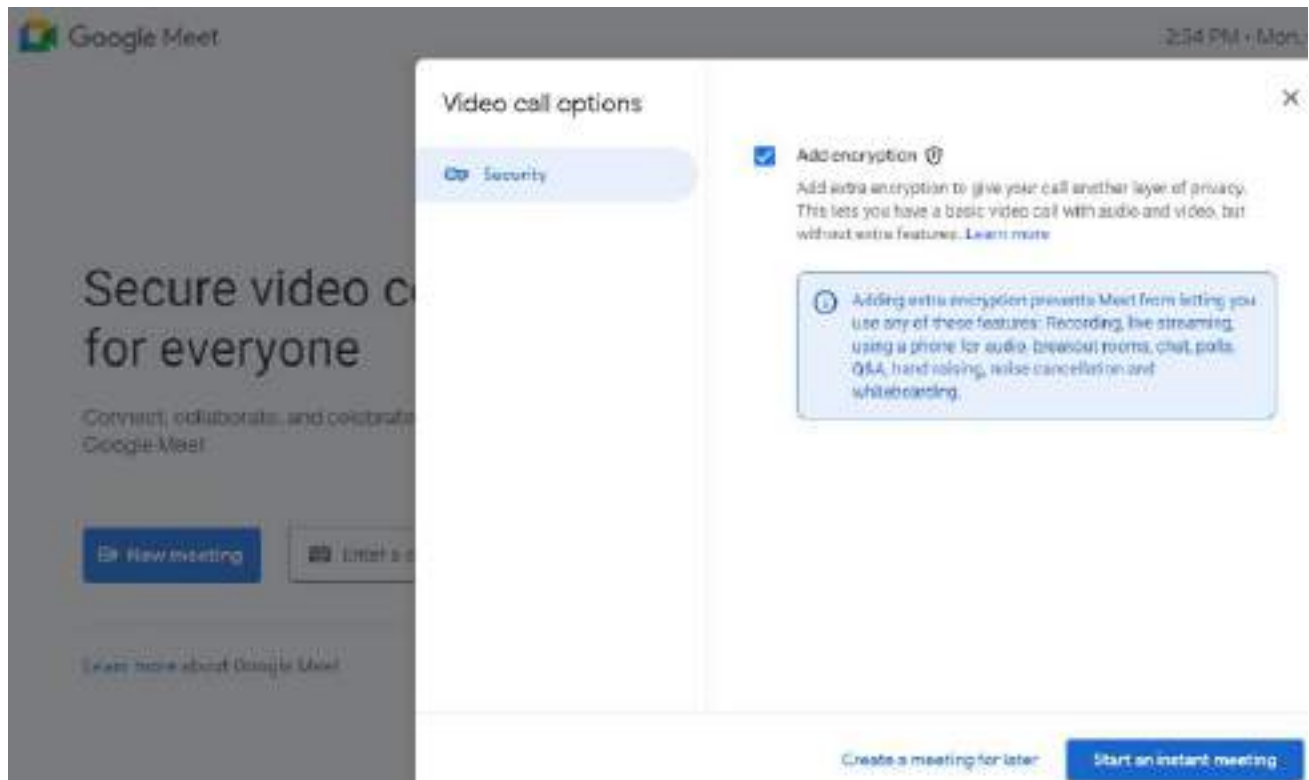
```
{
  "reason": "",
  "authorization": {
    "aud": "cse-authorization",
    "exp": 1643373456,
    "iat": 1643369856,
    "iss": "gsuitecse-tokenissuer-drive@system.gserviceaccount.com",
    "role": "writer",
    "email": "demo.user@domain.com",
    "kacls_url": "https://demo.domain.com/api/v1/cckm/GoogleWorkspaceCSE/endp
oints/fb692716-b749-47b5-a233-3e0e06a9ee30",
    "perimeter_id": "",
    "resource_name": "//googleapis.com/drive/files/1c-5qu5usfHFxeyo0FtlKBNUHH
xZHmXan"
  },
  "authentication": {
    "acr": "1",
    "aud": "b0ae52b6-c5e1-4931-9091-34d09df11960",
    "azp": "b0ae52b6-c5e1-4931-9091-34d09df11960",
    "exp": 1643370765,
    "iat": 1643369865,
    "iss": "https://<IDP>",
    "jti": "129cf52c-4394-4bc8-9fb2-d3e8c812d017",
    "sub": "6ba78312-6214-3ac5-b2a2-b99d83681091",
    "typ": "ID",
    "email": "demo.user@domain.com",
    "nonce": "fzmCUX-d6R74GQpdyLWvzA:https://docs.google.com",
    "s_hash": "hTpM40xvzcDvbGKKzzq9pg",
    "auth_time": 1643369865,
    "session_state": "95ae9ecc-d20b-4d44-bf4b-a485d8ff6773",
    "email_verified": false
  }
}
```

Encrypting Calls Over Google Meet

To initiate an encrypted Google Meet call:

1. Open the **Google Meet** console, <https://meet.google.com/>.
2. Log on as an end user.

3. Click **New meeting > Video call options > Security**.
4. Select **Add encryption**.



5. Create an encrypted call. The options are:

- Create a meeting for later
- Start an instant meeting

A message prompting to sign in with your identity provider is displayed, as shown below.

Sign in to join

You must sign in with <IDP> to join this encrypted call

Cancel **Sign in**

6. Sign in with the configured third-party identity provider's user credentials.

The call data is encrypted automatically with KACLS' **wrap** API. The **wrap** requests are logged at KACLS (under **Records > Server Records**), as shown below.

```
{
  "reason": "Client-side encryption for Google Meet",
  "authorization": {
    "aud": "cse-authorization",
    "exp": 1646761146,
    "iat": 1646757546,
    "iss": "gsuitecse-tokenissuer-meet@system.gserviceaccount.com",
    "role": "writer",
    "email": "demo.user@domain.com",
    "kacals_url": "https://demo.domain.com/api/v1/cckm/GoogleWorkspaceCSE/endpoint
s/07cabf0f-e59d-426f-927e-f41827bacf5b",
    "perimeter_id": "",
    "resource_name": "//meetings.googleapis.com/MeetingSpace/spaces/tJgsmRfbjDoB"
  },
  "authentication": {
    "acr": "1",
    "aud": "6f088e98-d071-4d24-b3f1-8c86c0090f4a",
    "azp": "6f088e98-d071-4d24-b3f1-8c86c0090f4a",
    "exp": 1646758044,
    "iat": 1646757144,
    "iss": "https://<IDP>",
    "jti": "4c217728-b7f0-4bfe-ae7c-dee238b69929",
    "sub": "07c9e65b-bdca-3a2f-a390-4cb1502e6ae8",
    "typ": "ID",
    "email": "demo.user@domain.com",
    "nonce": "evrsDJS1_sc9xQdrDljbW:https://meet.google.com",
    "s_hash": "LcSs9u0M5fuV20HI1ykv9Q",
    "auth_time": 1646757144,
    "session_state": "1d87d04d-e300-4db2-a1a4-19825aaaf603",
    "email_verified": false
  }
}
```

Encrypting Google Calendar Events

To create an encrypted Google Calendar event:

1. Open the **Google Calendar** console, <https://calendar.google.com/>.
2. Log on as an end user.
3. Click **Create**.
4. Enable **Turn on encryption**. After selection, the option looks like the following.



5. Add event details.

A message prompting to sign in with your identity provider is displayed.

6. Sign in with the configured third-party identity provider's user credentials.

The event data is encrypted automatically with KACLS' `wrap` API. The `wrap` requests are logged at KACLS (under **Records** > **Server Records**), as shown below.

```
{
  "reason": "Encrypting description for calendar demo.user@domain.com, event 73
tg8i2jdio57rib0o4s23a2em",
  "authorization": {
    "aud": "cse-authorization",
    "exp": 1647936974,
    "iat": 1647933374,
    "iss": "gsuitecse-tokenissuer-calendar@system.gserviceaccount.com",
    "role": "writer",
    "email": "demo.user@domain.com",
    "kacsl_url": "https://demo.domain.com/api/v1/cckm/GoogleWorkspaceCSE/endp
oints/7232123d-3d0d-4d5b-8b14-24c97540708e",
    "perimeter_id": "",
    "resource_name": "//googleapis.com/calendar/08927975989983541514/eef39626
6e4b5ca9"
  },
  "authentication": {
    "aud": "ecdUQSQtIVkZ7rYSarnN45nuUZkeLUqL",
    "exp": 1647968855,
    "iat": 1647932855,
    "iss": "https://<IDP>",
    "sub": "google-oauth2|110478923717308119755",
    "email": "demo.user@domain.com",
    "nonce": "asTwdJJLSzWP2hgGTkmTXw:https://krahsc.google.com",
    "email_verified": true
  }
}
```

Encrypting Gmail Messages

When a user composes an encrypted Gmail message, the message will be encrypted by using the recipient's public key associated with the wrapped private key and certificates uploaded to Google. These were uploaded while completing the [additional prerequisites for Gmail](#).

After the third-party identity provider user authentication is successful, an authentication token (3P JWT) is generated. The user is authenticated using a Google JWT and third-party JWT. When the sender composes the message, KACLS unwraps the sender's wrapped private key, signs the message digest with this private key, and returns the signed digest. The encrypted Gmail message is uploaded to the Google Workspace server.

To compose an encrypted Gmail message:

1. Open the **Gmail** console, <https://mail.google.com/>.
2. Log on as an end user.
3. Click **Compose**.
4. Click the **Message security** icon and **Turn on** the encryption.

New Message

To Cc Bcc

Subject

Your message is secure
Gmail protects your message during delivery with standard encryption. [Learn more](#)

Encryption by cckm-thalescpl.com
Protect sensitive information and comply with company policies. [Learn more](#)
[Turn on](#)

Send

Sans Serif

B *I* U A

☺ 📷 📍

5. Compose the message.

6. Click **Send**.

A message prompting to sign in with your identity provider is displayed.

7. Sign in with the configured third-party identity provider's user credentials.

The Gmail message is encrypted automatically with KACLS' `privatekeysign` API. The `privatekeysign` requests are logged at KACLS (under **Records** > **Server Records**), as shown below.

```
{
  "reason": "sign email",
  "authorization": {
    "aud": "cse-authorization",
    "exp": 1661487920,
```

```

    "iat": 1661484320,
    "iss": "gsuitecse-tokenissuer-gmail@system.gserviceaccount.com",
    "role": "signer",
    "email": "demo.user@domain.com",
    "kaccls_url": "https://demo.domain.com/api/v1/cckm/GoogleWorkspaceCSE/endp
oints/59e7c24f-4573-472d-8e8c-afb622793154",
    "spki_hash": "jFBT+M1baVaQqDRLmQTP1gS3zwrUPXpR/u0tyAnjbkY=",
    "message_id": "<CAGb6PCqtvd6d5n20xazMGkFxnzY1b9yBQPYmVgEdkM60=AmNQ@mai
l.gmail.com>",
    "perimeter_id": "",
    "resource_name": "//gmail.googleapis.com/gmail/users/demo.user%40domain.c
om/settings/cse/keypairs/ANe1BmhsTEntl0K99F0xA5Txc1jJgjUSxAms4RzSmRypRSQzh4nYso55
Mq9ktKmmPPNLP0Rgx0FLSQXhj6xdg6_OX-7hIg",
    "spki_hash_algorithm": "SHA-256"
  },
  "authentication": {
    "aud": "3kT5KerY3vXrEuzNRrQxVQqaARNvYUAWR",
    "exp": 1661485571,
    "iat": 1661484371,
    "iss": "https://<IDP>",
    "sid": "klqAk-BaWczEf_Unf6RoCM8DPwdXL7hU",
    "sub": "google-oauth2|110478923717308119755",
    "email": "demo.user@domain.com",
    "nonce": "LJOJ79acJoMBIqeTw9vHrg:https://krahsc.google.com",
    "email_verified": true
  }
}

```

Decrypting Data Encryption Keys

When a user opens an encrypted document, joins an encrypted call, or views an encrypted Calendar event, the wrapped key and the encrypted data are downloaded from the Google Workspace server. Google JWT is generated by the Google Workspace CSE library. After the third-party identity provider user authentication is successful, an authentication token (3P JWT) is generated. The user is authenticated using a Google JWT and third-party JWT. KACLS unwraps the wrapped key and returns the DEK. Google Workspace CSE decrypts the data using this DEK.

Decrypting Files on Google Drive

To decrypt an encrypted Google Docs document:

1. Open the **Google Drive** console, <https://drive.google.com>.
2. Log on as a super admin for the user domain.
3. Try to open the encrypted document (created in [Encrypting Data Encryption Keys](#)).
4. Sign in with the configured third-party identity provider user credentials.

KACLS' **unwrap** API is called internally to decrypt the document for the user. The **unwrap** requests are logged at KACLS (under **Records > Server Records**), as shown below.

```
{
```

```

"reason": "",
"authorization": {
  "aud": "cse-authorization",
  "exp": 1644323888,
  "iat": 1644320288,
  "iss": "gsuitecse-tokenissuer-drive@system.gserviceaccount.com",
  "role": "writer",
  "email": "demo.user@domain.com",
  "kaccls_url": "https://demo.domain.com/api/v1/cckm/GoogleWorkspaceCSE/endp
oints/a2feb391-6184-4b66-ad7f-1248d568375d",
  "perimeter_id": "",
  "resource_name": "//googleapis.com/drive/files/1g1rb0wvPRpIhHHJjbchgnZNSv
qII4TSp"
},
"authentication": {
  "acr": "1",
  "aud": "b0ae52b6-c5e1-4931-9091-34d09df11960",
  "azp": "b0ae52b6-c5e1-4931-9091-34d09df11960",
  "exp": 1644320477,
  "iat": 1644319577,
  "iss": "https://<IDP>",
  "jti": "bf7c35ba-67c7-4762-a945-72aeca382e1b",
  "sub": "6ba78312-6214-3ac5-b2a2-b99d83681091",
  "typ": "ID",
  "email": "demo.user@domain.com",
  "nonce": "LZ2oaPjdC052dAkY7LNTlw:https://docs.google.com",
  "s_hash": "7UIHrDHqf3sRbIRC-zZHTA",
  "auth_time": 1644319577,
  "session_state": "6b686726-bcc3-4d2d-9429-2ab80aad5eed",
  "email_verified": false
}
}

```

Decrypting Calls Over Google Meet

To decrypt an encrypted Google Meet call:

1. Open the **Google Meet** console, <https://meet.google.com/>.
2. Log on as an end user.
3. Join the call (created in [Encrypting Data Encryption Keys](#)).
4. Sign in with the configured third-party identity provider user credentials.

KACLS' **unwrap** API is called internally to decrypt the call for the user. The **unwrap** requests are logged at KACLS (under **Records > Server Records**), as shown below.

```

{
  "reason": "Client-side encryption for Google Meet",
  "authorization": {
    "aud": "cse-authorization",
    "exp": 1646761418,
    "iat": 1646757818,
    "iss": "gsuitecse-tokenissuer-meet@system.gserviceaccount.com",

```



```

    "role": "reader",
    "email": "demo.user@domain.com",
    "kacls_url": "https://demo.domain.com/api/v1/cckm/GoogleWorkspaceCSE/endpoints/07cabf0f-e59d-426f-927e-f41827bacf5b",
    "perimeter_id": "",
    "resource_name": "//meetings.googleapis.com/MeetingSpace/spaces/4Wbij56jcqAB",
  },
  "authentication": {
    "acr": "1",
    "aud": "6f088e98-d071-4d24-b3f1-8c86c0090f4a",
    "azp": "6f088e98-d071-4d24-b3f1-8c86c0090f4a",
    "exp": 1646758044,
    "iat": 1646757144,
    "iss": "https://<IDP>",
    "jti": "4c217728-b7f0-4bfe-ae7c-dee238b69929",
    "sub": "07c9e65b-bdca-3a2f-a390-4cb1502e6ae8",
    "typ": "ID",
    "email": "demo.user@domain.com",
    "nonce": "evrsDJS1_sc9xQdrDLjbnw:https://meet.google.com",
    "s_hash": "LcSs9u0M5fuV20HI1ykv9Q",
    "auth_time": 1646757144,
    "session_state": "1d87d04d-e300-4db2-a1a4-19825aaaf603",
    "email_verified": false
  }
}

```

Decrypting Google Calendar Events

To decrypt an encrypted Google Calendar event:

1. Open the **Google Calendar** console, <https://calendar.google.com/>.
2. Log on as an end user.
3. View the event (created in [Encrypting Data Encryption Keys](#)).
4. Sign in with the configured third-party identity provider user credentials.

KACLS' **unwrap** API is called internally to decrypt the event data for the user. The **unwrap** requests are logged at KACLS (under **Records > Server Records**), as shown below.

```

{
  "reason": "Decrypting description for calendar demo.user@domain.com, event 2c4nks2odk7nbk6nh0dpdhfn1p",
  "authorization": {
    "aud": "cse-authorization",
    "exp": 1647936814,
    "iat": 1647933214,
    "iss": "gsuitecse-tokenissuer-calendar@system.gserviceaccount.com",
    "role": "reader",
    "email": "demo.user@domain.com",
    "kacls_url": "https://demo.domain.com/api/v1/cckm/GoogleWorkspaceCSE/endpoints/7232123d-3d0d-4d5b-8b14-24c97540708e",
    "perimeter_id": "",
  }
}

```

```

    "resource_name": "//googleapis.com/calendar/08927975989983541514/d05a582b
b76c2af"
  },
  "authentication": {
    "aud": "ecdUQSQtIVkZ7rYSarnN45nuUZkeLUqL",
    "exp": 1647968855,
    "iat": 1647932855,
    "iss": "https://<IDP>",
    "sub": "google-oauth2|110478923717308119755",
    "email": "demo.user@domain.com",
    "nonce": "asTwdJJLSzWP2hgGtKmTXw:https://krahsc.google.com",
    "email_verified": true
  }
}

```

Decrypting Gmail Messages

When a recipient opens an encrypted Gmail message, the message will be decrypted by using the recipient's private key associated with the wrapped private key and certificates uploaded to Google. These were uploaded while completing the [additional prerequisites for Gmail](#).

After the third-party identity provider user authentication is successful, an authentication token (3P JWT) is generated. The user is authenticated using a Google JWT and third-party JWT. When the recipient opens the encrypted message, KACLS unwraps the recipient's wrapped private key, unwraps the encrypted content encryption key (ECEK) with this private key, and returns the message's DEK. Google Workspace CSE decrypts the message using this DEK.

To decrypt an encrypted Gmail message:

1. Open the **Gmail** console, <https://mail.google.com/>.
2. Log on as an end user.
3. Try to open the encrypted message.
4. Sign in with the configured third-party identity provider user credentials.

KACLS' `privatekeydecrypt` API is called internally to decrypt the message for the user. The `privatekeydecrypt` requests are logged at KACLS (under **Records** > **Server Records**), as shown below.

```

{
  "reason": "unpack request",
  "authorization": {
    "aud": "cse-authorization",
    "exp": 1661488112,
    "iat": 1661484512,
    "iss": "gsuitecse-tokenissuer-gmail@system.gserviceaccount.com",
    "role": "decrypter",
    "email": "demo.user@domain.com",
    "kacls_url": "https://demo.domain.com/api/v1/cckm/GoogleWorkspaceCSE/endp
oints/59e7c24f-4573-472d-8e8c-afb622793154",
    "spki_hash": "jFBT+M1baVaQqDRLmQTP1gS3zwrUPXpR/u0tyAnjbkY=",
    "message_id": "<CAGb6PCqtvd6d5n20xazMGkFxnzY1b9yBQPYmVgEdkM60=AmNQ@mai
l.gmail.com>",

```

```

    "perimeter_id": "",
    "resource_name": "//gmail.googleapis.com/gmail/users/demo.user%40domain.com/settings/cse/keypairs/ANe1BmhsTEntl0K99F0xA5Txc1jJgjUSxAms4RzSmRypRSQzh4nYso55Mq9ktKmmPPNLP0Rgx0FLSQXhj6xdg6_0X-7hIg",
    "spki_hash_algorithm": "SHA-256"
  },
  "authentication": {
    "aud": "3kT5KerY3vXrEuzNRrQxVQqaRNvYUAWR",
    "exp": 1661485571,
    "iat": 1661484371,
    "iss": "https://<IDP>",
    "sid": "klqAk-BaWczEf_Unf6RoCM8DPwdXL7hU",
    "sub": "google-oauth2|110478923717308119755",
    "email": "demo.user@domain.com",
    "nonce": "LJOJ79acJoMBIqeTw9vHrg:https://krahsc.google.com",
    "email_verified": true
  }
}

```

Endpoints

This section describes how to create, edit, disable/enable, archive, recover, and delete endpoints. Creation of an endpoint requires identity providers. At least one identity provider must be added to an endpoint. Identity providers can be created in advance or when creating the endpoints. This section assumes that you have already created identity providers, as described in [Creating Identity Providers](#).

Caution

Modify endpoints only through the CipherTrust Cloud Key Management (CCKM) GUI, API, or CLI. Do not modify the KEK using the **Keys** menu, the `ksctl keys` command, or the `/v1/vault/keys2` endpoint. Such modifications might make the KEK unavailable to the Google Workspace CSE service.

Creating KACLS Endpoints

To create an endpoint:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**. The Google Workspace Client Side Encryption page is displayed.
3. Click **Create Endpoint**. The Create Endpoint screen is displayed.
4. Specify a unique **Name** for the endpoint.
5. Specify the **Authentication Audience**.

Authentication Audience is the ID of the third-party identity provider. For example, for Auth0, it is represented by the **Client ID**.

6. (Optional) Switch the **Enable success audit events** toggle on or off to control the audit recording of success logs for the specified endpoint. By default, this toggle is set to active to troubleshoot performance issues.

Note

For improved performance, it is recommended to switch off the **Enable success audit events** toggle.

7. Specify the **Endpoint URL Hostname**. Enter the fully qualified domain name (FQDN) of the CCKM/ CipherTrust Manager appliance. Specify the hostname and port (`<hostname>:<port>`). If the port is not specified, **443** is the default port.
8. (Optional) Enable **Allow Guest Access**.

Note

The web interface port within CipherTrust Manager can be changed from the default port of **443** to another port. If you plan to change the default port for the CipherTrust Manager web interface, ensure to change it **before** configuring the Google Workspace CSE service on CCKM. Also, reflect this port change when creating a KACLS endpoint on CCKM. Changing the default port **after** configuring Google Workspace CSE is not supported. Refer to [Support for Changing the Default Port of Web Interface Setting](#) for instructions on changing the port when configuring a Google Workspace CSE endpoint.

9. Select an **Identity Provider**. The options are:
 - **All**: This is the default option. All the available identity providers will be trusted by the endpoint.
 - **Selected**: Select this option to select/remove the desired identity providers.

Optionally, you can add a new identity provider by clicking **Create Identity Provider**. Refer to [Creating Identity Providers](#) for details.

10. Click **Save**.

The newly created endpoint appears in the endpoints list, with an Active status. Similarly, add as many endpoints as required.

Every endpoint has an associated endpoint URL. This URL is needed to access the Thales key service. Google Workspace administrators use this URL to configure Google Workspace to communicate with the KACLS.

Viewing Endpoints

The Google Workspace Client Side Encryption page shows the available endpoints. The **ENDPOINTS** section shows Name, Endpoint URL (or *KACLS URL*), Status, Key Name, Key ID, and Key Version.

To view details of an endpoint:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**. The **ENDPOINTS** section shows the following details:

Column	Description
Name	Name of the endpoint.
Endpoint URL	(KACLS URL) URL of the endpoint. This URL is needed to access the Thales key service. Google Workspace administrators use this URL to configure Google Workspace to communicate with the KACLS.
Status	Status of the endpoint. The status can be: • Active: The endpoint is enabled. An endpoint is active when it is created, and can be disabled, archived, or deleted. • Disabled: The endpoint is disabled and cannot be used for encryption and decryption of data. A disabled endpoint can be enabled, archived, or deleted. • Archived: The endpoint is archived. An archived endpoint can be recovered later, if required.
Key Name	Name of the linked encryption key.
Key ID	ID of the linked encryption key. The ID changes on endpoint key rotation.
Key Version	Version of the encryption key. The key version changes on endpoint key rotation.

To view the perimeters of an endpoint, click the expand icon  to the left of the desired endpoint.

Viewing or Editing Endpoints

After an endpoint is created, you can view and modify the linked authentication audience, hostname for the endpoint URL, and identity provider.

To view and edit an endpoint details:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**.
3. Under **ENDPOINTS**, click the overflow icon  corresponding to the endpoint you want to edit.

Note

The web interface port within CipherTrust Manager can be changed from the default port of 443 to

another port. If you plan to change the default port for the CipherTrust Manager web interface, ensure to change it **before** configuring the Google Workspace CSE service on CCKM. Also, reflect this port change when creating a KACLS endpoint on CCKM. Changing the default port **after** configuring Google Workspace CSE is not supported. Refer to [Support for Changing the Default Port of Web Interface Setting](#) for instructions on changing the port when configuring a Google Workspace CSE endpoint.

4. Click **View/Edit**. The Edit Endpoint wizard is displayed.

5. (Optional) Make the necessary changes.

- Add or remove **Authentication Audience**.
- Update **Endpoint URL Hostname**.
- Enable **Allow Guest Access**.
- Update the **Identity Provider**.

You can also create a new identity provider by clicking **Create Identity Provider**. Refer to [Creating Identity Providers](#) for details.

6. Click **Save**.

The endpoint details are updated.

Rotating Endpoint Keys

Key rotation is process of changing an endpoint's existing key used to encrypt or decrypt the DEK.

To rotate encryption key for an endpoint:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**.
3. Under **ENDPOINTS**, click the overflow icon  corresponding to the desired endpoint.
4. Click **Rotate Keys**. A message appears prompting to confirm the action.
5. Click **Rotate Key**.

The endpoint key is rotated successfully.

Disabling Endpoints

When an endpoint is not needed for certain period of time, it can be disabled from the CipherTrust Manager.

Note

Before disabling the endpoint, ensure that it is not in use. If an in-use endpoint is disabled, Google Workspace cannot encrypt or decrypt content using the endpoint URL. Also, disabling an endpoint does not delete the associated encryption key.

To disable an endpoint:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**.
3. Under **ENDPOINTS**, click the overflow icon  corresponding to the endpoint you want to disable.
4. Click **Disable**.

The endpoint status becomes Disabled.

Enabling Endpoints

When a disabled endpoint is needed again, enable it from the CipherTrust Manager.

To enable an endpoint:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**.
3. Under **ENDPOINTS**, click the overflow icon  corresponding to the endpoint you want to enable.
4. Click **Enable**.

The endpoint status becomes Active.

Archiving Endpoints

Whenever needed, you can archive an endpoint from the CipherTrust Manager. An archived endpoint can be recovered later, if needed.

To archive an endpoint:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**.
3. Under **ENDPOINTS**, click the overflow icon  corresponding to the endpoint you want to archive.
4. Click **Archive**.

The endpoint status becomes Archived.

Recovering Archived Endpoints

An archived endpoint can be recovered from the CipherTrust Manager. A recovered endpoint can be used again to encrypt and decrypt data.

To recover an archived endpoint:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**.
3. Under **ENDPOINTS**, click the overflow icon  corresponding to the archived endpoint you want to recover.
4. Click **Recover**.

The endpoint status becomes Active.

Deleting Endpoints

When an endpoint is no longer needed, delete it from the CipherTrust Manager.

Note

Before deleting the endpoint, ensure that it is not in use. If an in-use endpoint is deleted, Google Workspace cannot encrypt or decrypt content using the endpoint URL. Also, deleting an endpoint does not delete the associated encryption key.

To delete an endpoint:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**.
3. Under **ENDPOINTS**, click the overflow icon  corresponding to the endpoint you want to delete.
4. Click **Delete**.

A warning message appears stating that the endpoint will be deleted permanently. Do you want to delete this endpoint?

5. Select **I wish to delete this endpoint**.
6. Click **Delete**.

The endpoint is removed from the endpoints list.

Identity Providers

This section describes how to create and delete identity providers. An identity provider authenticates the identities of users logging on to Google Drive to use Google Workspace CSE. It is verified using either an issuer or a JSON Web Key Set (JWKS). Identity providers are required when creating endpoints. Refer to [Creating Endpoints](#) for details.

Creating Identity Providers

An issuer or an openId configuration URL is required to create an identity provider.

To create an identity provider:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**. The Google Workspace Client Side Encryption page is displayed.
3. Under **IDENTITY PROVIDERS**, click **Create Identity Provider**. The Create Identity Provider screen is displayed.
4. Specify a unique **Name** for the provider. This is a mandatory field.
5. Select the **Provider Verifier**. This is a mandatory field.

The verifier can be:

- **Issuer:** Select and specify the valid **Issuer** and **jwksURL**. The **jwksURL** field is optional, and is populated automatically based on the specified issuer.
 - **OpenId Configuration URL:** Select and specify the **OpenId Configuration URL**. The URL must be valid. For example, when using SafeNet Trusted Access (STA) as an identity provider, the URL is represented by **WELL KNOWN CONFIGURATION URL** on the STA Management Console. Other supported identity providers are Auth0, Okta, and Google Identity Provider.
6. Click **Next**. The Confirm and Create screen is displayed.
 7. Verify the provider details. If the details are incorrect or you want to modify them, click **Back** and update the details.
 8. Click **Save**.

The newly created identity provider appears in the providers list. Similarly, add as many identity providers as required.

Viewing Identity Providers

The Google Workspace Client Side Encryption page shows the available identity providers. The **IDENTITY PROVIDERS** section shows the Name, Issuer, OpenID Configuration URL, and jwksURL.

To view identity providers:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**.
3. Navigate to **IDENTITY PROVIDERS**. The list of identity providers is displayed.

The following details are shown:

Column	Description
Name	Name of the identity provider.
Issuer	Issuer string from the identity provider JWT.
OpenID Configuration URL	URL of the OpenID configuration.
jwksURL	URL of JWKS.

Deleting Identity Providers

When an identity provider is no longer needed, delete it from the CipherTrust Manager. Before deleting the provider, ensure that it is not in use by any endpoints.

To delete an identity provider:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **Services > Google Workspace CSE**.
3. Under **IDENTITY PROVIDERS**, click the overflow icon  corresponding to the provider you want to delete.
4. Click **Delete**.

A warning message appears stating that the provider will be deleted permanently. Do you want to delete this identity provider?

5. Select **I wish to delete this provider**.
6. Click **Delete**.

The identity provider is removed from the providers list.

Viewing Google Workspace CSE Records

This section describes the logs of Google Workspace events. Whenever an action is performed on an endpoint, an event is logged. The actions can be rotate keys, create endpoint, delete endpoint, wrap, unwrap, privileged-unwrap, private key sign, private key unwrap, and private key privileged-unwrap.

To view Google Workspace CSE events:

1. Open the **CipherTrust Manager** Application.
2. In the left pane, click **Records > Loki Audit Records**. The default **Server Records** tab shows the following details:

Column	Description
Status	Status of the event.
Event Created	Time when the event is created.
Severity	Severity of the event. The severity can be: • INFO • WARNING • ERROR • CRITICAL • FATAL
Message	Description of the event.
Service	Name of the service that logged the event.
Details	Details of the event.
Source	Hostname of the CipherTrust Manager.

Note

In some cases, you might see the following error:

```
invalid authorization token - cause: untrusted authentication token:
square/go-jose/jwt: validation field, token issued in the future (iat)
```

To resolve this issue, [add the Google NTP server to the CipherTrust Manager \(Admin Settings > NTPs\)](#).

Performance Summary

This section provides the performance summary of different Google Workspace CSE services managed through CCKM on the CipherTrust Manager server. The summary describes:

- What is the configuration (CPU, RAM, and Disk) of each server?
- Where are the clients located in relation to the servers?
- What are the minimum network requirements for reliable connectivity?
- How many requests per second (RPS) are handled in different scenarios?

Test Setup

This section describes the test environment in which the performance of the CCKM Google Workspace CSE service has been tested. The performance numbers provided in this section are based on this setup.

Note

Actual performance numbers in your environment can be different. The results can vary based on factors such as how and where the CipherTrust Manager is deployed, the CipherTrust Manager resources, the location of clients, the network connectivity, and how the traffic is load-balanced.

CipherTrust Manager

- Virtual CipherTrust Manager k170v deployed to
 - Google Cloud
 - Amazon Web Services (AWS)
- Physical Appliance k570

Client

k6 (an open-source load testing tool) is used as a client to simulate traffic between the Google Workspace CSE and the CipherTrust Manager.

The k6 client is deployed on the Google Cloud in the `us-central1-a` region.

Network Requirements

The following ports must be open for CipherTrust Manager/CCKM communication:

Type	Protocol	Port Range
SSH	TCP	22
HTTPS	TCP	443
PostgreSQL (for Cluster)	TCP	5432

Performance Numbers

Refer to the following sections for performance numbers, comparison graphs, and conclusion of performance for the Google services managed through CCKM on the CipherTrust Manager server.

- Google Calendar
- Google Drive
- Google Meet
- Gmail

Google Calendar

This section provides the performance summary of the Google Calendar service managed through CCKM on the CipherTrust Manager server.

Requests Per Second

The following sections list the wrap requests per second, approximate latency, and the number of virtual users for different deployment scenarios.

Note

To provide a good user experience, Google recommends a maximum latency of 200 ms (for 99% of the requests). Therefore, the performance numbers on this page are based on a latency of around 200 ms.

- Google Cloud
- AWS Cloud
- Physical Appliance

Google Cloud

Server Location	Client Location
us-central1-a	us-central1-a

Simulated the **wrap** requests for Google Calendar on the CipherTrust Manager deployed on Google Cloud Platform using the k6 tool. The following table shows the handled number of requests per second (RPS), with approximate latency, and the number of virtual users for different data samples on a standalone CipherTrust Manager and a two-node CipherTrust Manager cluster connected with a load balancer.

Click a tab to view performance numbers based on two different specifications.

Specification 1

System Volume	Memory	CPUs	NICs
50 GB	16 GB	4	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	22.16	16.94002
20	25.87	33.700992

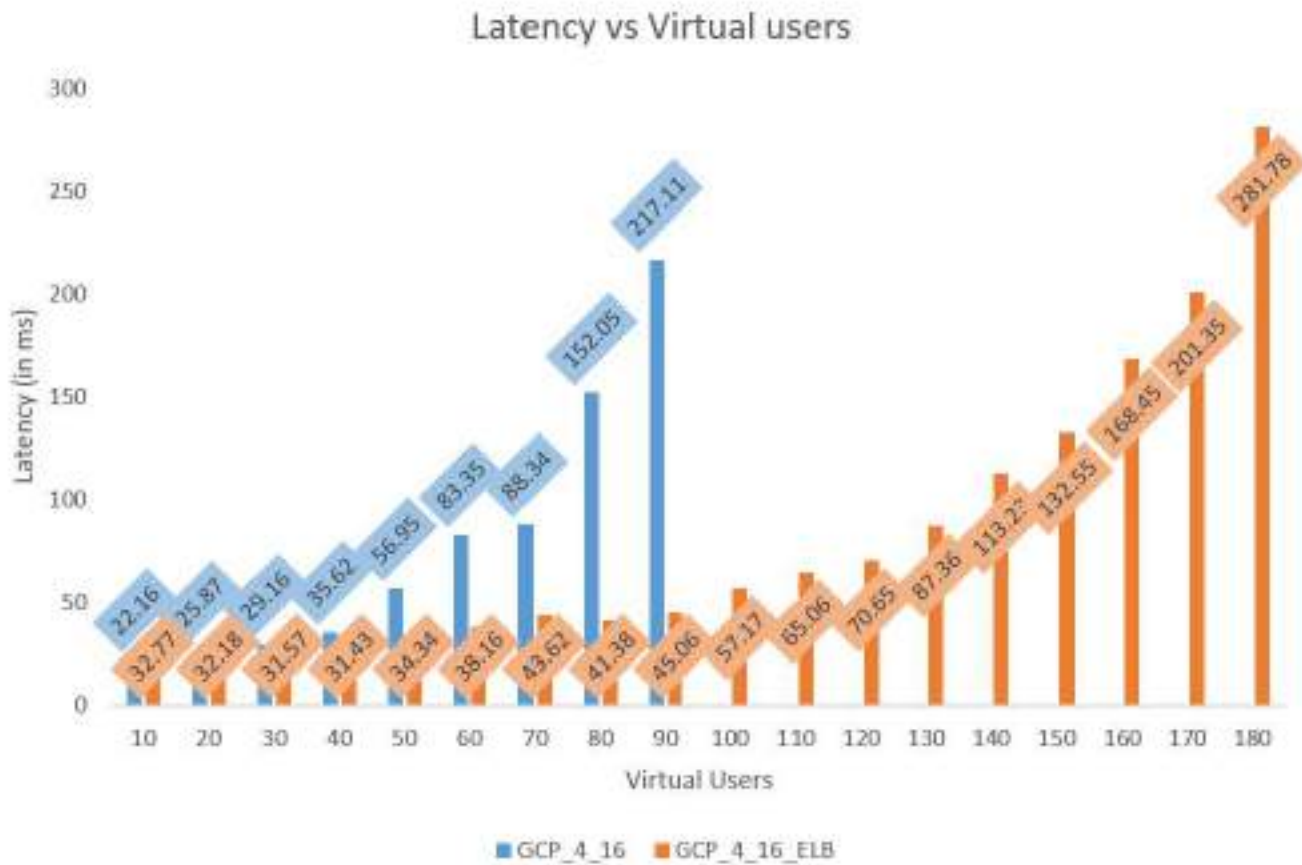
Users	Latency (in ms)	Requests/Second
30	29.16	50.699
40	35.62	66.905176
50	56.95	82.709484
60	83.35	97.367296
70	88.34	111.414325
80	152.05	122.93235
90	217.11	131.68626

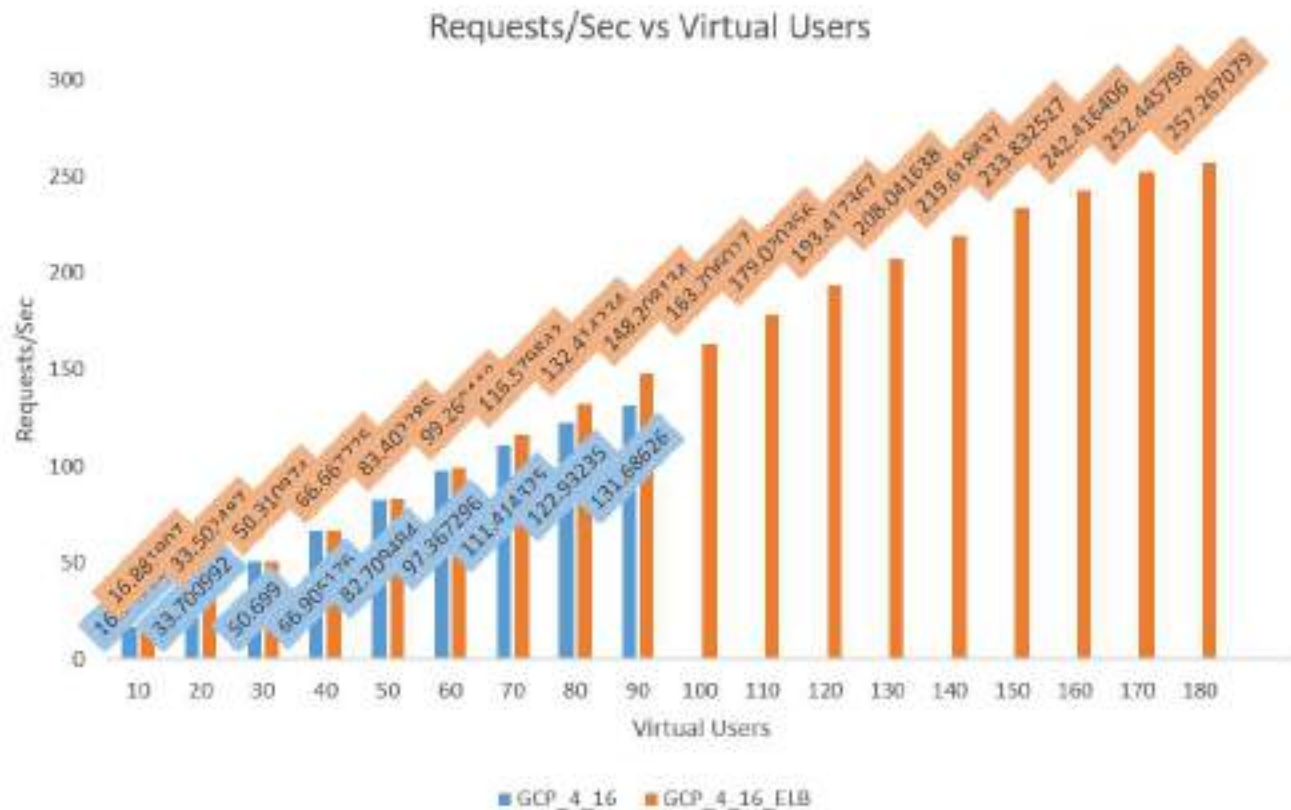
Two-Node Cluster with Load Balancer

Users	Latency (in ms)	Requests/Second
10	32.77	16.881907
20	32.18	33.502487
30	31.57	50.310974
40	31.43	66.667725
50	34.34	83.402285
60	38.16	99.260418
70	43.62	116.579842
80	41.38	132.414234
90	45.06	148.208134
100	57.17	163.706027
110	65.06	179.020356
120	70.65	193.417367
130	87.36	208.041638
140	113.23	219.618837
150	132.55	233.832527
160	168.45	242.416406
170	201.35	252.445798

Comparison Graphs

Specification 1: Standalone vs Two-Node Cluster with Load Balancer





Specification 2

System Volume	Memory	CPUs	NICs
50 GB	64 GB	8	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	15.2	17.169325
30	18.71	51.008778
50	17.95	85.034018
70	19.99	118.406895
90	22.8	151.867389
110	29.53	185.042165
130	26.8	218.012588
150	50.45	247.726422
170	60.21	277.771915
190	82.69	308.844717

Users	Latency (in ms)	Requests/Second
210	89.9	336.886335
230	110.77	360.546328
250	129.13	386.616301
270	186.98	400.53527
290	231.83	415.022139

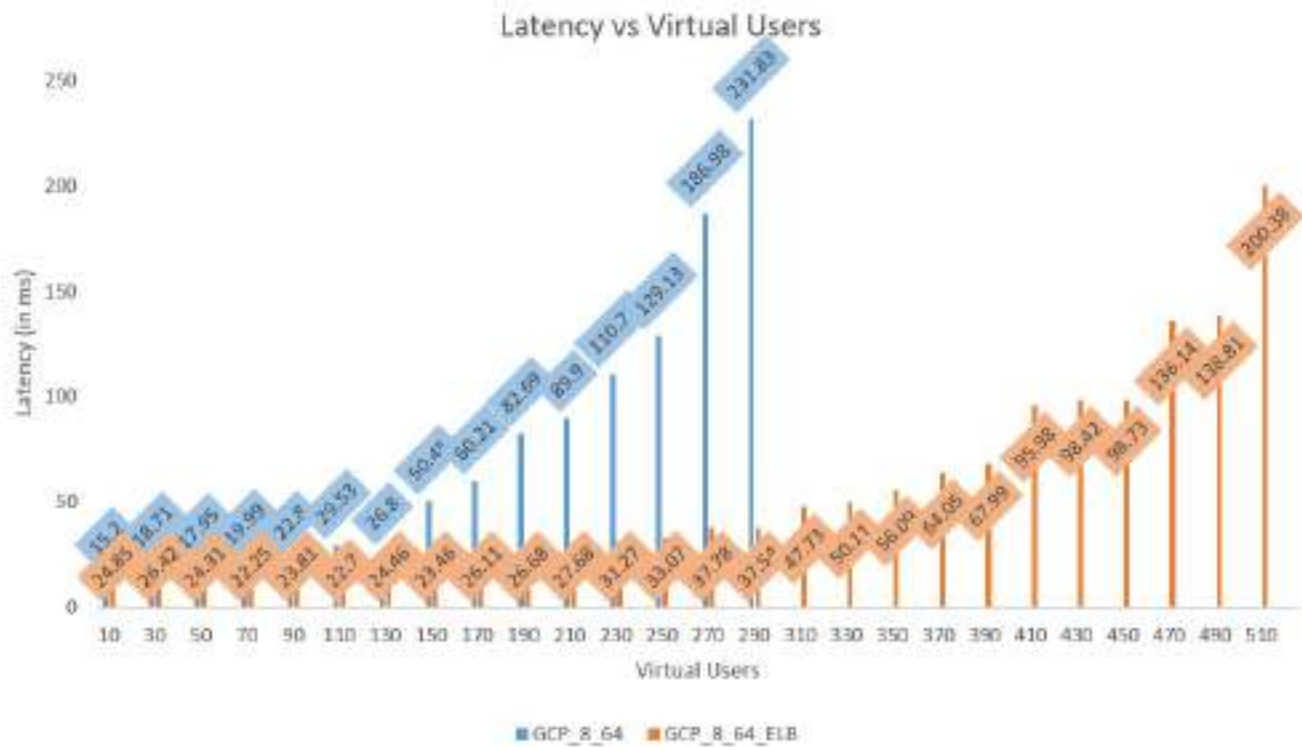
Two-Node Cluster with Load Balancer

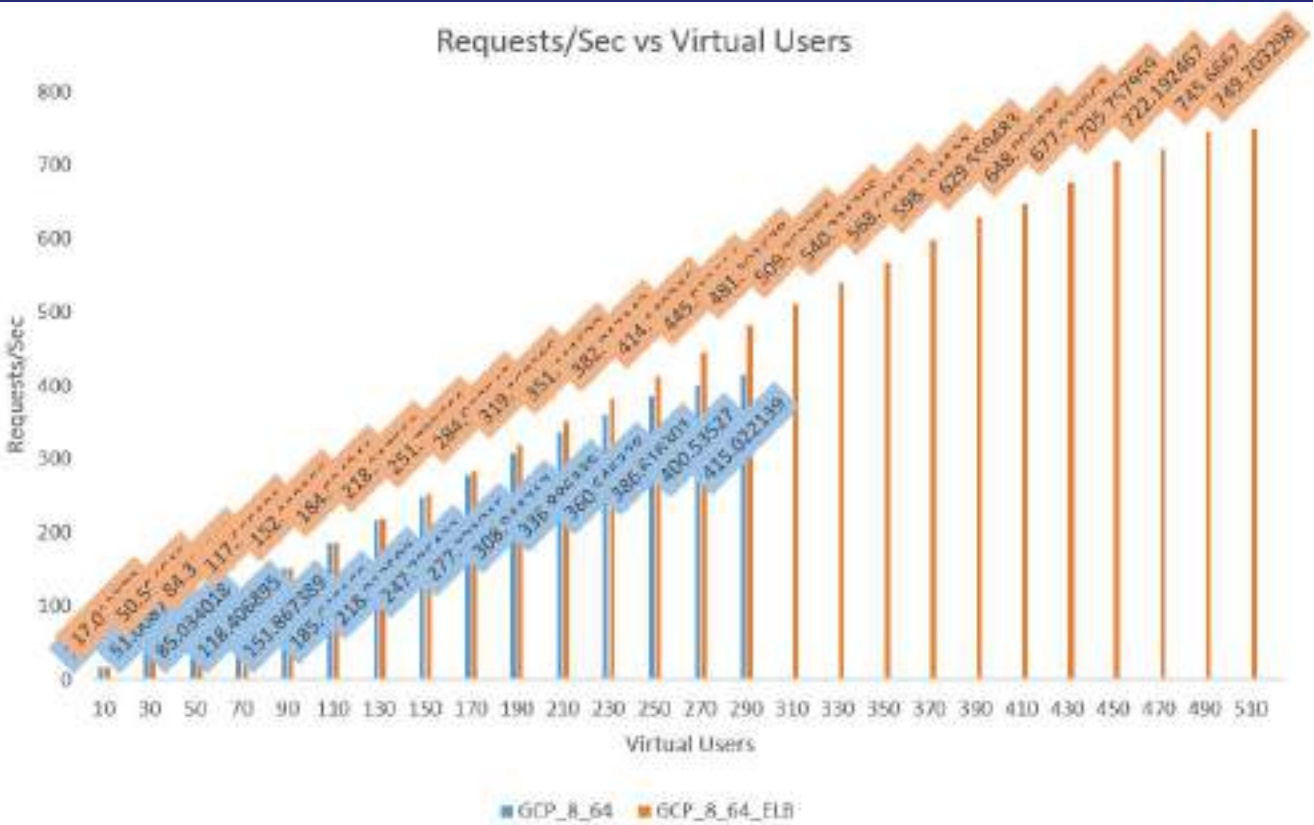
Users	Latency (in ms)	Requests/Second
10	24.85	17.013772
30	26.42	50.594645
50	24.31	84.351819
70	22.25	117.585101
90	23.81	152.119166
110	22.7	184.904241
130	24.46	218.317553
150	23.46	251.799291
170	26.11	284.927543
190	26.68	319.168286
210	27.68	351.441629
230	31.27	382.913143
250	33.07	414.149236
270	37.78	445.583614
290	37.54	481.201628
310	47.73	509.962281
330	50.11	540.231395
350	56.09	568.691533
370	64.05	598.184122
390	67.99	629.559483
410	95.98	648.096876
430	98.42	677.029967
450	98.73	705.757959
470	136.14	722.192467
490	138.81	745.6667

Users	Latency (in ms)	Requests/Second
510	200.38	749.703298

Comparison Graphs

Specification 2: Standalone vs Two-Node Cluster with Load Balancer





Users	Latency (in ms)	Requests/Second
20	45.38	32.437778
30	44.43	48.449399
40	45.52	64.597006
50	45.76	80.851734
60	50.06	96.457289
70	49.02	112.148661
80	71.87	127.43983
90	62.98	142.926131
100	76.5	157.384469
110	85.53	172.640989
120	88.88	187.20207
130	122.48	198.203387
140	125.95	212.196902
150	177.16	220.655164
160	196.31	232.058391
170	205.23	242.47253

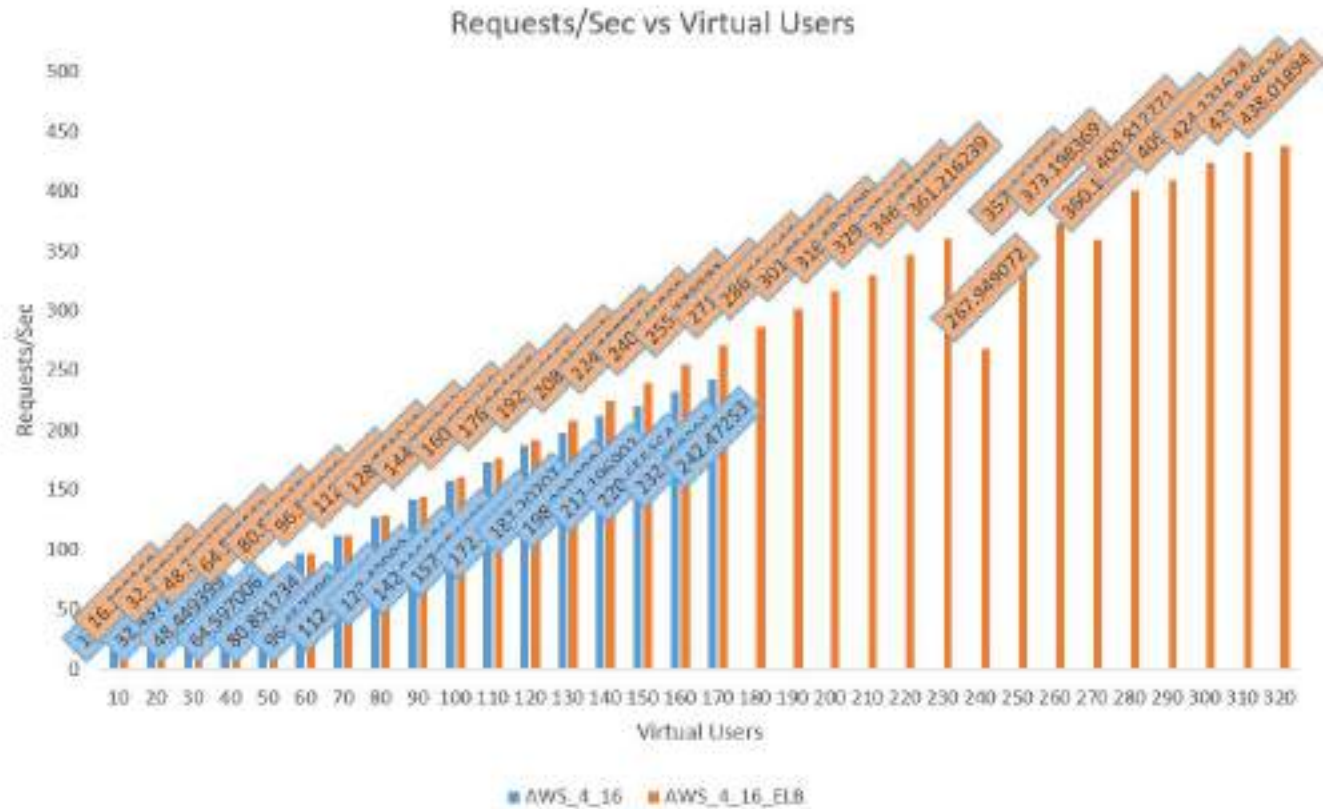
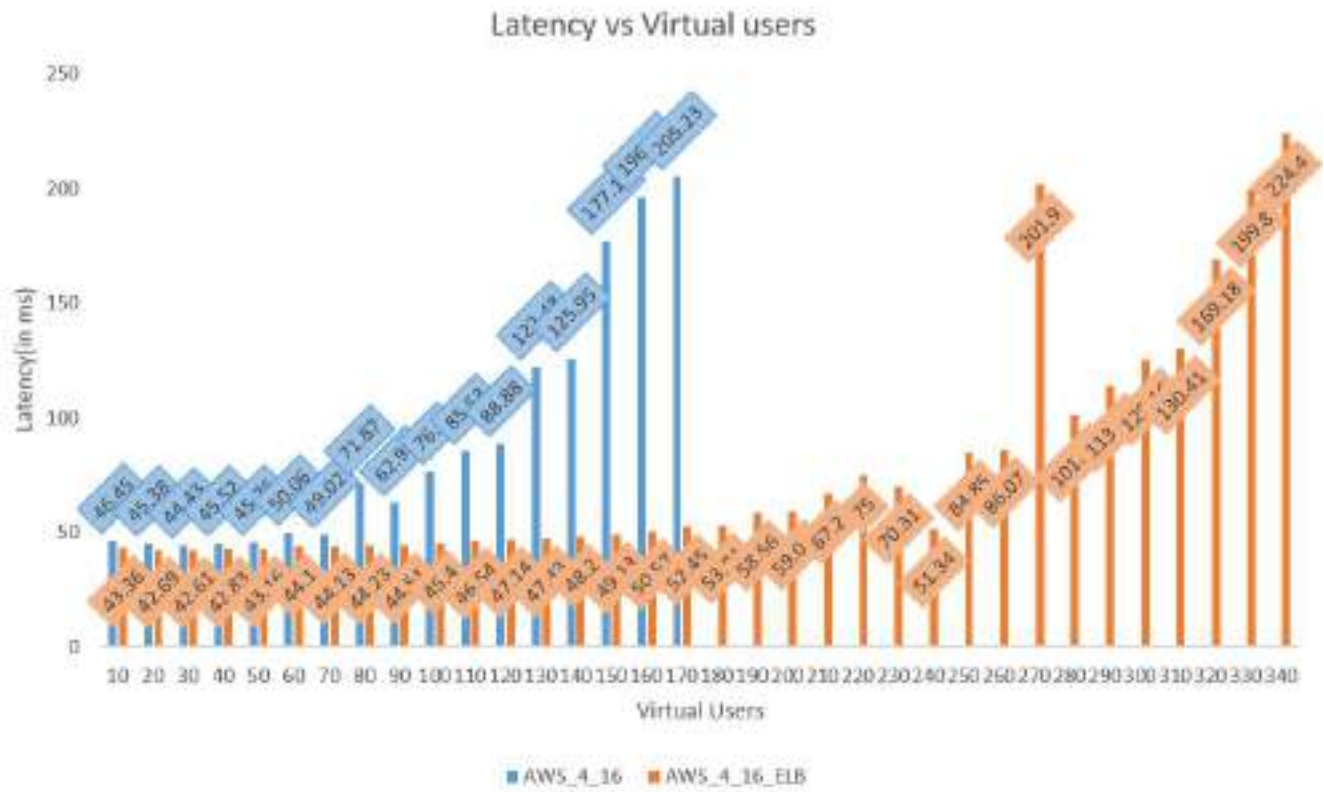
Two-Node Cluster with Load Balancer

Users	Latency (in ms)	Requests/Second
10	43.36	16.230968
20	42.69	32.307195
30	42.61	48.362801
40	42.83	64.588292
50	43.16	80.524303
60	44.1	96.912002
70	44.13	112.44579
80	44.23	128.580249
90	44.51	144.772264
100	45.4	160.873428
110	46.54	176.926386
120	47.14	192.216205
130	47.43	208.636698
140	48.2	224.307708

Users	Latency (in ms)	Requests/Second
150	49.13	240.141065
160	50.57	255.278072
170	52.45	271.01118
180	53.04	286.524106
190	58.56	301.984366
200	59.01	316.592678
210	67.2	329.969441
220	75	346.092386
230	70.31	361.216239
240	51.34	267.949072
250	84.85	357.444582
260	86.07	373.198369
270	201.9	360.15469
280	101.04	400.812771
290	113.94	409.21364
300	125.16	424.121674
310	130.41	432.858526
320	169.18	438.01894
330	199.8	449.49049
340	224.4	455.050062

Comparison Graphs

Specification 1: Standalone vs Two-Node Cluster with Load Balancer



Specification 2

System Volume	Memory	CPUs	NICs
50 GB	64 GB	8	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	39.73	16.465342
30	39.32	48.614473
50	38.94	81.009533
70	39.96	113.01632
90	38.93	145.338532
110	39.68	177.809499
130	41.22	210.307813
150	43.79	242.919137
170	45.37	273.825955
190	52.8	304.935724
210	57.28	335.559065
230	68.32	365.948438
250	81.12	394.664553
270	83.21	423.025098
290	95.12	449.812886
310	123.71	471.120772
330	149.41	491.916512
350	170.34	514.60998
370	214.64	524.963419

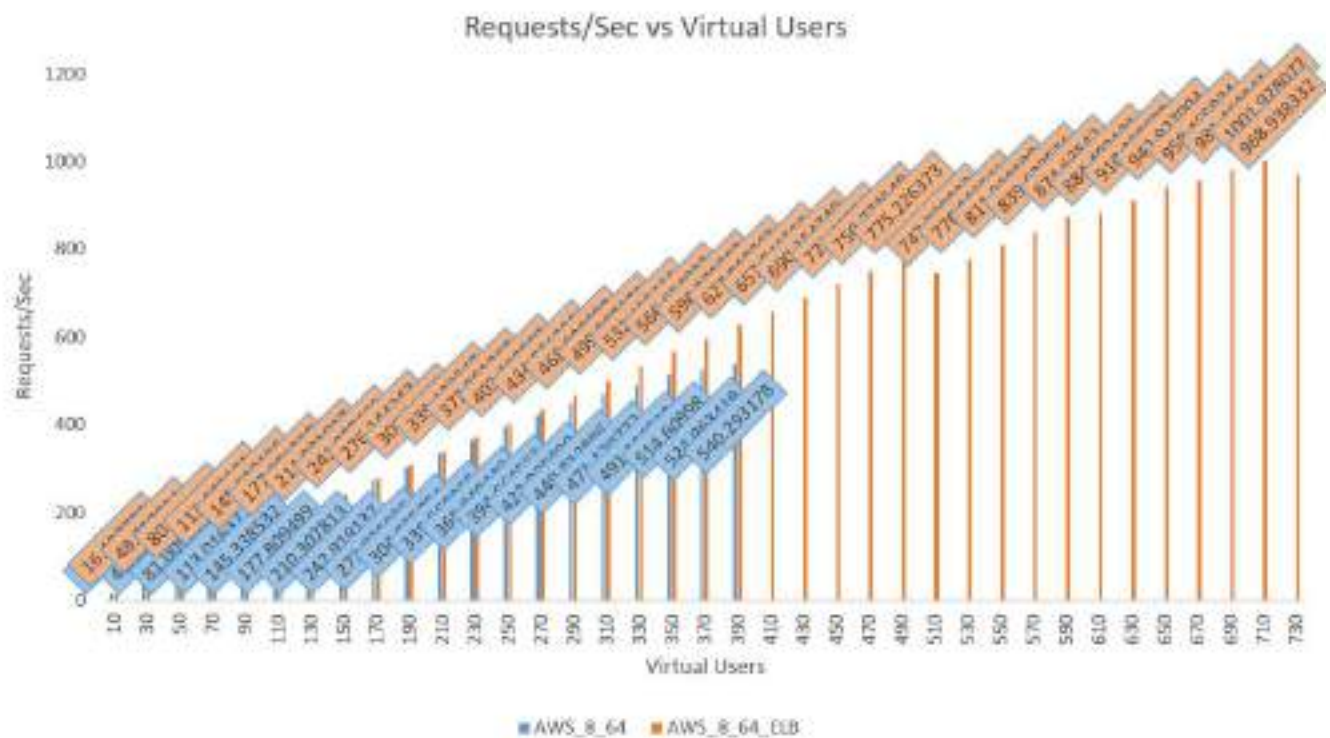
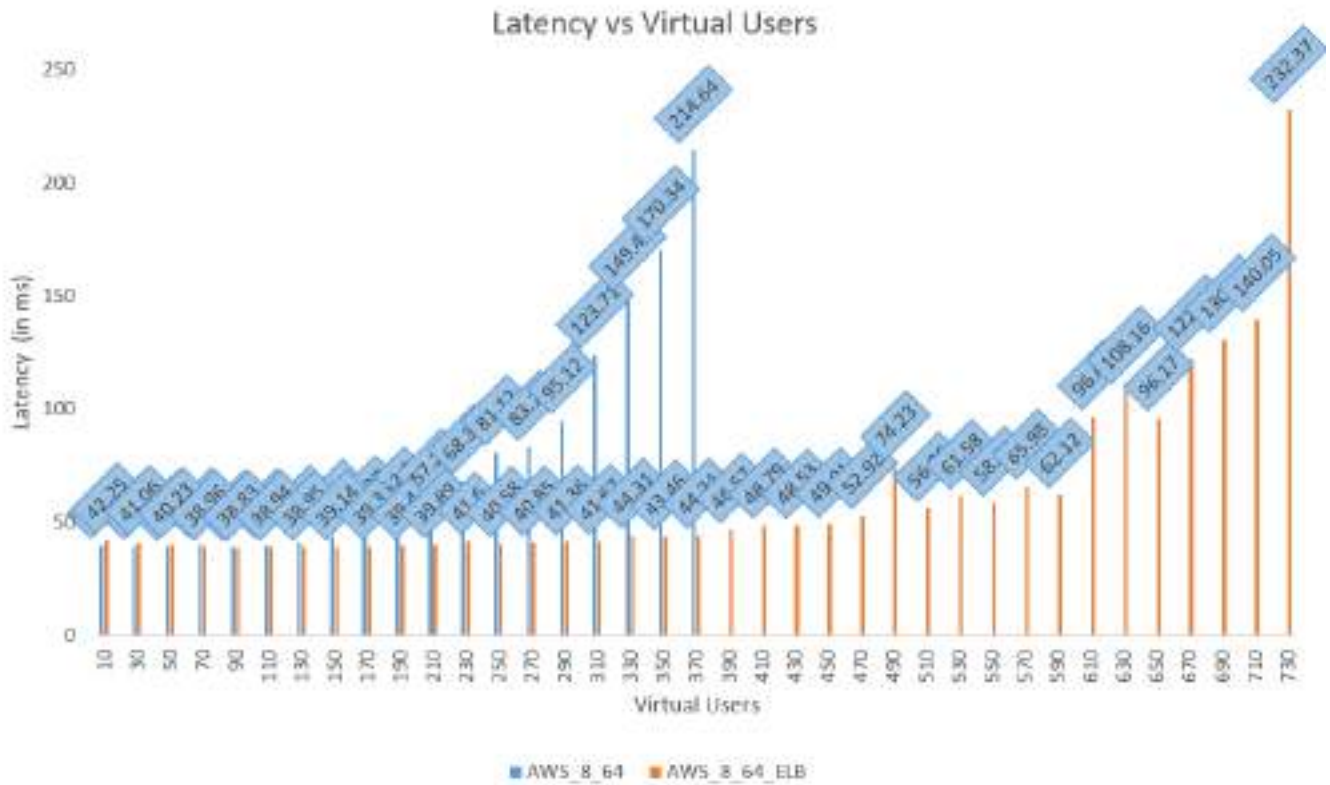
Two-Node Cluster with Load Balancer

Users	Latency (in ms)	Requests/Second
10	42.25	16.137739
30	41.06	48.750214
50	40.23	80.995676
70	38.96	113.089121
90	38.83	145.603135

Users	Latency (in ms)	Requests/Second
110	38.94	177.971337
130	38.95	211.891827
150	39.14	241.927469
170	39.3	276.144247
190	39.4	307.94167
210	39.89	339.423347
230	41.6	372.051267
250	40.58	403.209987
270	40.85	434.881269
290	41.36	468.022003
310	41.67	499.597171
330	44.31	532.419496
350	43.46	566.654822
370	44.24	596.071302
390	46.57	627.917302
410	48.79	657.741502
430	48.53	690.154749
450	49.01	721.60321
470	52.92	750.774548
490	74.23	775.226373
510	56.06	747.230233
530	61.58	776.806541
550	58.52	811.338109
570	65.98	839.637571
590	62.12	874.63542
610	96.62	884.402171
630	108.16	910.027679
650	96.17	942.927004
670	122.17	958.465024
690	130.92	981.462841
710	140.05	1001.978077
730	232.37	968.938332

Comparison Graphs

Specification 2: Standalone vs Two-Node Cluster with Load Balancer



Physical Appliance

Server Location	Client Location
San Jose	us-central1-a

CipherTrust Manager Configuration

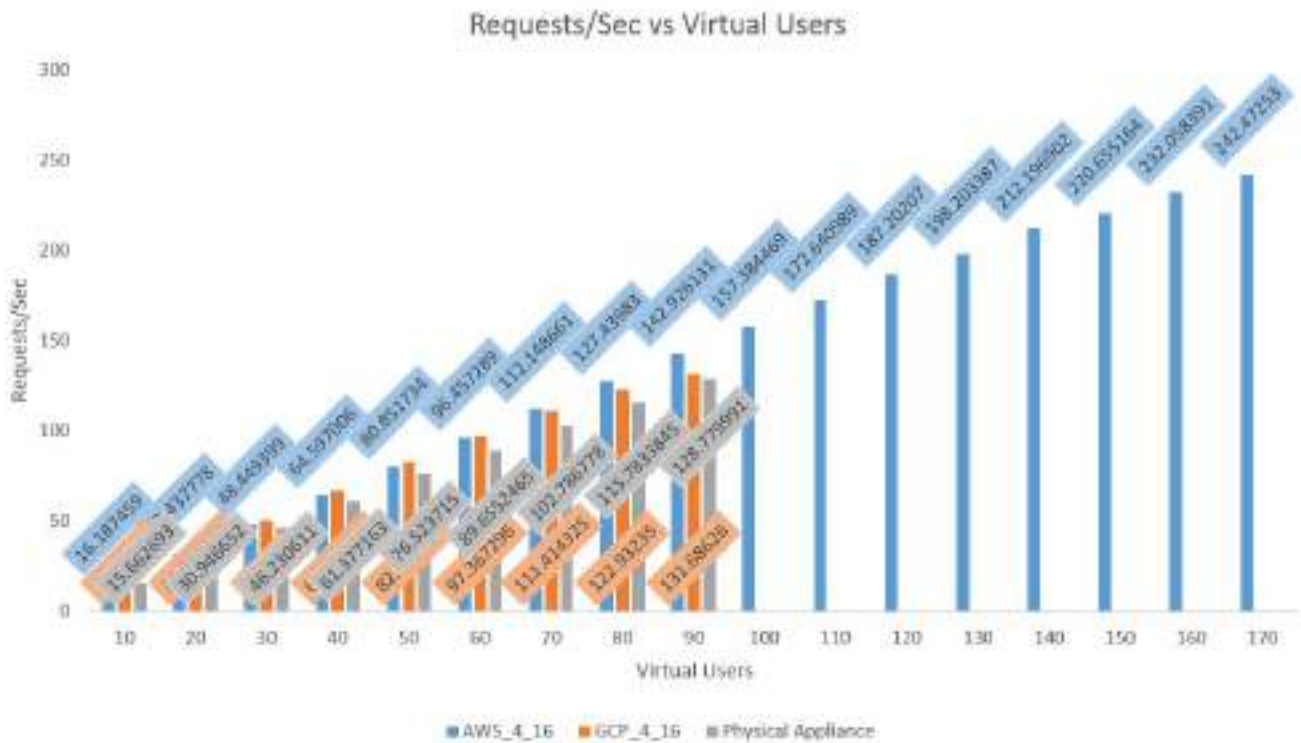
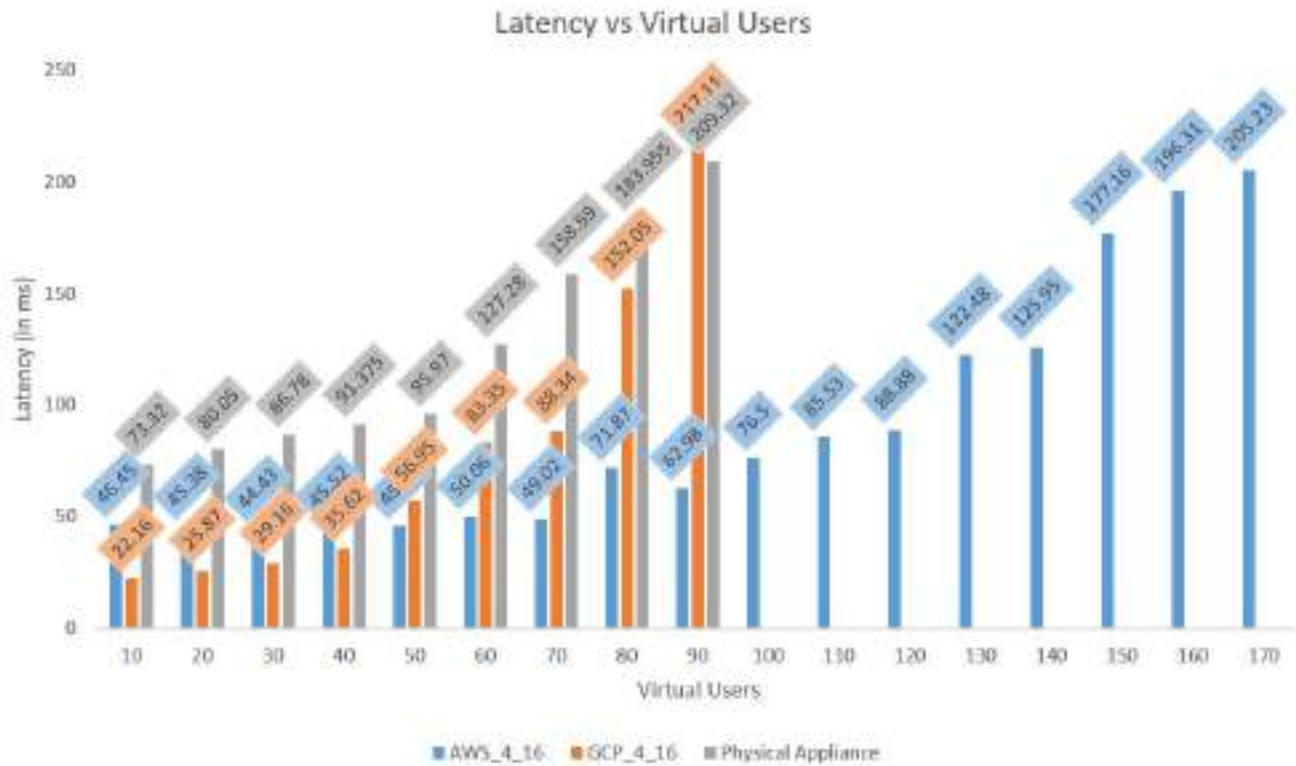
System Volume	Memory	CPUs	NICs
2 TB	16 GB	1 with 4 Cores	1

Simulated the **wrap** requests for Google Calendar on the CipherTrust Manager deployed on a physical appliance using the k6 tool. The following table shows the handled number of requests per second (RPS), with approximate latency, and the number of virtual users for different data sample for three runs.

Users	Latency (in ms)	Requests/Second
10	73.32	15.662693
20	80.05	30.946652
30	86.78	46.230611
40	91.375	61.377163
50	95.97	76.523715
60	127.28	89.6552465
70	158.59	102.786778
80	183.955	115.7833845
90	209.32	128.779991

Comparison Graphs

Physical Appliance vs AWS Cloud vs Google Cloud



Recommendations

Note

Editing a Google Calendar event results in one **wrap** request on every save.

Google Cloud

1. Response time compliance of around **200** ms was met for a maximum throughput of **131.69** operations per second with a standalone CipherTrust Manager k170v instance with **4** CPUs and **16** GB RAM.
2. Response time compliance of around **200** ms was met for a maximum throughput of **252.45** operations per second with a two-node CipherTrust Manager k170v cluster (each node with **4** CPUs and **16** GB RAM) connected with a Google Cloud load balancer.
3. Response time compliance of around **200** ms was met for a maximum throughput of **415.02** operations per second with a standalone CipherTrust Manager k470v instance with **8** CPUs and **64** GB RAM.
4. Response time compliance of around **200** ms was met for a maximum throughput of **749.70** operations per second with a two-node CipherTrust Manager k470v cluster (each node with **8** CPUs and **64** GB RAM) connected with a Google Cloud load balancer.

AWS Cloud

1. Response time compliance of around **200** ms was met for a maximum throughput of **242.47** operations per second with a standalone CipherTrust Manager k170v instance with **4** CPUs and **16** GB RAM.
2. Response time compliance of around **200** ms was met for a maximum throughput of **455.05** operations per second with a two-node CipherTrust Manager k170v cluster (each node with **4** CPUs and **16** GB RAM) connected with an AWS load balancer.
3. Response time compliance of around **200** ms was met for a maximum throughput of **524.96** operations per second with a standalone CipherTrust Manager k470v instance with **8** CPUs and **64** GB RAM.
4. Response time compliance of around **200** ms was met for a maximum throughput of **968.94** operations per second with a two-node CipherTrust Manager k470v cluster (each node with **8** CPUs and **64** GB RAM) connected with an AWS load balancer.

Conclusion

The number of users and throughput almost doubles up on moving from CipherTrust Manager k170v to k470v. Moreover, adding an additional node to the cluster also doubles up the throughput. Overall, a performance gain of 400 percent is achieved by moving from a standalone CipherTrust Manager k170v to a two-node CipherTrust Manager k470v cluster.

Google Drive

This section provides the performance summary of the Google Drive service managed through CCKM on the CipherTrust Manager server.

Requests Per Second

The following sections list the wrap requests per second, approximate latency, and the number of virtual users for different deployment scenarios.

Note

To provide a good user experience, Google recommends a maximum latency of 200 ms (for 99% of the requests). Therefore, the performance numbers on this page are based on a latency of around 200 ms.

- Google Cloud
- AWS Cloud
- Physical Appliance

Google Cloud

Server Location	Client Location
us-central1-a	us-central1-a

Simulated the **wrap** requests for Google Drive on the CipherTrust Manager deployed on Google Cloud Platform using the k6 tool. The following table shows the handled number of requests per second (RPS), with approximate latency, and the number of virtual users for different data samples on a standalone CipherTrust Manager and a two-node CipherTrust Manager cluster connected with a load balancer.

Click a tab to view performance numbers based on two different specifications.

Specification 1

System Volume	Memory	CPUs	NICs
50 GB	16 GB	4	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	65.34	15.71
20	66.57	31.05

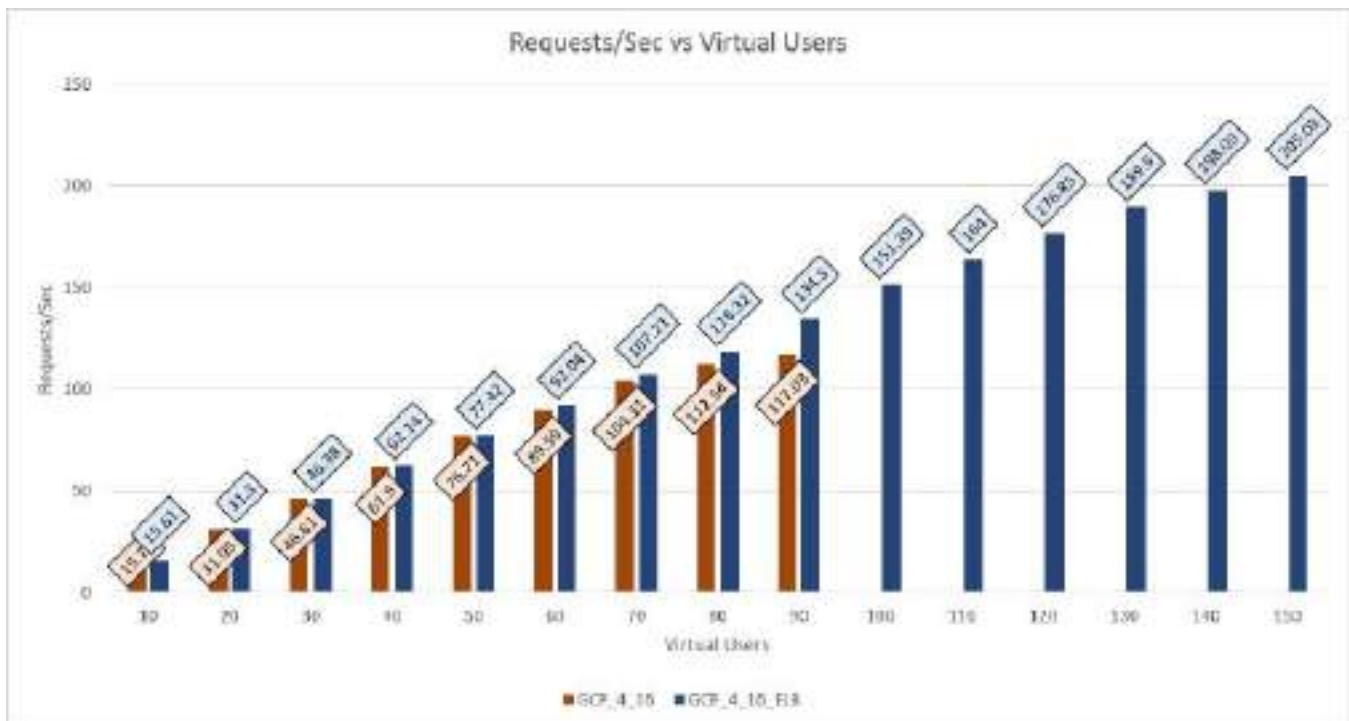
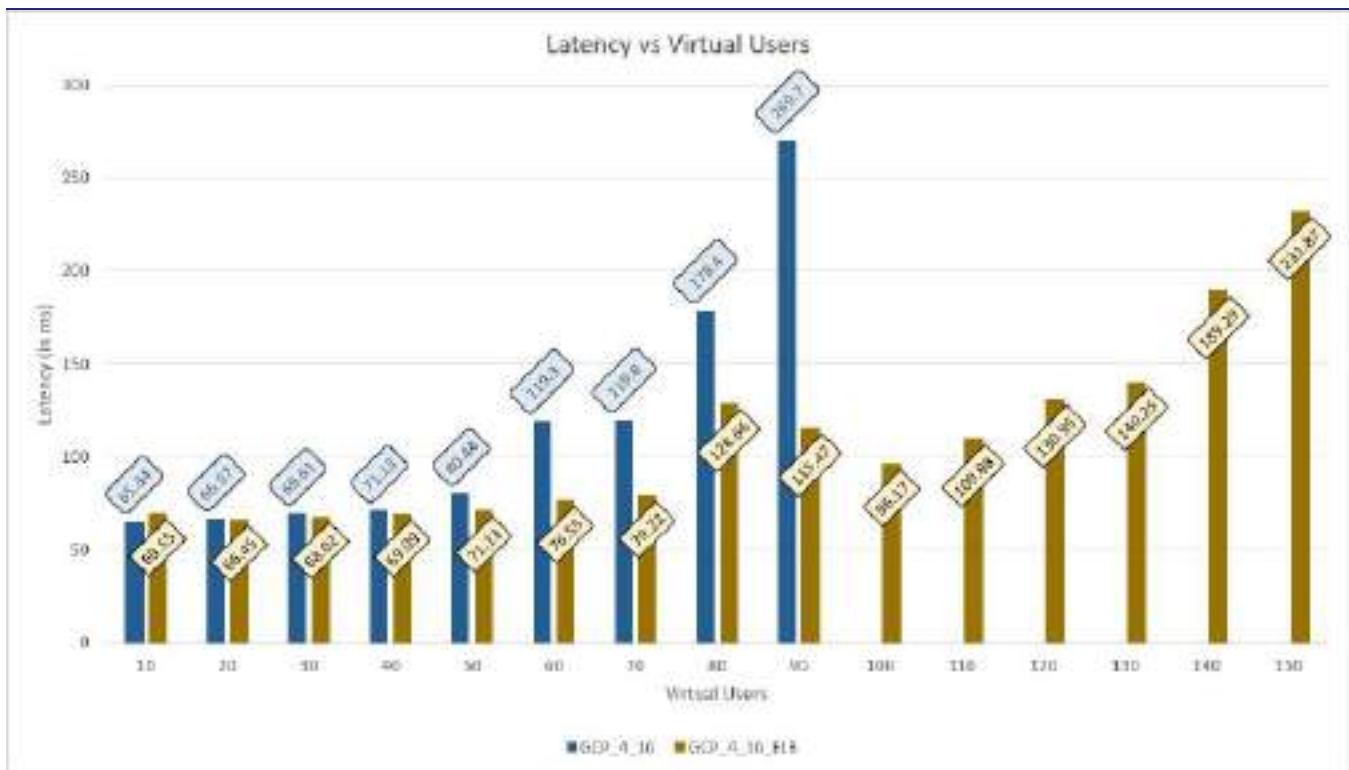
Users	Latency (in ms)	Requests/Second
30	69.61	46.61
40	71.13	61.9
50	80.44	76.71
60	119.31	89.59
70	119.84	104.31
80	178.42	112.54
90	269.76	117.03

Two-Node Cluster with Load Balancer

Users	Latency (in ms)	Requests/Second
10	69.55	15.61
20	66.45	31.3
30	68.02	46.38
40	69.09	62.14
50	71.13	77.42
60	76.55	92.04
70	79.22	107.21
80	128.66	118.32
90	115.47	134.5
100	96.17	151.39
110	109.88	164
120	130.95	176.85
130	140.25	189.6
140	189.29	198.03
150	231.87	205.03

Comparison Graphs

Specification 1: Standalone vs Two-Node Cluster with Load Balancer



Specification 2

System Volume	Memory	CPUs	NICs
50 GB	64 GB	8	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load

balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	78.16	14.89
20	76.14	30.58
30	78.09	45.61
40	77.73	60.73
50	78.17	76.35
60	79.50	90.83
70	81.83	106.42
80	88.93	120.26
90	86.51	134.79
100	89.79	150.23
110	94.87	164.41
120	110.48	177.5
130	108.17	191.41
140	130.7	202.44
150	132.99	216.24
160	147.72	228.45
170	199.9	231.91
180	222.12	241.67

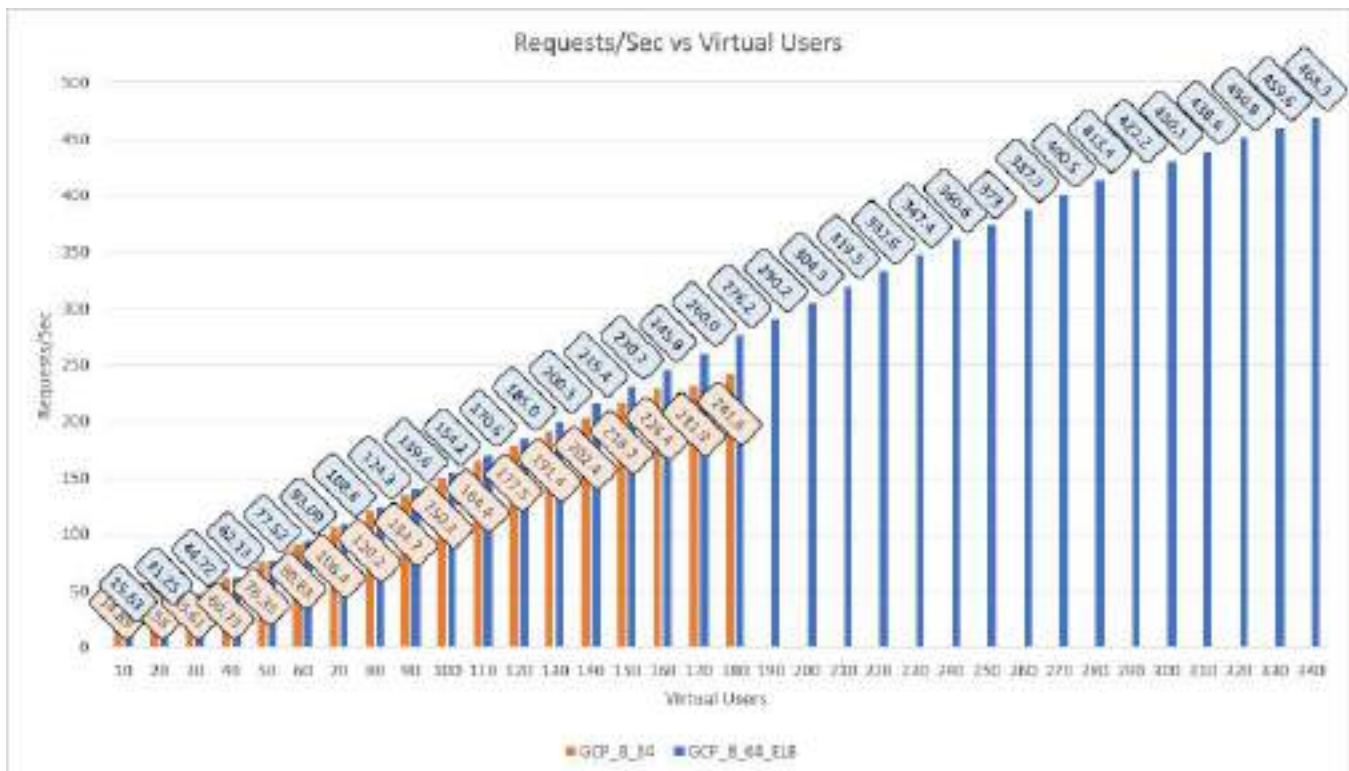
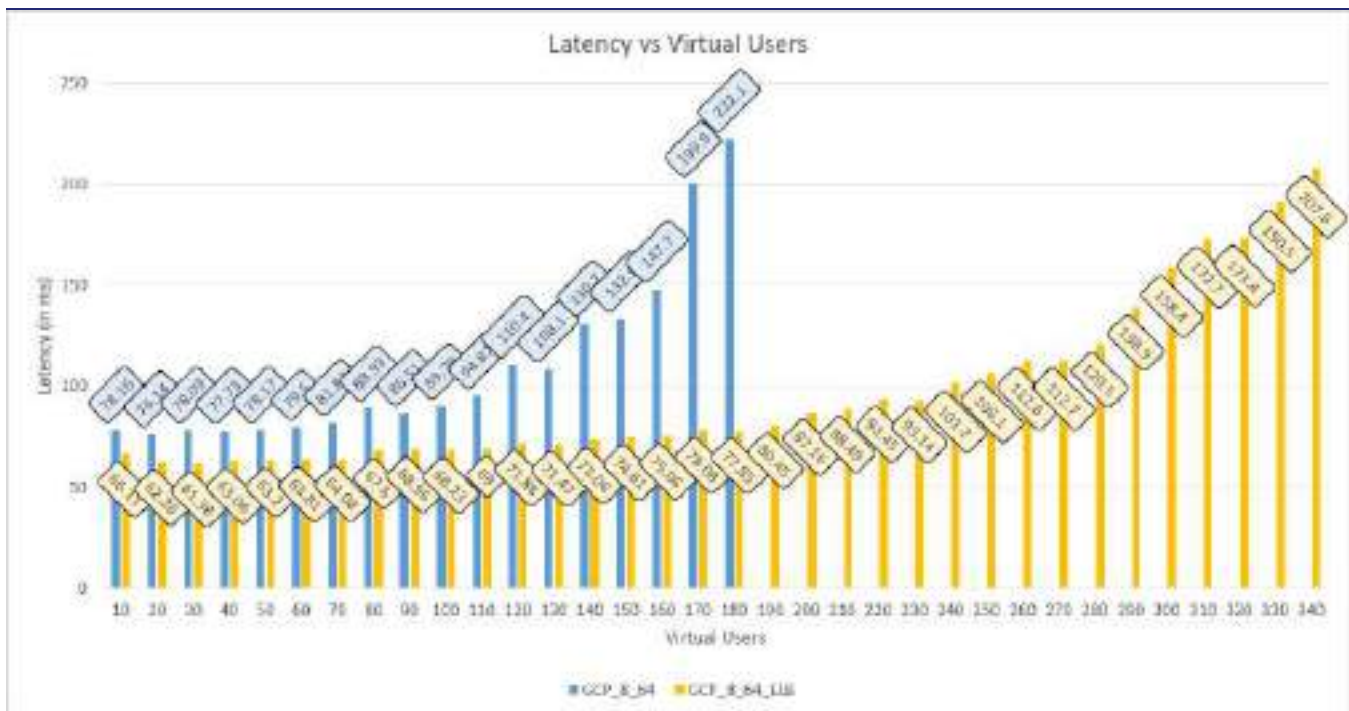
Two-Node Cluster with Load Balancer

Users	Latency (in ms)	Requests/Second
10	66.11	15.63
20	62.26	31.25
30	61.38	46.72
40	63.06	62.13
50	63.2	77.52
60	63.81	93.09
70	64.08	108.82
80	67.6	124.32
90	68.56	139.64
100	68.23	154.2

Users	Latency (in ms)	Requests/Second
110	69.00	170.6
120	71.88	185.07
130	71.47	200.31
140	73.06	215.42
150	74.61	230.29
160	75.06	245.8
170	78.08	260.04
180	77.85	276.29
190	80.45	290.22
200	87.16	304.36
210	88.49	319.56
220	93.45	332.61
230	93.14	347.45
240	101.26	360.64
250	106.16	373
260	112.68	387.33
270	112.72	400.57
280	120.5	413.48
290	138.9	422.29
300	158.41	430.16
310	172.73	438.65
320	173.43	450.89
330	190.55	459.62
340	207.67	468.35

Comparison Graphs

Specification 2: Standalone vs Two-Node Cluster with Load Balancer



AWS Cloud

Server Location	Client Location
us-east-1b	us-central1-a

Simulated the **wrap** requests for Google Drive on the CipherTrust Manager deployed on AWS cloud using the k6 tool. The following table shows the handled number of requests per second (RPS), with approximate latency, and the number of virtual users for different data samples on a standalone CipherTrust Manager and a two-node CipherTrust Manager cluster connected with a load balancer.

Click a tab to view performance numbers based on two different specifications.

Specification 1

System Volume	Memory	CPUs	NICs
50 GB	16 GB	4	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	49.58	16.03
20	47.56	31.97
30	47.48	48.08
40	50.11	63.68
50	51.43	79.52
60	54.08	95.39
70	62.57	110.33
80	67.65	126.25
90	90.55	138.89
100	99.06	152.43
110	157.68	159.96
120	246.52	165.1

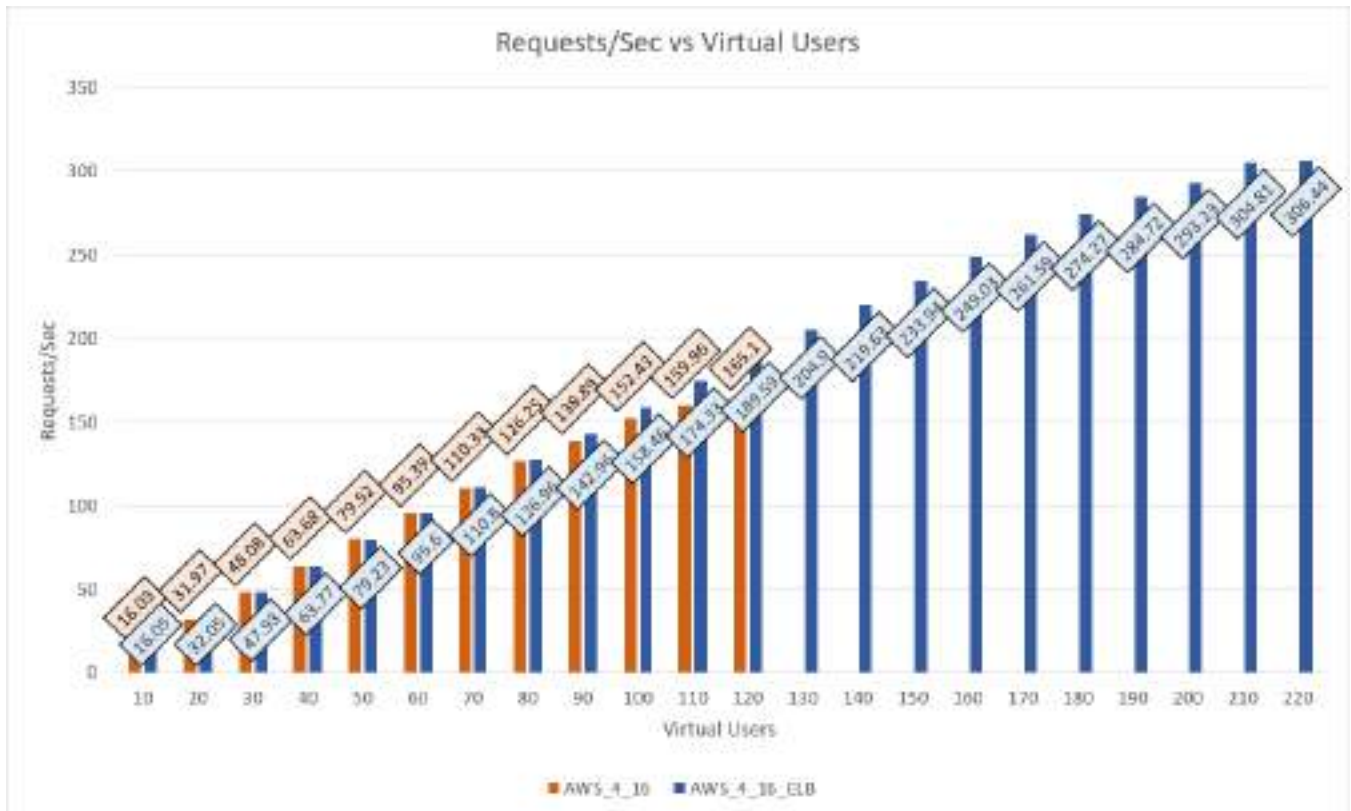
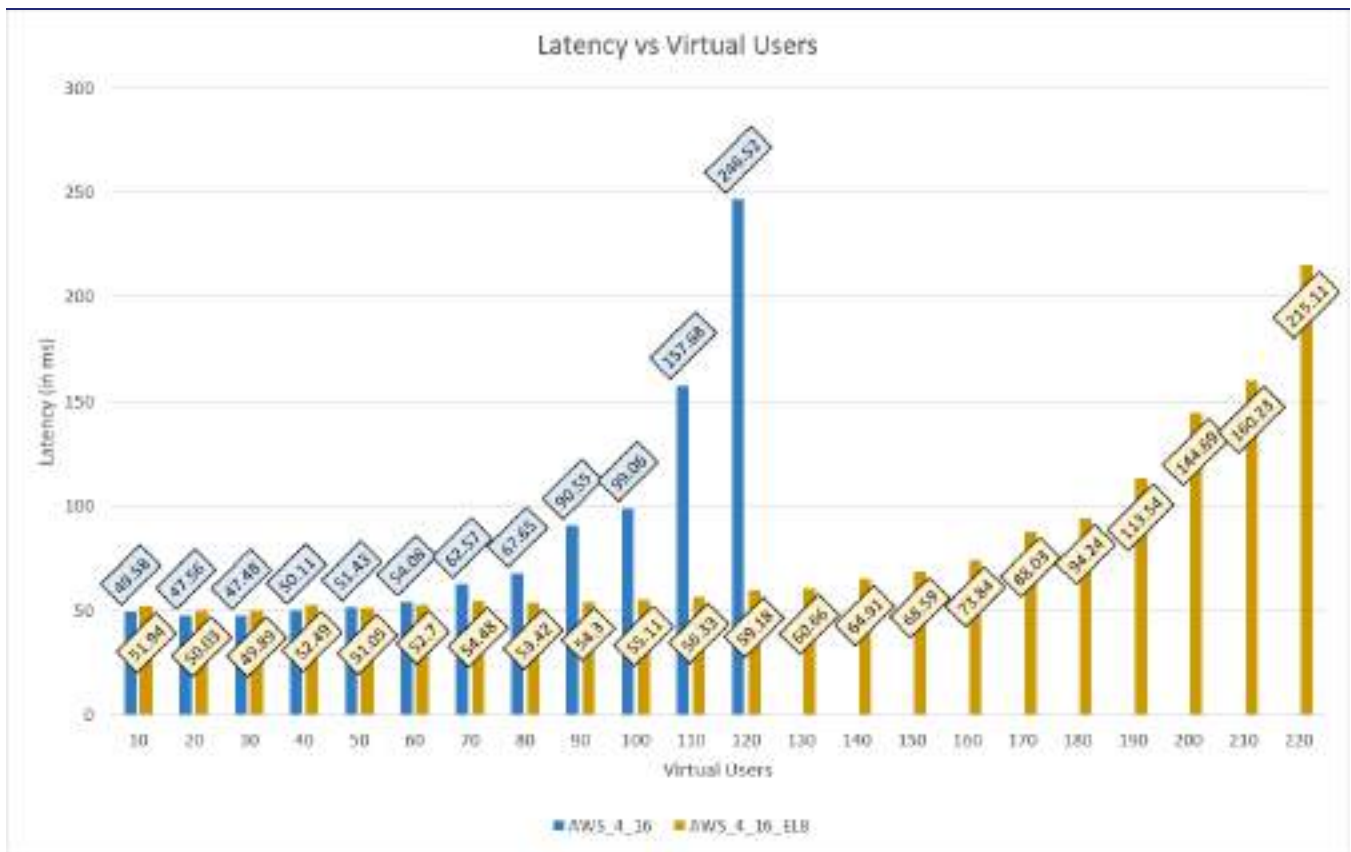
Two-Node Cluster with Load Balancer

Users	Latency (in ms)	Requests/Second
10	51.94	16.05
20	50.03	32.05
30	49.89	47.93
40	52.49	63.77
50	51.05	79.23
60	52.7	95.6
70	54.48	110.8
80	53.42	126.96

Users	Latency (in ms)	Requests/Second
90	54.3	142.96
100	55.11	158.46
110	56.33	174.33
120	59.18	189.59
130	60.66	204.9
140	64.91	219.63
150	68.59	233.94
160	73.84	249.03
170	88.03	261.59
180	94.24	274.27
190	113.54	284.72
200	144.69	293.23
210	160.23	304.81
220	215.11	306.44

Comparison Graphs

Specification 1: Standalone vs Two-Node Cluster with Load Balancer



Specification 2

System Volume	Memory	CPUs	NICs
50 GB	64 GB	8	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	43.97	16.21
20	43.01	32.51
30	42.93	48.42
40	56.53	63.45
50	42.66	80.83
60	50.9	95.36
70	45.54	112.43
80	47.02	128.36
90	65.12	142.32
100	68.99	156.6
110	68.81	170.76
120	79.2	185.46
130	70.51	204.54
140	100.1	214.89
150	131.71	225.22
160	125.13	241.78
170	84.51	262.35
180	104.14	276.44
190	99.66	290.61
200	127.36	301.65
210	136.48	313.56
220	143.68	324.27
230	245.09	318.67

Two-Node Cluster with Load Balancer

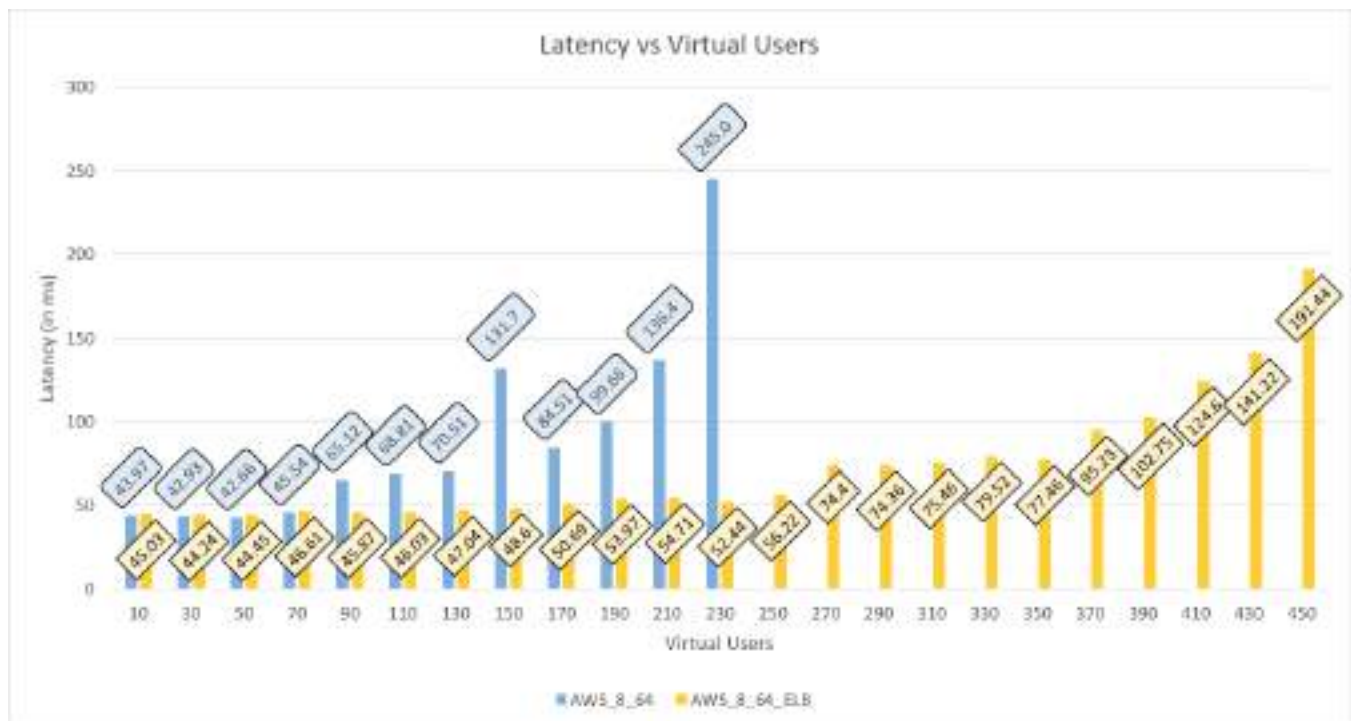
Users	Latency (in ms)	Requests/Second
10	45.03	16.08

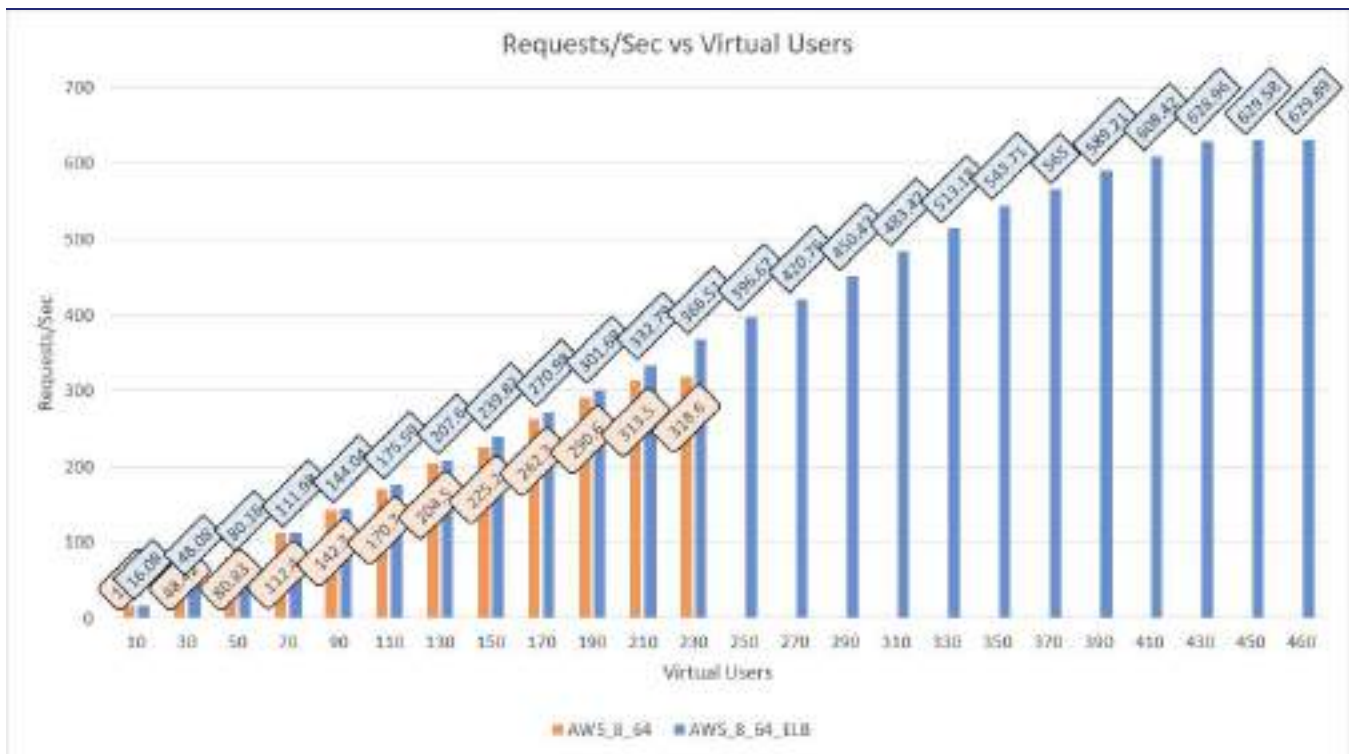
Users	Latency (in ms)	Requests/Second
20	44.06	32.07
30	44.24	48.08
40	44.07	64.46
50	44.45	80.16
60	44.94	96.24
70	46.61	111.98
80	45.69	127.84
90	45.87	144.04
100	45.62	160.35
110	46.03	175.59
120	46.11	191.64
130	47.04	207.6
140	48.32	222.59
150	48.6	239.82
160	46.87	256.42
170	50.69	270.98
180	52.53	285.45
190	53.97	301.68
200	51.31	317.32
210	54.71	332.79
220	55.66	349.72
230	52.44	366.51
240	54.63	381.84
250	56.22	396.67
260	67.06	407.14
270	74.4	420.76
280	78.71	433.79
290	74.36	450.47
300	73.85	468.84
310	75.46	483.42
320	75.96	499.24
330	79.52	513.18
340	88.56	524.6

Users	Latency (in ms)	Requests/Second
350	77.46	543.71
360	84	554.31
370	95.23	565
380	100.92	576.25
390	102.75	589.21
400	103.8	605.46
410	124.6	608.42
420	132.51	620.63
430	141.22	628.96
440	166.33	629.27
450	191.44	629.58
460	216.55	629.89

Comparison Graphs

Specification 2: Standalone vs Two-Node Cluster with Load Balancer





Physical Appliance

Server Location	Client Location
San Jose	us-central1-a

CipherTrust Manager Configuration

System Volume	Memory	CPUs	NICs
2 TB	16 GB	1 with 4 Cores	1

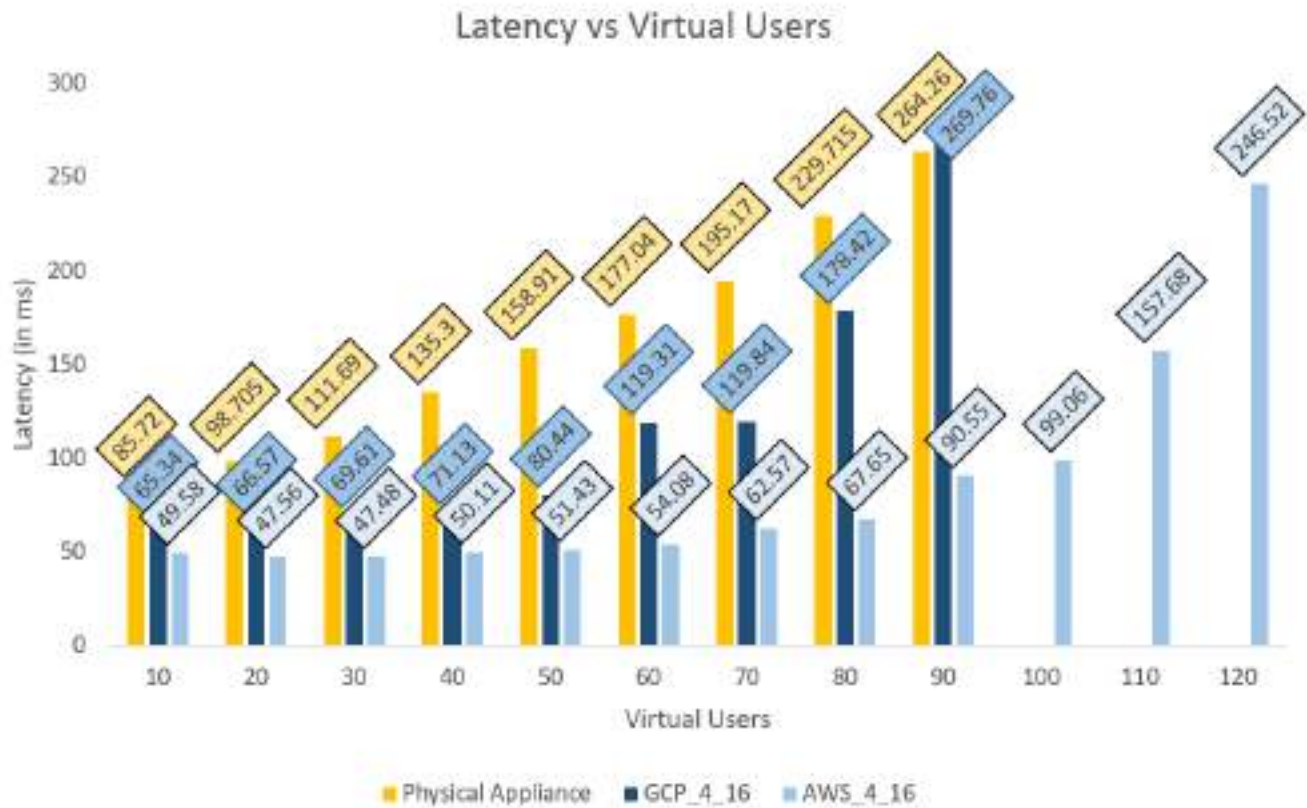
Simulated the **wrap** requests for Google Drive on the CipherTrust Manager deployed on a physical appliance using the k6 tool. The following table shows the handled number of requests per second (RPS), with approximate latency, and the number of virtual users for different data sample for three runs.

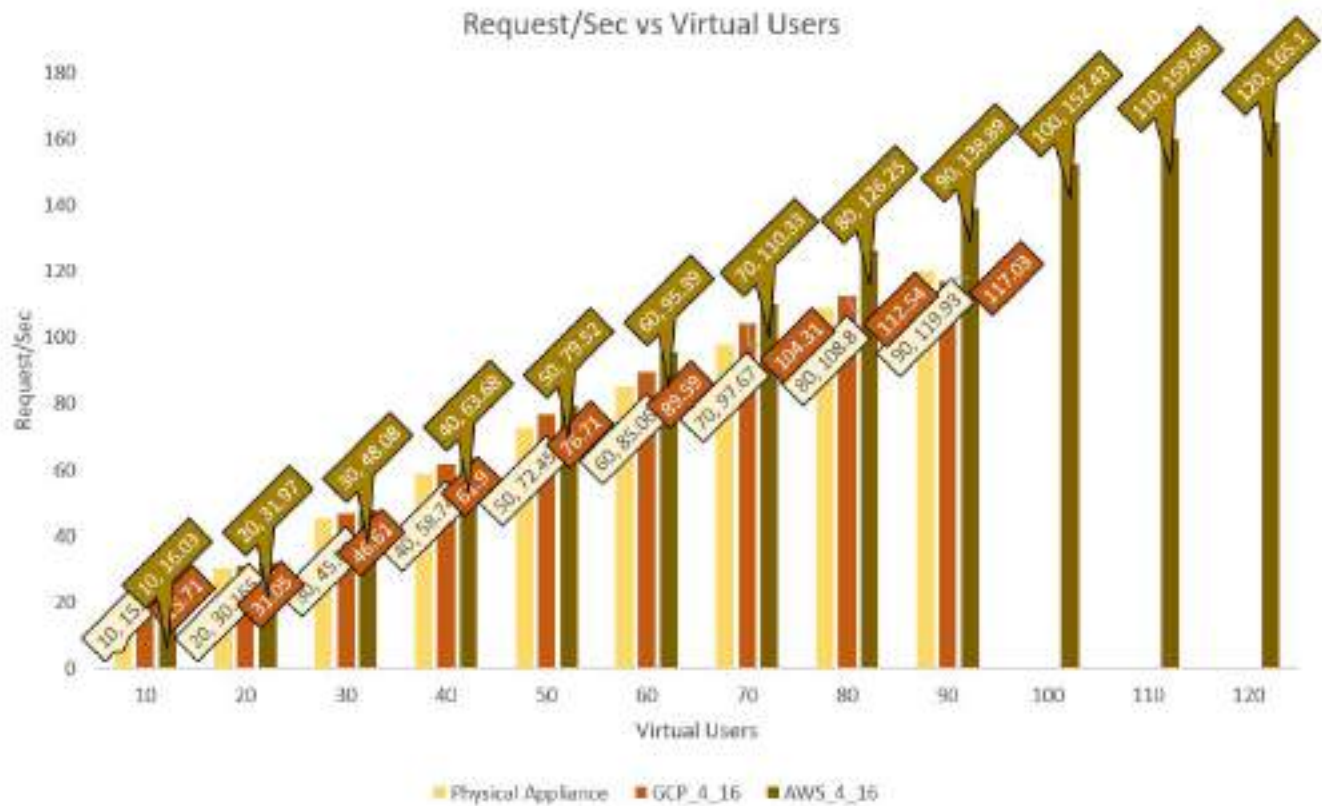
Users	Latency (in ms)	Requests/Second
10	85.72	15.3
20	98.705	30.165
30	111.69	45.03
40	135.3	58.74
50	158.91	72.45
60	177.04	85.06
70	195.17	97.67
80	229.715	108.8

Users	Latency (in ms)	Requests/Second
90	264.26	119.93

Comparison Graphs

Physical Appliance vs AWS Cloud vs Google Cloud





Recommendations

Assumption

Each user has **3** documents open and is editing them resulting in an autosave every **30** seconds, that is, **3/30=0.1** transactions per second (tps) per user.

Number of users = (throughput for around 200 ms latency)/0.1

Google Cloud

1. Response time compliance of around **200** ms was met for a maximum throughput of **112.54** operations per second with a standalone CipherTrust Manager k170v instance with **4** CPUs and **16** GB RAM.

The approximate number of users this configuration can handle is **1125**.

2. Response time compliance of around **200** ms was met for a maximum throughput of **198.03** operations per second with a two-node CipherTrust Manager k170v cluster (each node with **4** CPUs and **16** GB RAM) connected with a Google Cloud load balancer.

The approximate number of users this configuration can handle is **1980**.

3. Response time compliance of around **200** ms was met for a maximum throughput of **231.91** operations per second with a standalone CipherTrust Manager k470v instance with **8** CPUs and **64** GB RAM.

The approximate number of users this configuration can handle is **2319**.

4. Response time compliance of around **200** ms was met for a maximum throughput of **459.62** operations per second with a two-node CipherTrust Manager k470v cluster (each node with **8** CPUs and **64** GB RAM) connected with a Google Cloud load balancer.

The approximate number of users this configuration can handle is **4596**.

AWS Cloud

1. Response time compliance of around **200** ms was met for a maximum throughput of **159.96** operations per second with a standalone CipherTrust Manager k170v instance with **4** CPUs and **16** GB RAM.

The approximate number of users this configuration can handle is **1600**.

2. Response time compliance of around **200** ms was met for a maximum throughput of **304.81** operations per second with a two-node CipherTrust Manager k170v cluster (each node with **4** CPUs and **16** GB RAM) connected with an AWS load balancer.

The approximate number of users this configuration can handle is **3048**.

3. Response time compliance of around **200** ms was met for a maximum throughput of **324.27** operations per second with a standalone CipherTrust Manager k470v instance with **8** CPUs and **64** GB RAM.

The approximate number of users this configuration can handle is **3243**.

4. Response time compliance of around **200** ms was met for a maximum throughput of **629.58** operations per second with a two-node CipherTrust Manager k470v cluster (each node with **8** CPUs and **64** GB RAM) connected with an AWS load balancer.

The approximate number of users this configuration can handle is **6296**.

Conclusion

The number of users and throughput almost doubles up on moving from CipherTrust Manager k170v to k470v. Moreover, adding an additional node to the cluster also doubles up the throughput. Overall, a performance gain of 400 percent is achieved by moving from a standalone CipherTrust Manager k170v to a two-node CipherTrust Manager k470v cluster.

Google Meet

This section provides the performance summary of the Google Meet service managed through CCKM on the CipherTrust Manager server.

Requests Per Second

The following sections list the wrap requests per second, approximate latency, and the number of virtual users for different deployment scenarios.

Note

To provide a good user experience, Google recommends a maximum latency of 200 ms (for 99% of the requests). Therefore, the performance numbers on this page are based on a latency of around 200 ms.

- Google Cloud
- AWS Cloud
- Physical Appliance

Google Cloud

Server Location	Client Location
us-central1-a	us-central1-a

Simulated the **wrap** requests for Google Meet on the CipherTrust Manager deployed on Google Cloud Platform using the k6 tool. The following table shows the handled number of requests per second (RPS), with approximate latency, and the number of virtual users for different data samples on a standalone CipherTrust Manager and a two-node CipherTrust Manager cluster connected with a load balancer.

Click a tab to view performance numbers based on two different specifications.

Specification 1

System Volume	Memory	CPUs	NICs
50 GB	16 GB	4	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	20.05	17.020454
20	25.03	33.965335
30	26.66	50.521666
40	35.21	66.966574
50	40.49	83.158219
60	59.01	98.682035
70	97.44	111.418039
80	133.97	123.70189
90	176.21	135.2053

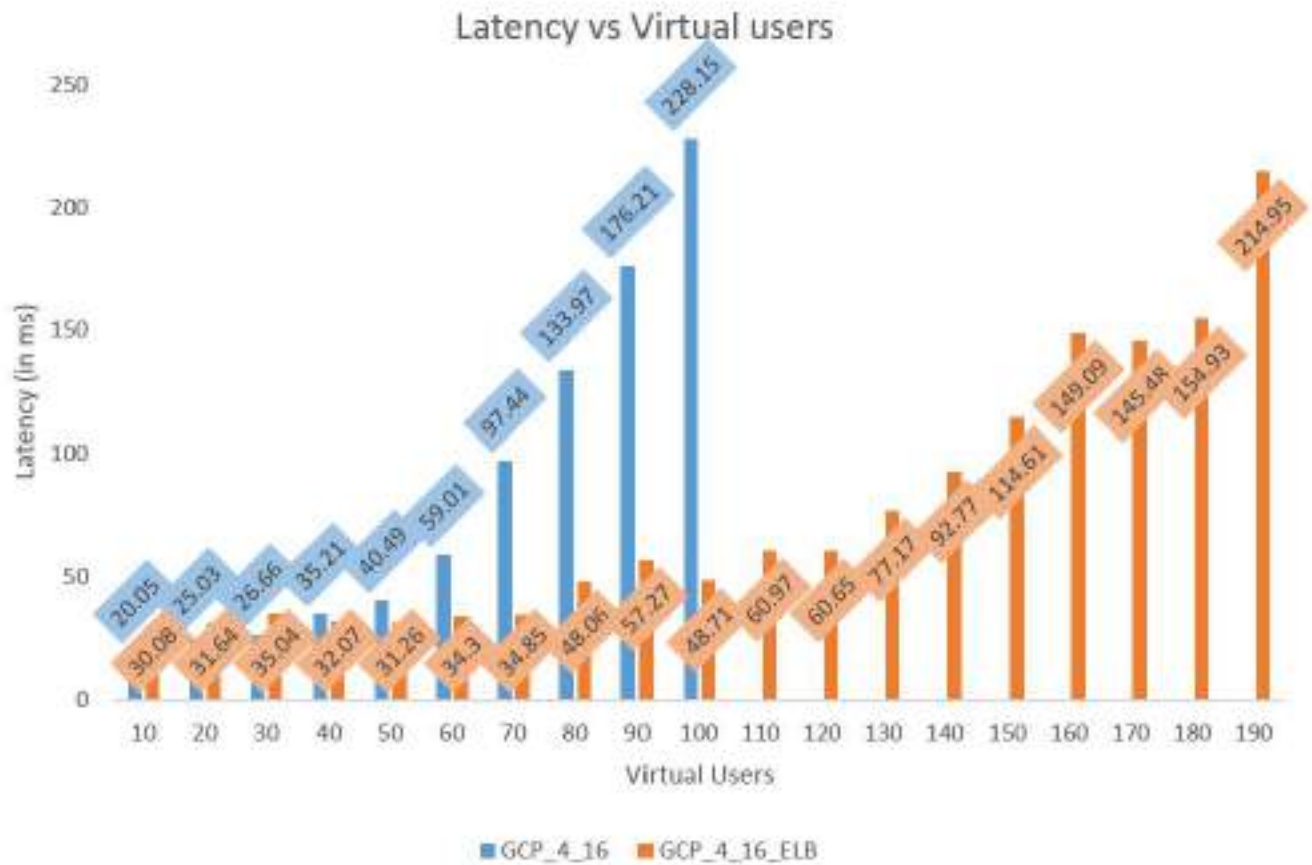
Users	Latency (in ms)	Requests/Second
100	228.15	144.659485

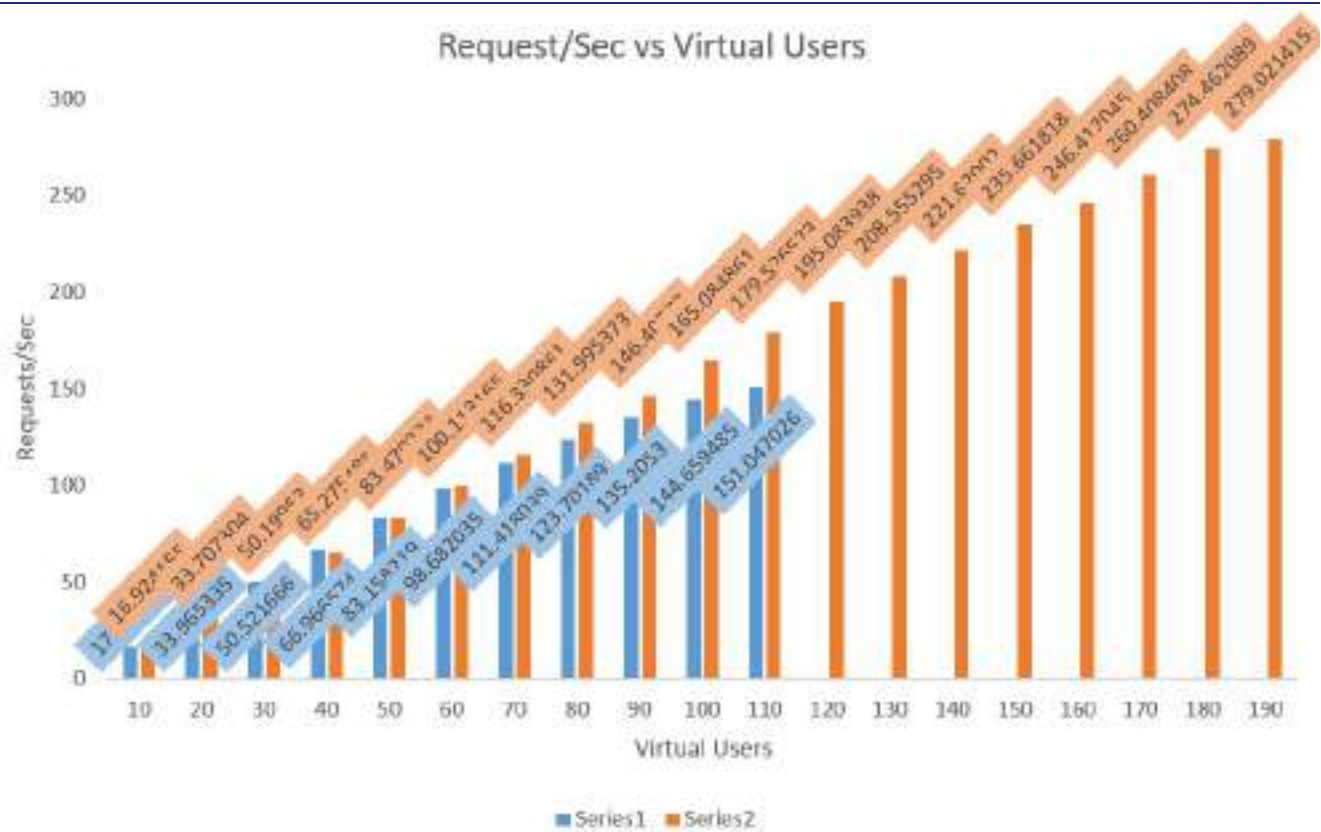
Two-Node Cluster with Load Balancer

Users	Latency (in ms)	Requests/Second
10	30.08	16.924155
20	31.64	33.707304
30	35.04	50.19953
40	32.07	65.275485
50	31.26	83.479334
60	34.3	100.113165
70	34.85	116.330861
80	48.06	131.995373
90	57.27	146.40572
100	48.71	165.084861
110	60.97	179.526573
120	60.65	195.083938
130	77.17	208.555295
140	92.77	221.62002
150	114.61	235.661818
160	149.09	246.412045
170	145.48	260.408408
180	154.93	274.462089
190	214.95	279.021415

Comparison Graphs

Specification 1: Standalone vs Two-Node Cluster with Load Balancer





Specification 2

System Volume	Memory	CPUs	NICs
50 GB	64 GB	8	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	17.88	17.06529
20	19.73	33.744022
30	21.58	50.422754
40	22.09	67.2538575
50	22.6	84.084961
60	23.285	101.1174825
70	23.97	118.150004
80	25.61	134.7643055
90	27.25	151.378607
100	30.03	167.515104

Users	Latency (in ms)	Requests/Second
110	32.81	183.651601
120	40.725	199.4530875
130	48.64	215.254574
140	56.685	229.612928
150	64.73	243.971282
160	93.945	254.7707775
170	123.16	265.570273
180	122.315	280.222493
190	121.47	294.874713
200	140.825	304.60713
210	160.18	314.339547
220	185.175	322.6192025
230	210.17	330.898858

Two-Node Cluster with Load Balancer

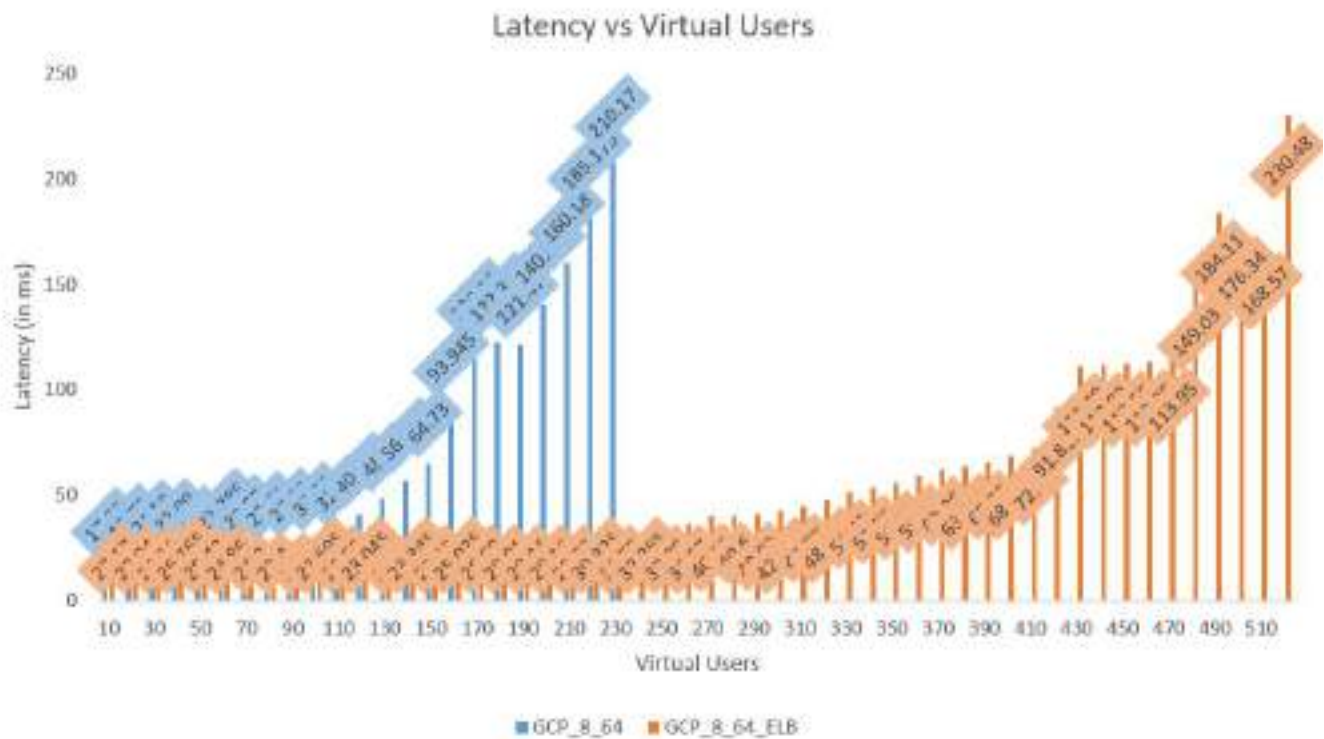
Users	Latency (in ms)	Requests/Second
10	27.17	16.953615
20	27.04	33.779457
30	26.91	50.605299
40	26.165	67.53955
50	25.42	84.473801
60	24.86	101.253739
70	24.3	118.033677
80	23.4	134.5760395
90	22.5	151.118402
100	22.695	168.0278065
110	22.89	184.937211
120	23.045	202.25405
130	23.2	219.570889
140	23.345	236.5859305
150	23.49	253.600972
160	25.035	269.5941965
170	26.58	285.587421

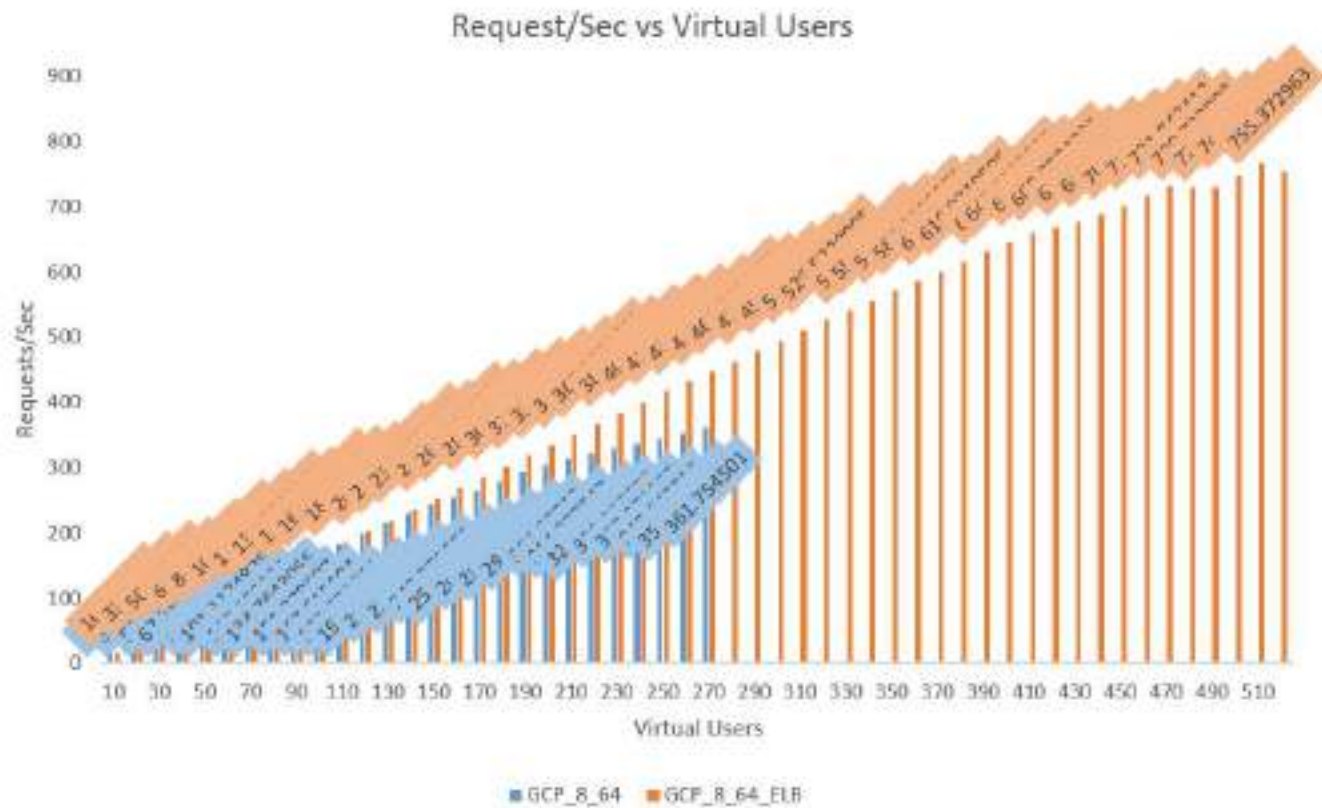
Users	Latency (in ms)	Requests/Second
180	28.01	302.063464
190	29.44	318.539507
200	29.16	335.178603
210	28.88	351.817699
220	30.325	367.5871685
230	31.77	383.356638
240	32.265	400.296817
250	32.76	417.236996
260	36.57	432.212196
270	40.38	447.187396
280	40.6	463.0674135
290	40.82	478.947431
300	42.795	495.239256
310	44.77	511.531081
320	48.095	526.5320905
330	51.42	541.5331
340	53.57	556.861547
350	55.72	572.189994
360	59.04	586.2393975
370	62.36	600.288801
380	63.795	616.0219505
390	65.23	631.7551
400	68.685	645.9995005
410	72.14	660.243901
420	91.815	668.2584325
430	111.49	676.272964
440	112.02	688.700369
450	112.55	701.127774
460	113.25	716.485093
470	113.95	731.842412
480	149.03	730.723998
490	184.11	729.605584
500	176.34	748.613218

Users	Latency (in ms)	Requests/Second
510	168.57	767.620852
520	230.48	755.372963

Comparison Graphs

Specification 2: Standalone vs Two-Node Cluster with Load Balancer





AWS Cloud

Server Location	Client Location
us-east-1b	us-central1-a

Simulated the **wrap** requests for Google Meet on the CipherTrust Manager deployed on AWS cloud using the k6 tool. The following table shows the handled number of requests per second (RPS), with approximate latency, and the number of virtual users for different data samples on a standalone CipherTrust Manager and a two-node CipherTrust Manager cluster connected with a load balancer.

Click a tab to view performance numbers based on two different specifications.

Specification 1

System Volume	Memory	CPUs	NICs
50 GB	16 GB	4	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	40.74	16.462869

Users	Latency (in ms)	Requests/Second
20	41.405	32.643829
30	42.07	48.824789
40	42.99	64.9318095
50	43.91	81.03883
60	45.02	96.9078885
70	46.13	112.776947
80	50.8	128.259757
90	55.47	143.742567
100	59.32	159.43998
110	63.17	175.137393
120	86.17	187.6379445
130	109.17	200.138496
140	132.855	212.1405135
150	156.54	224.142531
160	171.395	235.5833245
170	186.25	247.024118
180	233.91	252.1894955

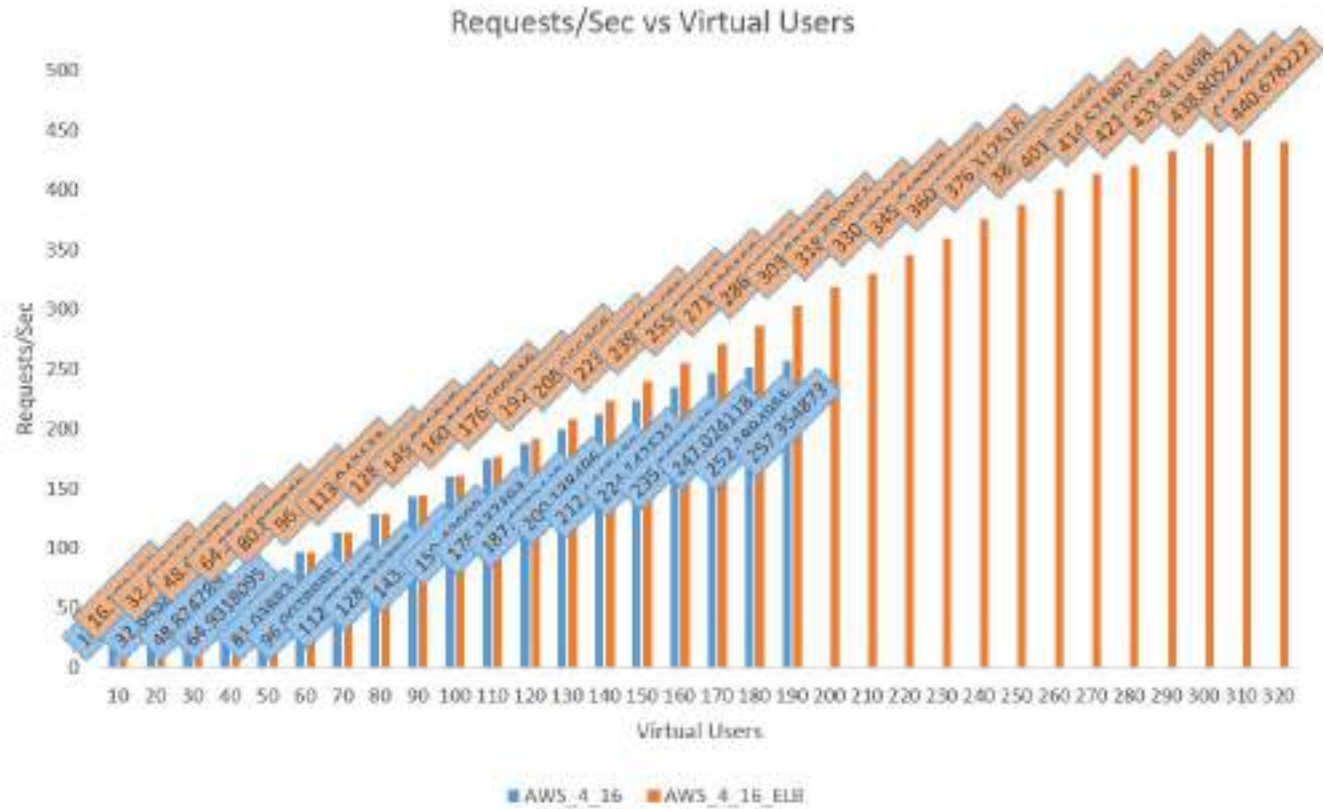
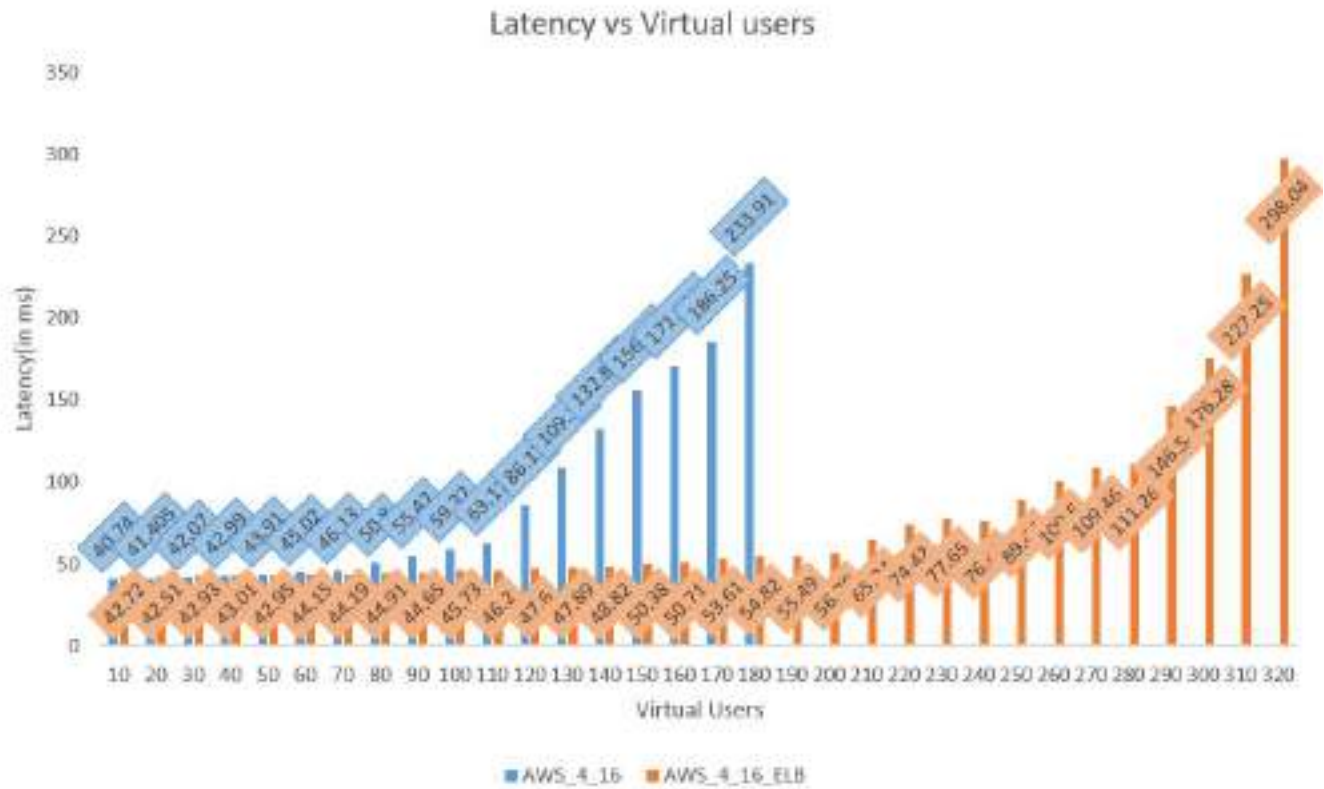
Two-Node Cluster with Load Balancer

Users	Latency (in ms)	Requests/Second
10	42.72	16.175128
20	42.51	32.420987
30	42.93	48.446633
40	43.01	64.355091
50	42.95	80.807762
60	44.15	96.639114
70	44.19	113.047133
80	44.91	128.65812
90	44.65	145.081845
100	45.73	160.505509
110	46.2	176.699848
120	47.6	192.15855
130	47.89	208.666368

Users	Latency (in ms)	Requests/Second
140	48.82	223.88651
150	50.38	239.599473
160	50.71	255.035196
170	53.61	271.573122
180	54.82	286.369364
190	55.49	303.581732
200	56.79	318.599251
210	65.34	330.573892
220	74.42	345.847797
230	77.65	360.133676
240	76.49	376.312516
250	89.46	387.2406
260	100.6	401.232406
270	109.46	414.571807
280	111.26	421.506349
290	146.54	433.911498
300	176.28	438.805221
310	227.25	442.49615

Comparison Graphs

Specification 1: Standalone vs Two-Node Cluster with Load Balancer



Specification 2

System Volume	Memory	CPUs	NICs
50 GB	64 GB	8	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

Users	Latency (in ms)	Requests/Second
10	37.78	16.322682
30	37.18	48.820999
50	37.3	81.370941
70	37.88	113.599354
90	38	146.272925
110	38.09	179.040711
130	40.53	210.124152
150	39.86	243.147436
170	43.51	274.400142
190	43.16	306.649272
210	63.38	336.642994
230	82.02	363.879907
250	71.4	395.708195
270	76	425.888079
290	89.94	452.540139
310	107.85	477.489702
330	139.12	498.423816
350	135.27	526.025136
370	191.95	533.67742
390	203.22	559.302403

Two-Node Cluster with Load Balancer

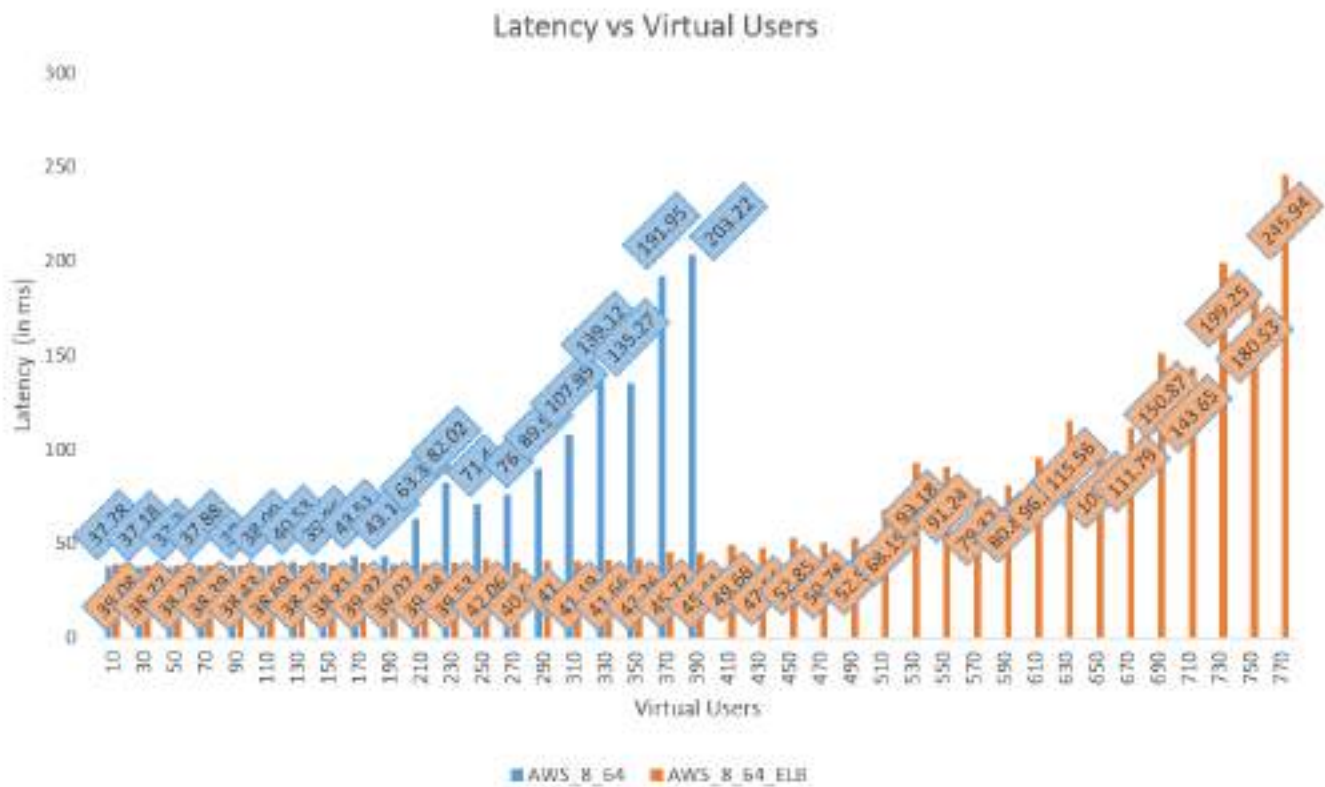
Users	Latency (in ms)	Requests/Second
10	39.08	16.436472
30	38.72	48.568014
50	38.29	81.638752
70	38.39	113.121597

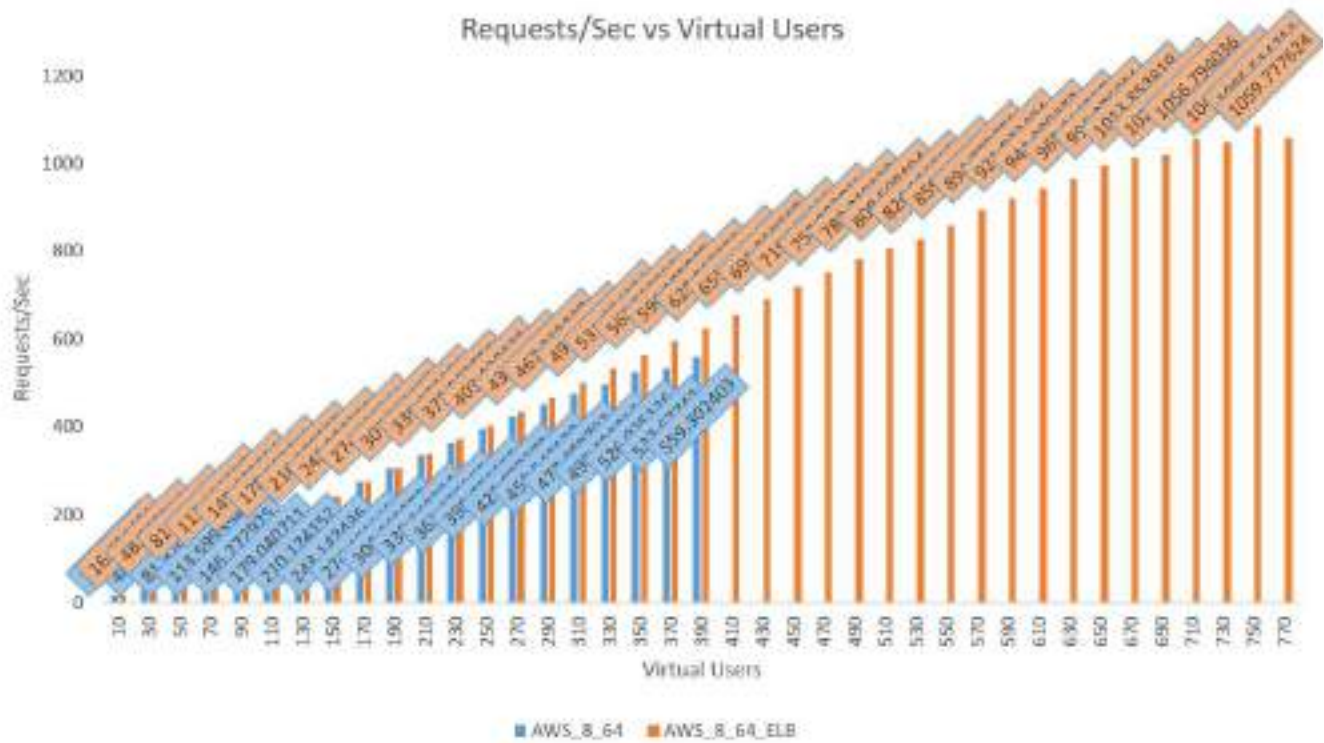
Users	Latency (in ms)	Requests/Second
90	38.43	145.660924
110	38.69	178.108611
130	38.75	210.357358
150	38.81	242.386283
170	39.92	274.120341
190	39.02	307.585027
210	39.38	339.976054
230	39.53	373.269723
250	42.06	403.133135
270	40.5	435.59251
290	41	467.333349
310	41.19	499.85638
330	41.66	531.694894
350	42.26	563.708383
370	45.77	596.375765
390	45.41	625.913359
410	49.66	655.925265
430	47.56	691.350867
450	52.85	719.563383
470	50.78	753.029761
490	52.99	783.212132
510	68.15	808.508404
530	93.18	826.906719
550	91.24	859.003234
570	79.32	894.770676
590	80.85	921.922404
610	96.21	944.098372
630	115.56	965.814237
650	103.16	996.276651
670	111.79	1013.853819
690	150.87	1020.147071
710	143.65	1056.794036
730	199.25	1048.523167

Users	Latency (in ms)	Requests/Second
750	180.53	1085.514714
770	245.94	1059.777624

Comparison Graphs

Specification 2: Standalone vs Two-Node Cluster with Load Balancer





Physical Appliance

Server Location	Client Location
San Jose	us-central1-a

CipherTrust Manager Configuration

System Volume	Memory	CPUs	NICs
2 TB	16 GB	1 with 4 Cores	1

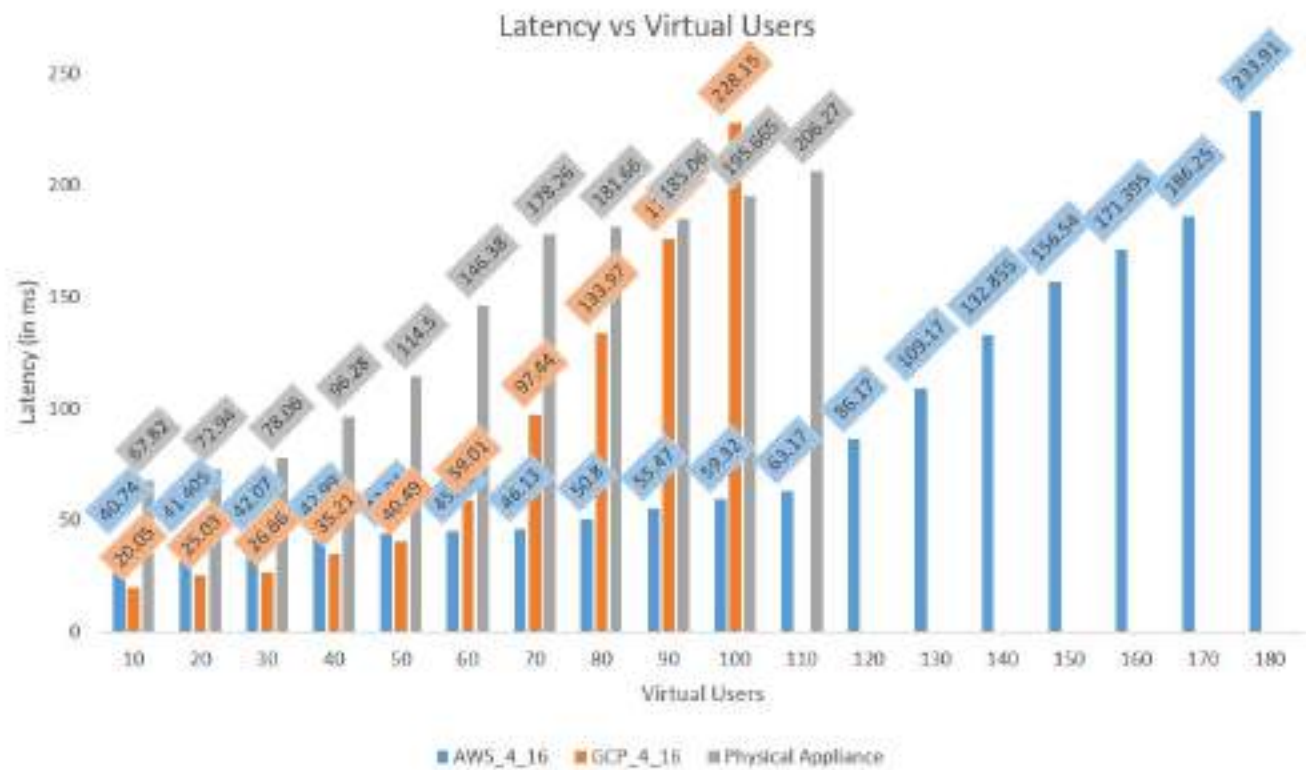
Simulated the **wrap** requests for Google Meet on the CipherTrust Manager deployed on a physical appliance using the k6 tool. The following table shows the handled number of requests per second (RPS), with approximate latency, and the number of virtual users for different data sample for three runs.

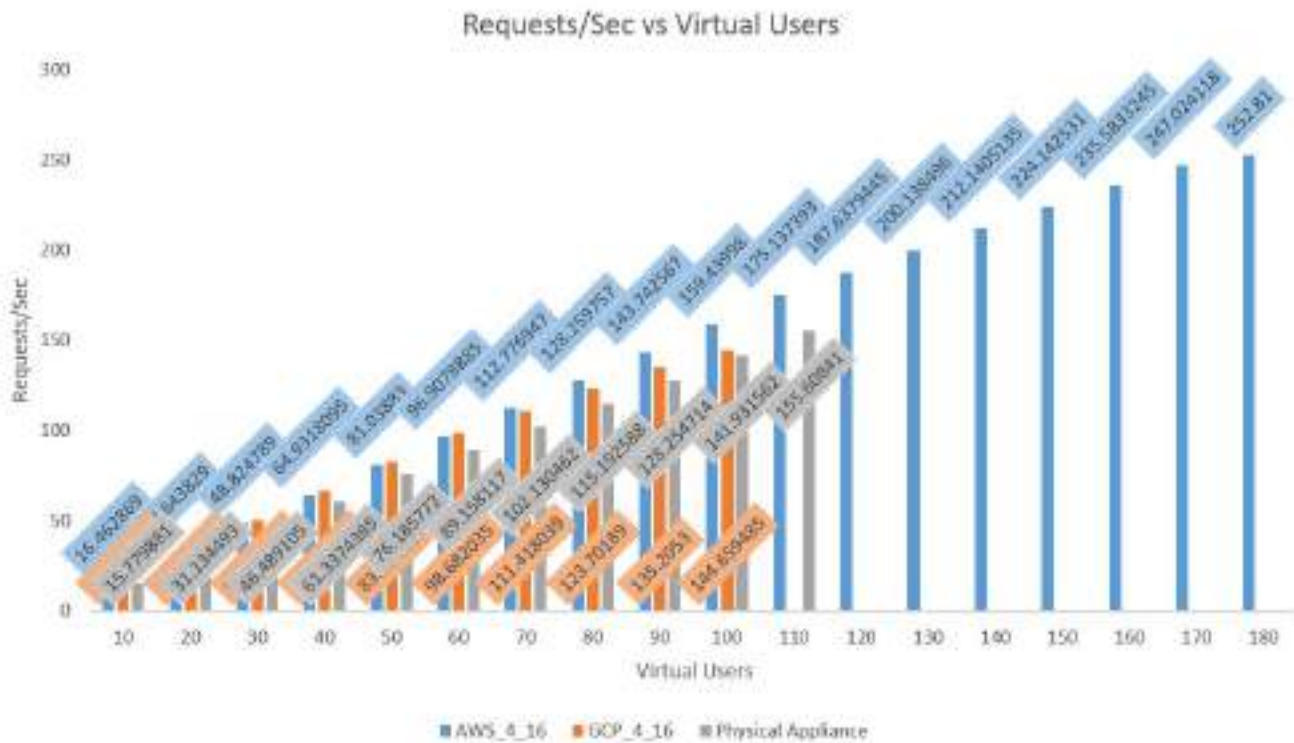
Users	Latency (in ms)	Requests/Second
10	67.82	15.779881
20	72.94	31.134493
30	78.06	46.489105
40	96.28	61.3374385
50	114.5	76.185772
60	146.38	89.158117
70	178.26	102.130462

Users	Latency (in ms)	Requests/Second
80	181.66	115.192588
90	185.06	128.254714
100	195.665	141.931562
110	206.27	155.60841

Comparison Graphs

Physical Appliance vs AWS Cloud vs Google Cloud





Recommendations

Note

When an encrypted Google Meet call is started, a **wrap** request is initiated. Similarly, when a participant joins the call, an **unwrap** request is initiated.

Google Cloud

1. Response time compliance of around **200 ms** was met for a maximum throughput of **144.66** operations per second with a standalone CipherTrust Manager k170v instance with **4 CPUs** and **16 GB RAM**.
2. Response time compliance of around **200 ms** was met for a maximum throughput of **279.02** operations per second with a two-node CipherTrust Manager k170v cluster (each node with **4 CPUs** and **16 GB RAM**) connected with a Google Cloud load balancer.
3. Response time compliance of around **200 ms** was met for a maximum throughput of **330.899** operations per second with a standalone CipherTrust Manager k470v instance with **8 CPUs** and **64 GB RAM**.
4. Response time compliance of around **200 ms** was met for a maximum throughput of **755.37** operations per second with a two-node CipherTrust Manager k470v cluster (each node with **8 CPUs** and **64 GB RAM**) connected with a Google Cloud load balancer.

AWS Cloud

1. Response time compliance of around **200 ms** was met for a maximum throughput of **252.19** operations per second with a standalone CipherTrust Manager k170v instance with **4 CPUs** and **16 GB RAM**.
2. Response time compliance of around **200 ms** was met for a maximum throughput of **442.50** operations per second with a two-node CipherTrust Manager k170v cluster (each node with **4 CPUs** and **16 GB RAM**) connected with an AWS load balancer.
3. Response time compliance of around **200 ms** was met for a maximum throughput of **559.30** operations per second with a standalone CipherTrust Manager k470v instance with **8 CPUs** and **64 GB RAM**.
4. Response time compliance of around **200 ms** was met for a maximum throughput of **1059.78** operations per second with a two-node CipherTrust Manager k470v cluster (each node with **8 CPUs** and **64 GB RAM**) connected with an AWS load balancer.

Conclusion

The number of users and throughput almost doubles up on moving from CipherTrust Manager k170v to k470v. Moreover, adding an additional node to the cluster also doubles up the throughput. Overall, a performance gain of 400 percent is achieved by moving from a standalone CipherTrust Manager k170v to a two-node CipherTrust Manager k470v cluster.

Gmail

This section provides the performance summary of the Gmail service managed through CCKM on the CipherTrust Manager server.

Requests Per Second

Note

To provide a good user experience, Google recommends a maximum latency of 200 ms (for 99% of the requests). Therefore, the performance numbers on this page are based on a latency of around 200 ms.

Google Cloud

Server Location	Client Location
us-central1-a	us-central1-a

Simulated the **wrap** requests for Gmail on the CipherTrust Manager deployed on Google Cloud Platform using the k6 tool. The following table shows the handled number of requests per second (RPS), with approximate latency, and the number of virtual users for different data samples on a standalone CipherTrust Manager and a two-node CipherTrust Manager cluster connected with a load balancer.

Click a tab to view performance numbers based on two different specifications.

Specification 1

System Volume	Memory	CPUs	NICs
50 GB	16 GB	4	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

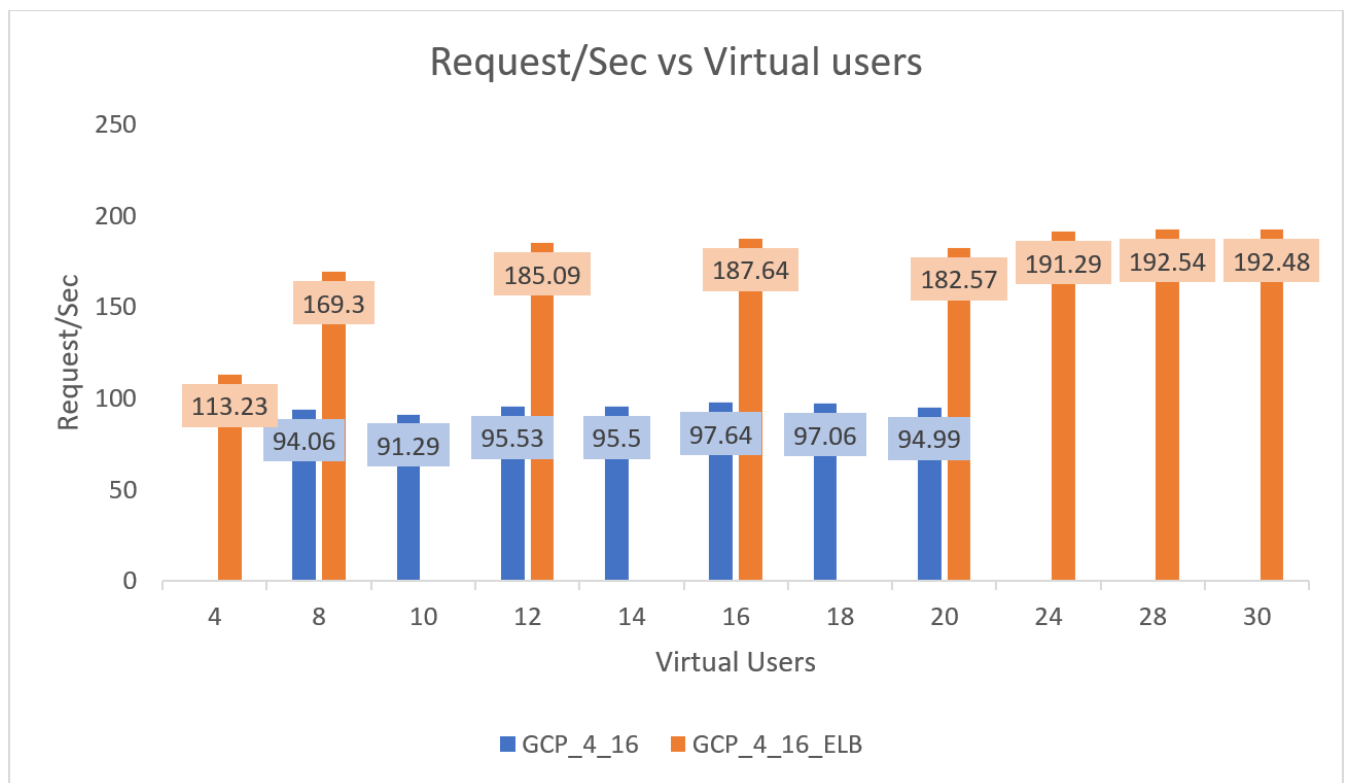
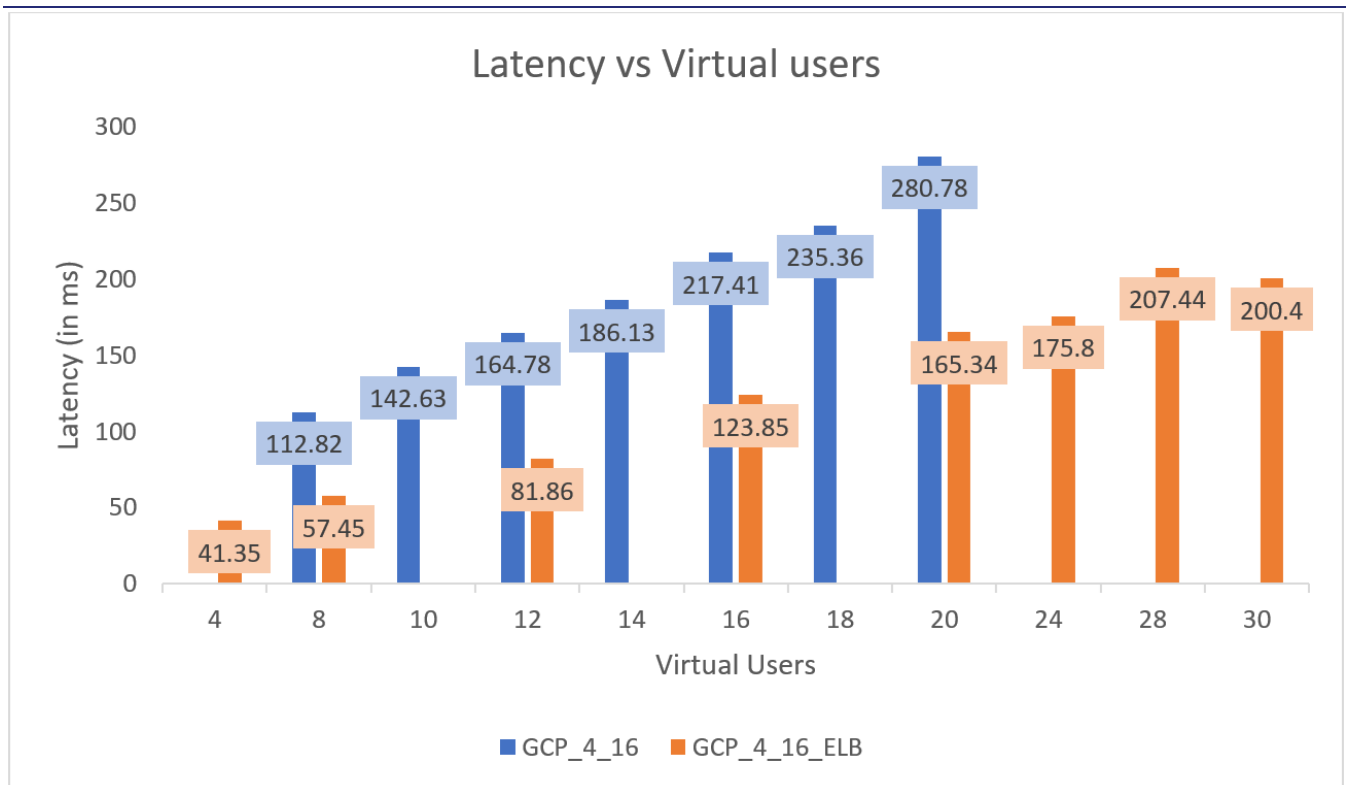
Users	Latency (in ms)	Requests/Second
8	112.82	94.06
10	142.63	91.29
12	164.78	95.53
14	186.13	95.50
16	217.41	97.64
18	235.36	97.06
20	280.78	94.99

Two-Node Cluster with Load Balancer

Users	Latency (in ms)	Requests/Second
4	41.35	113.23
8	57.45	169.3
12	81.86	185.09
16	123.85	187.64
20	165.34	182.57
24	175.80	191.29
28	207.44	192.54
30	200.4	192.48

Comparison Graphs

Specification 1: Standalone vs Two-Node Cluster with Load Balancer



Specification 2

System Volume	Memory	CPUs	NICs
50 GB	64 GB	8	1

Click a tab to view performance numbers for a standalone CipherTrust Manager or a two-node cluster with a load balancer.

Standalone

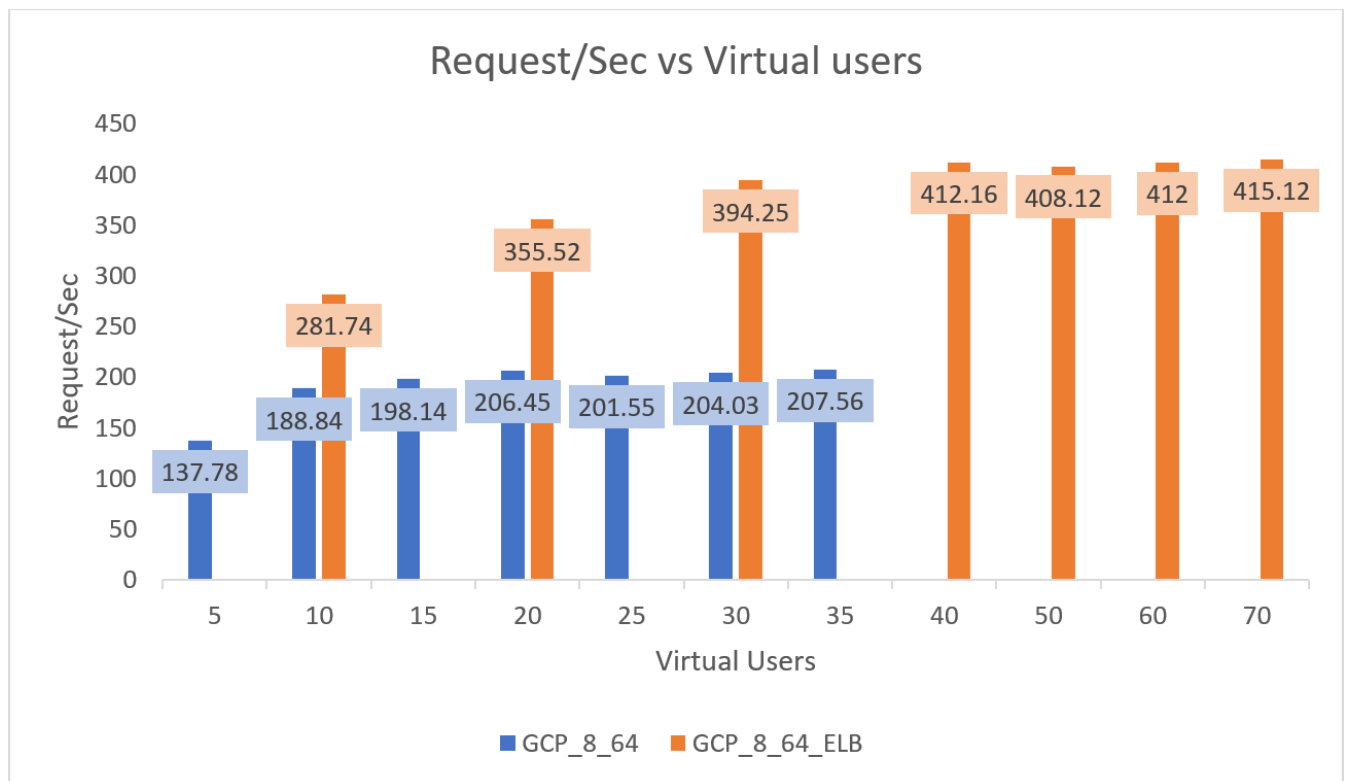
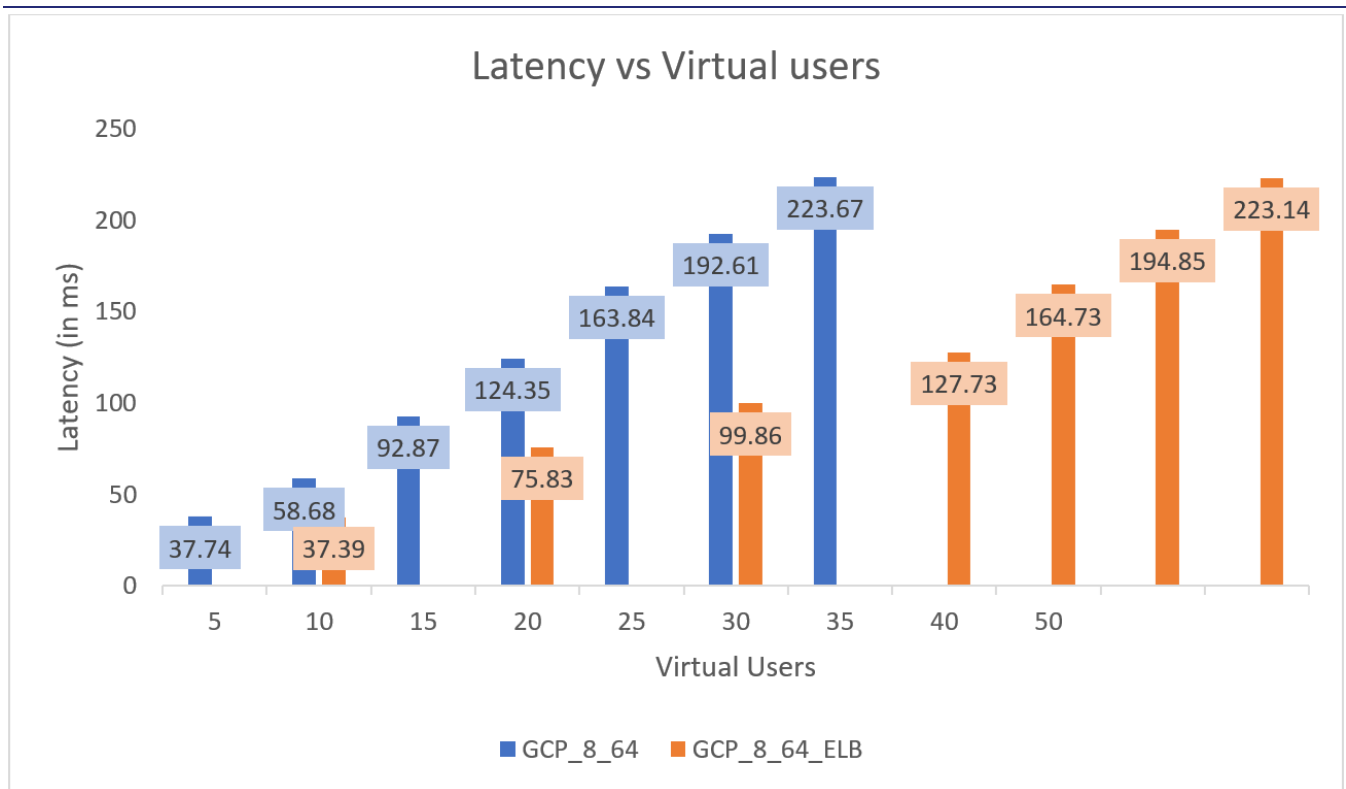
Users	Latency (in ms)	Requests/Second
5	37.74	137.78
10	58.68	188.84
15	92.87	198.14
20	124.35	206.45
25	163.84	201.55
30	192.61	204.03
35	223.67	207.56

Two-Node Cluster with Load Balancer

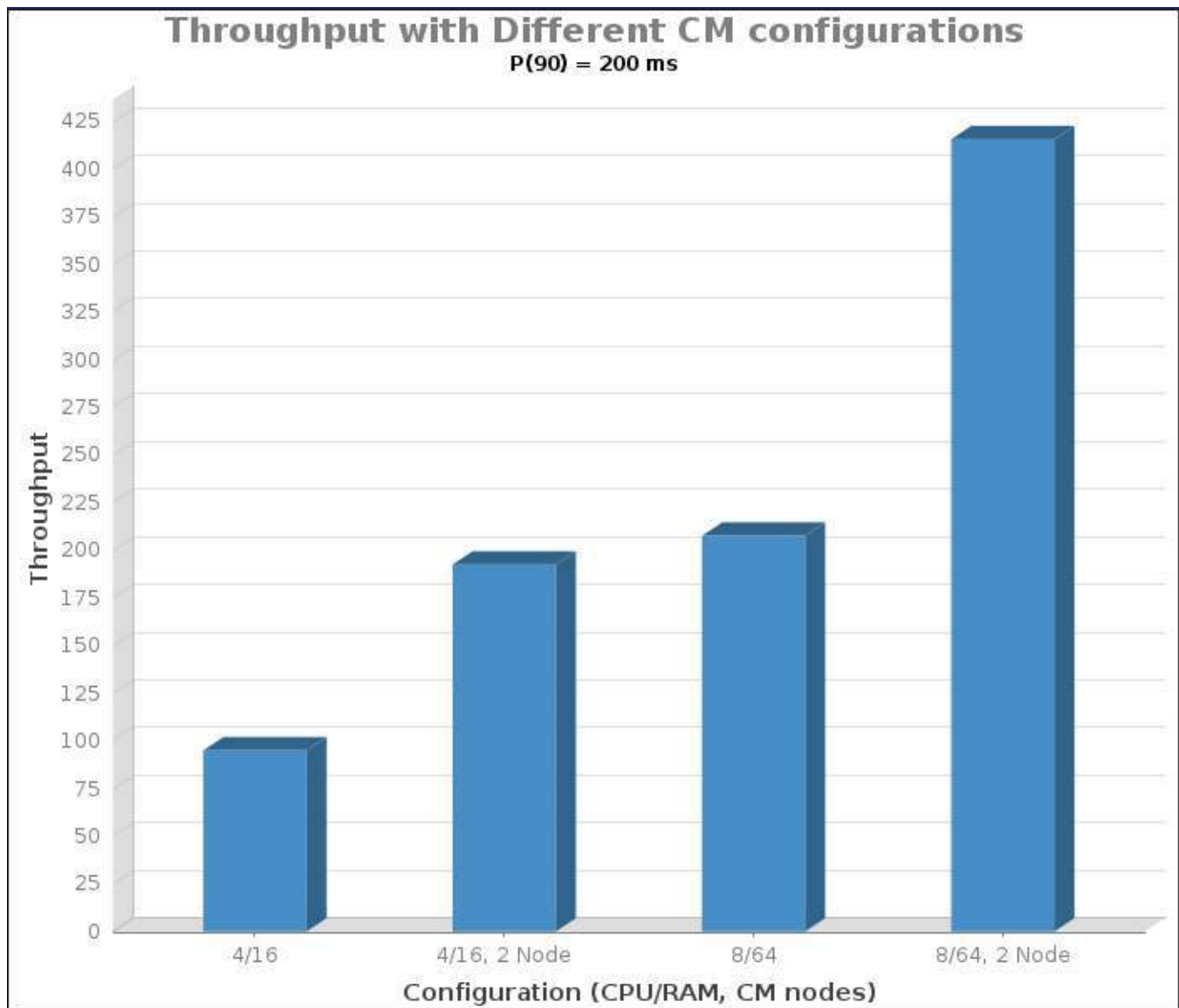
Users	Latency (in ms)	Requests/Second
10	37.39	281.74
20	75.83	355.52
30	99.86	394.25
40	127.73	412.16
50	164.73	408.12
60	194.85	412.00
70	223.14	415.12

Comparison Graphs

Specification 2: Standalone vs Two-Node Cluster with Load Balancer



Throughput Configuration



Recommendations

Note

When an encrypted email is sent, a **wrap** request is initiated. Similarly, when a participant opens the email, an **unwrap** request is initiated.

Google Cloud

1. Response time compliance of around **200 ms** was met for a maximum throughput of **97.64** operations per second with a standalone CipherTrust Manager k170v instance with **4 CPUs** and **16 GB RAM**.
2. Response time compliance of around **200 ms** was met for a maximum throughput of **192.48** operations per

second with a two-node CipherTrust Manager k170v cluster (each node with **4** CPUs and **16** GB RAM) connected with a Google Cloud load balancer.

- Response time compliance of around **200** ms was met for a maximum throughput of **207.56** operations per second with a standalone CipherTrust Manager k470v instance with **8** CPUs and **64** GB RAM.
- Response time compliance of around **200** ms was met for a maximum throughput of **415.12** operations per second with a two-node CipherTrust Manager k470v cluster (each node with **8** CPUs and **64** GB RAM) connected with a Google Cloud load balancer.

Conclusion

The number of users and throughput almost doubles up on moving from CipherTrust Manager k170v to k470v. Moreover, adding an additional node to the cluster also doubles up the throughput. Overall, a performance gain of 400 percent is achieved by moving from a standalone CipherTrust Manager k170v to a two-node CipherTrust Manager k470v cluster.

Key Material Export and Upload

This section describes how to prevent CCKM users from exporting source key material. The section also describes how to allow CCKM users other than key owners to upload source key material to the cloud.

Preventing CCKM Users from Exporting Key Material

When a CCKM user creates a new source key on the CipherTrust Manager and uploads it to the cloud, then, as a key owner, the CCKM user has all permissions on the source key. So, the CCKM user can export the source key material after upload.

To prevent the CCKM user from exporting the source key material:

- Create a custom group, for example, `<deny-export-policy-group>`. This group will be denied permission to export the source key material through a policy.

```
POST /api/v1/usermgmt/groups
{
  "name": "<deny-export-policy-group>"
}
```

- Create a policy, for example, `<deny-export-policy>` to deny the source key export.

```
POST /api/v1/admin/policies
{
  "name": "<deny-export-policy>",
  "allow": true,
  "effect": "deny",
  "actions": [
```

```

    "ExportKey"
  ]
}

```

3. Attach the policy, for example, `<deny-export-policy>` to the custom group.

```

POST /api/v1/admin/policy-attachments
{
  "policy": "<deny-export-policy>",
  "principalSelector": {
    "cust": {
      "groups": [
        "<deny-export-policy-group>"
      ]
    }
  }
}

```

4. Add the CCKM user, `<cckm-user>`, to the custom group.

```

POST /api/v1/usermgmt/groups/<deny-export-policy-group>/users/<cckm-user>
{
  "name": "<deny-export-policy-group>",
  "created_at": "<timestamp>",
  "updated_at": "<timestamp>"
}

```

The CCKM user can no longer export the key material.

Permitting CCKM Users to Upload Key Material

When the CCKM user is not the key owner, grant the read key and export/upload key permission on the source key to the CCKM Users group or the custom group.

```

PATCH /api/v1/vault/keys2/<key-id>
{
  "meta": {
    "ownerId": "<owner-ID>",
    "permissions": {
      "ReadKey": [
        "<group-name>"
      ],
      "UploadKey": [
        "<group-name>"
      ]
    }
  }
}

```

```
}
}
```

The CCKM user can upload the source key material successfully.

Luna HSM Resources

This section describes prerequisites to manage Luna HSM resources on CCKM.

Note

The information presented in this section also applies to Azure Dedicated HSM.

Luna Network HSM Compatibility

We have tested specific combinations of Luna 7.x firmware and Luna 7.x software. Consult [Luna Network HSM Customer Release Notes](#) for the current FIPS certified Luna firmware versions.

Appliance Software	HSM firmware
7.2	7.2.0
7.2	7.0.3
7.3	7.3.3
7.4	7.4.0
7.4	7.3.3
7.7.0	7.7.0
7.7.0	7.3.3
7.7.1	7.7.1
7.8.4	7.8.4
7.9.0	7.9.0

Note

- The above compatibilities apply for connections using the default Network Trust Link (NTLS) between the CipherTrust Manager and the Luna Network HSM. Secure Trusted Channel (STC) connections are only supported for HSM firmware version 7.7.0. or higher.
- The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

This release supports the following Luna HSM features:

- PED and Password-based HSM configurations.
- Both V0 and V1 partitions.
- Import of RSA-4096 keys from Luna Network HSM Software and Firmware versions 7.4.x and higher.
- Symmetric and asymmetric keys.

Prerequisites

Note

- Depending on the use case, the Luna HSM partitions can be configured in the Key Export or Clone mode. For AWS HYOK operations, you can configure the partitions in either mode. For BYOK operations, the partitions must be configured in the Key Export mode.
- If the Luna HSM partitions are configured in Clone mode but the connection manager is configured in Key Export mode, the key is created on the primary partition but you receive a wrap key error.
- If the Luna HSM partitions are configured in Export mode but the connection manager is configured in Clone mode, the key is created with an error. The public key will be present on all partitions, but the private key will only be present on the primary partition.

Before proceeding:

- Make sure that a common cipher is enabled on the Luna HSM and the CipherTrust Manager to allow successful connection between them.
- On Luna HSM, disable the HSM NTLS IP check by running the `ntls ipcheck disable` command.

Also, make sure the CipherTrust Manager is registered with the Luna HSM, as described below:

On the Luna HSM

1. Create a Client. Refer to [Luna HSM Client Software Installation](#).

The client certificate needed when creating the *client* on the Luna HSM can be downloaded by clicking **Download Luna Client Cert** on the CipherTrust Manager GUI. When uploading the certificate file to the Luna HSM, the file must have the same name as the internally generated CN. You can use `openssl` (or some other tool) to inspect the certificate CN. The name will look similar to `cckm-client-c2b39a4b-0f02-4be8-b37f-f3cadfc3ac11`.

2. Register the Client with the Luna HSM. Refer to [Multi-Step NTLS Connection Procedure](#).

-
3. Assign a Partition to the Client. Refer to [Client Partition Connections](#).

On the CipherTrust Manager

1. Add the Luna HSM Server to the CipherTrust Manager. Refer to [Adding an Internal Connection \(Server\)](#) under [Connection Manager](#).
2. Add Connection to the Luna HSM Server on the CipherTrust Manager. Refer to [Creating a Luna Connection](#).
3. Test the Connection. Refer to [Testing a New Luna Connection](#). Make sure the "connection_status" is `connection ok`.

The connection test can fail if:

- CipherTrust Manager is not successfully authenticated to the Luna HSM device
- NTLS service is down on the Luna HSM Server
- Luna HSM partition is not assigned to a Luna HSM client

Inspect the logs on the Luna HSM for details.

Now, [Luna HSM partitions](#) and [Luna HSM keys](#) can be managed on the CipherTrust Manager.

Managing Luna HSM Partitions

This section describes how to manage Luna HSM partitions on CCKM.

Before proceeding, make sure to fulfill [prerequisites](#).

Adding Luna HSM Partitions

Warning

Thales strongly discourages adding an Luna HSM partition that contains the CipherTrust Manager's root of trust (RoT) key to CCKM. Do not add a connection to a Luna HSM root-of-trust partition with Connection Manager.

To add a Luna HSM partition to CCKM:

1. Log on to the **CipherTrust Manager** GUI as administrator.
 2. Open the **Cloud Key Manager** application.
 3. In the left pane, click **KMS Containers > Luna HSM Partitions**. The Luna HSM Partitions page is displayed.
 4. Click **Add Partition**. The Add Existing Partition page is displayed.
-

5. From the **Select Connection** drop-down list, select the desired connection.
6. Click **Add**.

The Luna HSM partition is added to CCKM.

A message **partition added successfully...** is displayed on the screen.

Refreshing Luna HSM Keys

Refreshing is the process of downloading keys created on the Luna HSM partitions to CCKM. Refresh can be achieved using any of the following ways:

- [Refreshing Specific Partitions](#)
- [Refreshing All Partitions](#)

Refreshing Specific Partitions

To refresh a specific partition:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Luna HSM Partitions**. The Luna HSM Partitions page is displayed. This page displays the list of Luna HSM partitions.
3. Click the overflow icon () corresponding to the desired Luna HSM partition and click **Refresh Now**.

A message **Refresh started...** is displayed on the screen.

After successful refresh, the refreshed keys are listed on the **Cloud Keys > Luna > Luna Keys** page. Refer to [Viewing Luna HSM Keys](#) for details.

Refreshing All Partitions

To refresh all Luna HSM partitions:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Luna HSM Partitions**. The Luna HSM Partitions page is displayed. This page displays the list of Luna HSM partitions.
3. Click **Refresh All**. The "This may take a while..." message is displayed.

Note

Refreshing all Luna HSM partitions is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Luna > Luna Keys** page. Refer to [Viewing Luna HSM Keys](#) for details.

Syncing All Partitions

Syncing is the process of synchronizing keys in Luna HSM partitions. After successful sync, the partitions have the synced Luna HSM keys. Syncing is useful for Luna HSM partitions in the High Availability (HA) mode.

To synchronize all Luna HSM partitions:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Luna HSM Partitions**. The Luna HSM Partitions page is displayed. This page displays the list of Luna HSM partitions.
3. Click **Sync All**. The "This may take a while..." message is displayed.

Note

Synchronizing all Luna HSM partitions is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Sync All** to continue.

A message **Sync started...** is displayed on the screen. To cancel the sync, click **Cancel Sync**.

The synchronized keys are listed on the **Cloud Keys > Luna > Luna Keys** page. Refer to [Viewing Luna HSM Keys](#) for details.

Viewing/Editing Details of Luna HSM Partitions

The Luna HSM Partitions page shows the list of existing Luna HSM partitions. Search for partitions by Partition ID, Connection, or Label.

Viewing Luna HSM Partitions Details

To view the details of Luna HSM partitions:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Luna HSM Partitions**. The Luna HSM Partitions page displays the following details.

Column	Description
Partition ID	ID of the partition. Click the link to view more details about the partition.
Label	Label of the Luna HSM partition.
Firmware	Firmware running on the Luna HSM partition.
HA Enabled	Whether high availability (HA) is enabled (✓) on the Luna HSM partition. For partitions that do not have HA enabled, ✗ is displayed.
Connection	Name of the connection.
Last Refreshed	When the partition was last refreshed. Never is displayed for partitions that are never refreshed.
Model	Model of the Luna HSM partition. By default, this column is not visible. Click the Customize View (⚙️) icon, select Model , and click OK to display the column.

Note

The **Label** and **Firmware** details will be visible only after the linked Luna HSM connection is in **Ready** state after performing Test Connection (on the CipherTrust Manager GUI).

Modifying Luna HSM Partition Details

To modify the details of a Luna HSM partition:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Luna HSM Partitions**. The Luna HSM Partitions page displays the list of added Luna HSM partitions.
3. Click the overflow icon (⋮) corresponding to the desired Luna HSM partition and click **View/Edit Details**.

You can change the Luna HSM connection and modify user/group permissions on the Luna HSM partition. For details, refer to:

- [Changing the Luna HSM Connection](#)
- [Managing User Permissions on Luna HSM Partitions](#)

Changing the Luna HSM Connection

To change the Luna HSM connection:

1. Expand **Connection**.
2. From the **Connection ID** drop-down list, select the desired Luna HSM connection.
3. Click **Update**.

A message **Updated connection for this partition** is displayed on the screen.

Managing User Permissions on Luna HSM Partitions

To work with Luna HSM, users/groups must have the minimum set of permissions that allow them to use the Luna HSM resources such as Luna HSM keys and partitions. Initially, the user only has permission to view the keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform operations on the Luna HSM partition.

To add permission for a user/group:

1. In the **Access Control** section, click **Assign User/Group**.
2. On the **Assign User/Group** screen, select the user or group to be assigned permissions from the **User/Group** drop-down list.
3. Click **Save**.

The newly added user/group is displayed under **Name** in the **Access Control** section.

CCKM allows the following operations on the Luna HSM partitions:

- View Keys, Create Key, Edit Key, Sync Key, and Delete Key
- Refresh Partition, Sync Partition
- Unassign

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. Select the check box under the desired operation corresponding to the desired users or groups.
 2. Click **Update**.
-

A message **Updated access control for this partition** is displayed on the screen.

Removing a Permission

To remove a permission assigned to a user or group:

1. Clear the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A message **Updated access control for this partition** is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. Under **Unassign**, click the **X** button corresponding to the desired user/group.
2. On the **Unassign User / Unassign Group** screen, click **Unassign**.

Note

Unassigning this user/group will remove all permissions currently assigned to the user/group. Are you sure you want to continue?

3. Click **Unassign**.

A message **Updated access control for this key partition** is displayed on the screen.

Deleting Luna HSM Partitions

To delete a Luna HSM partition:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Luna HSM Partitions**. The Luna HSM Partitions page displays the list of added Luna HSM partitions.
3. Click the overflow icon () corresponding to the desired Luna HSM partition and click **Delete**.
4. On the **Delete Luna HSM Partition** screen, select **I wish to delete this partition**.
5. Click **Delete Partition**.

A message **Luna HSM partition deleted** is displayed on the screen.

Warning

The deleted partition's keys will no longer be available on the Luna Keys page, but the keys will still exist on the Luna HSM. If you later add this partition with the same ID, the keys will be available again.

Managing Luna HSM Keys

This section describes how to manage Luna HSM keys on CCKM. Before proceeding, you must have a Luna HSM partition added to the CCKM. Refer to [Managing Luna HSM Partitions](#) for details.

Note

CCKM supports replication of RSA keys in the High Availability (HA) mode for Luna HSM partitions configured in the Key Export mode.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Adding Luna HSM Keys

To add a Luna HSM key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Luna**. The list of available Luna HSM keys is displayed.
3. Click **Add Key**. The Add Luna Key dialog box is displayed.

Add Luna Key

1. **Select Key Type**, either **Symmetric** or **Asymmetric**.
2. Select the **Partition ID** of the desired Luna HSM.
3. Specify a **Key Label**. This helps in uniquely identifying a key.
4. Select the **Mechanism**. The supported key mechanisms are:

Symmetric

- CKM_AES_KEY_GEN

Asymmetric

- CKM_RSA_FIPS_186_3_AUX_PRIME_KEY_PAIR_GEN
- CKM_RSA_X9_31_KEY_PAIR_GEN
- CKM_RSA_FIPS_186_3_PRIME_KEY_PAIR_GEN
- CKM_RSA_PKCS_KEY_PAIR_GEN
- CKM_EC_KEY_PAIR_GEN

5. Select the **Key Size**. The supported sizes are:

- 2048, 3072, and 4096 for asymmetric keys
- 128, 192, and 256 for symmetric keys

6. (Applicable to the **CKM_EC_KEY_PAIR_GEN** mechanism) Select the **Elliptic Curve**.

- Click **Select**. The Select Luna Curve dialog box is displayed.
- Select the desired **Luna Curve**.
- Click **Select**.

Note

When **HSM policy 12:Allow non-FIPS algorithms** is disabled (FIPS mode):

- Curves with a length of 224 bits or greater can be used for all operations
- Curves ≥ 160 -bits and < 224 -bits can be used for signature verification only
- Curves less than 160-bits are not permitted

These restrictions comply with the recommendations in NIST SP 800-131A Rev2.

7. Selecting the **Key Attributes**.

You can select a set of key attributes. The options are **BYOK Compatible**, **AWS HYOK Compatible**, and **OCI HYOK Compatible**.

You can also select the key attributes. The options are:

- Modifiable, Extractable, Sensitive

- Encrypt, Decrypt, Wrap, Unwrap
- Sign, Verify, Derive

Note

- BYOK Compatible attributes are Modifiable, Extractable, and Sensitive. They apply to symmetric and asymmetric keys.
- AWS HYOK Compatible attributes are Sensitive, Encrypt, Decrypt, Wrap, and Unwrap. Ensure that the Modifiable and Extractable check boxes are clear. AWS HYOK Compatible applies to symmetric keys.
- OCI HYOK Compatible attributes are Sensitive, Encrypt, and Decrypt. Ensure that the Extractable check box is clear. OCI HYOK Compatible applies to symmetric keys.

8. Click **Next**.

Review And Add Key

This screen shows the key details that you have provided. Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the LUNA HSM KEY section and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the Key ID link is displayed, the key is created successfully.

3. Click **OK**. The Add Luna Key wizard is closed.

The newly created key is displayed in the list of Luna keys.

Note

Creation of a Luna HSM key fails if:

- The selected Luna HSM partition is full. If the key creation fails, ensure enough free disk space is available on the partition and retry the key creation.

- The Luna HSM user credentials are incorrect or the password has expired. Check the credentials or reset the user password, as appropriate, test the connection, and retry the key creation.

Viewing Luna HSM Keys

The Luna Keys page displays the available Luna HSM keys. Search for Luna HSM keys by Key ID, Label, or Partition Label.

To view a Luna HSM key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Luna**. The Luna Keys page displays following details:

Field	Description
Key ID	Unique ID of the Luna HSM key. Click to view details and edit keys. Refer to Viewing or Editing Luna HSM Keys for details.
Label	Label of the Luna HSM key.
Status	Status of the Luna HSM key. The status can be: • Available • In Progress • Deleted • Failed
Class	Class of the Luna HSM key. The class can be: • Private Key • Public Key • Symmetric Key
Algorithm	Type of the Luna HSM key - RSA, AES, or EC.
Detection Date	Time when the key is added to CCKM.
Partition Label	Label of the Luna HSM partition where the key is created.
Synced	Whether the key is synchronized (✓). For keys that are not synced, ✗ is displayed.

To hide/display columns, click the Customize View (⚙️) icon, select or clear the desired check boxes, and click **OK**.

Viewing or Editing Luna HSM Keys

The Luna Keys page displays the list of available keys with their details. After a key is created, you can change its attributes. However, the key mechanism and size cannot be modified.

To edit a Luna HSM key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Luna**. The list of available Luna HSM keys is displayed.
3. Click the **Key ID** link of the desired key. The edit view of the Luna Keys page is displayed. The mini detail view displays the key details, including a link to its private/public key.

Alternatively, click the overflow icon () corresponding to the desired key and click **View/Edit**.

4. Under the GENERAL INFO section, modify the desired **Key Attributes**.
5. Click **Update**.

The message **Updating Key is in progress** is displayed. It may take some time to reflect the changes.

Refreshing Luna HSM Keys

Refreshing is the process of downloading keys from configured Luna HSM partitions and updating their details on the CCKM GUI. You can refresh keys of all partitions at once.

To refresh keys of all partitions:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Luna**. The list of available Luna HSM keys is displayed.
3. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refreshing all Luna HSM Partitions is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Luna > Luna Keys** page.

Syncing Luna HSM Keys

Syncing is the process of synchronizing a key in all Luna HSM partitions. After successful sync, all the partitions have the synced Luna HSM key. Syncing is useful for Luna HSM partitions in the HA mode.

To sync individual keys in all partitions:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Luna**. The list of available Luna HSM keys is displayed.
3. Click the overflow icon () corresponding to the desired key and click **Sync**.

A message **Sync started...** is displayed on the screen. If you want to cancel the sync, click **Cancel Sync**.

The synced key is listed on the **Cloud Keys > Luna > Luna Keys** page. The synced key will be visible under all the partitions.

Deleting Luna HSM Keys

To delete a Luna HSM key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Luna**. The list of available Luna HSM keys is displayed.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Delete**. The Delete Key dialog box is displayed.
5. Select **I wish to delete this key**.
6. Click **Delete**.

A success message **Delete is in progress** is displayed. It may take some time to reflect the changes. After a key is deleted, its status becomes **DeLeted** on the Luna Keys page.

Scheduling Operations

CCKM allows you to periodically refresh keys in the background. To refresh keys, you need to create schedule configuration. A schedule configuration defines when to run a refresh job. You can use either basic format or cron format to specify the time when the refresh job will run.

Specify basic format in the following order:

[Repeats, at]

Where,

- **Repeats**: frequency of the scheduler. Possible values are daily, weekly, monthly, and yearly.
- **at**: specific time at which the refresh operation will be performed. Possible value lies between 12:00-11:00 AM/

PM UTC.

Specify cron format in the following order:

" * * * * * "

These five values indicate when the job should be executed. These values are mandatory and must be specified in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

Field	Allowed values	Allowed special characters
Minute	0-59	* / , -
Hour	0-23	* / , -
Day of month	1-31	* / , - ?
Month	1-12 or JAN-DEC	* / , -
Day of week	0-6 or SUN-SAT	* / , - ?

Adding Key Refresh Schedule

To add a key refresh schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. Under Select Schedule Type, select **Key Refresh**.
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **Luna HSM** from the **Cloud Name** drop-down list.
 - (Optional) Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. On the **Schedule Config** screen, enter or select the following details.
 - a. Configure **DURATION** for the scheduler. Specify the following values:
 - **Schedule Starts**

▪ Schedule Ends

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.

- b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will refresh keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
- c. Click **Next**.

8. On the **Add Partitions** screen:

- a. Select the partition name from the **Refresh Keys from Selected Partition(s)** drop-down list.
- b. Click **+** button.

Similarly, you can add more partitions.

- c. Select the **Refresh All Partitions** option if you want to refresh all the existing partitions.
- d. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Viewing/Editing Schedules

To view/edit a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The Schedules page displays following details:

Field	Description
Name	Unique name of the scheduler configuration.
Schedule Type	Type of the schedule. Possible types are: • Key Rotation • Key Refresh • Add KMS Container (GCP and Azure only) For Luna HSM, only Key Refresh is supported.
Cloud Name	Name of the cloud.
Last Modified	Time when the schedule is modified.
Frequency	Frequency of the scheduler configuration.
Start Date	Creation time of the scheduler configuration.
End Date	Expiry time of the scheduler configuration.

Field	Description
Status	Status of the scheduler configuration. Possible values are: • Enabled • Disabled

3. Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.

4. Edit or configure the following fields and click **Update**:

- Description.
- Status of the scheduler configuration.
- Scheduler configuration parameters such as duration, frequency, and expiry time of the key.

The Schedules page also contains a section named **JOB HISTORY** that contains all the related information about a schedule such as **Run Date**, **Job ID**, **Status**, and **Errors**.

Disabling Schedules

To disable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Disable**. The Disable Schedule message is displayed.
4. Click **Disable Schedule**.

A message **Successfully disabled the schedule** is displayed on the screen.

Enabling Schedules

To enable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Enable**. The Enable Schedule message is displayed.
4. Click **Enable Schedule**.

A message **Successfully enabled the schedule** is displayed on the screen.

Manually Running Schedules

To manually run a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Run Now**.

The Run Now schedule is started in the background. A message **Schedule is running now. It will take a few seconds to finish.** is displayed on the screen. After the schedule is run successfully, a message **Successfully finished running the schedule** is displayed.

Deleting Schedules

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Delete Schedule**. The Delete Schedule message is displayed.
4. Click **Delete**.

A message **Successfully deleted schedule** is displayed on the screen.

Oracle Cloud Resources

Oracle Cloud Infrastructure (OCI) resources include vaults and keys. Manage OCI keys stored in vaults linked to your OCI accounts using CCKM.

Prerequisites

Before you can manage OCI resources on CCKM:

1. [Get an OCI Account](#)
2. [Create an OCI User Group](#)
3. [Apply Policies to the User Group](#)
4. [Create a User in the User Group](#)
5. [Create an OCI Vault](#)
6. [Create an API Key for the User](#)
7. [Add OCI Connection on CipherTrust Manager](#)

Click each link to view details.

Note

Currently, only commercial OCI regions are supported.

Get an OCI Account

Sign up with OCI. When signing up, you need to specify a name for your cloud account (referred to as *tenancy* in OCI) and OCI credentials (username and password). These are required to access the OCI resources.

Note

Steps on this page are documented for the username and password based authentication. If Single Sign-On (SSO) configured for your OCI accounts, steps to log on might differ.

To log on to OCI:

1. Open the Oracle Cloud Sign In page.
2. Enter your tenancy name in the **Cloud Account Name** field.
3. Click **Next**.
4. Enter your **User Name** and **Password**.
5. Click **Sign In**.

Access permissions on a compartment are controlled at the user group level. So, first of all, you need to add an OCI user group. Then, you can apply policy to this user group, and add new users to the group.

Create an OCI User Group

To create an OCI user group:

1. Open the navigation menu and click **Identity & Security**. The Identity & Security page is displayed.
2. Under **Identity**, click **Groups**.
3. Click **Create Group**. The Create Group dialog box is displayed.
4. Enter a unique **Name** for your group, for example, **cckm-group**.

Note

The name cannot contain spaces.

5. Enter a **Description** for your group.
6. Click **Create**.

The OCI group, **cckm-group** is created. Now, you can apply appropriate policies to this group to grant it permissions to manage vaults and keys in your compartments.

Apply Policies to the User Group

Create the policy to give your group (**cckm-group**, created above) permissions in your compartment.

1. Open the navigation menu and click **Identity & Security**. The Identity & Security page is displayed.
2. Under **Identity**, click **Policies**.
3. In the left, under **List Scope**, make sure that your compartment is selected. The Policies in **<your - compartment>** (root) *Compartment* page is displayed.
4. Click **Create Policy**. The Create Policy page displayed.
5. Enter a unique **Name** for your policy, for example, **cckm-policy**.

Note

The name cannot contain spaces.

6. Enter a **Description** for your policy.
7. In the **Policy Builder** section, click **Show manual editor**.
8. In the text field under **Policy Builder**, enter the following:

Syntax

```
Allow group <group-name> to manage vaults in compartment <compartment-name>
Allow group <group-name> to manage keys in compartment <compartment-name>
Allow group <group-name> to read audit-events in <location>
Allow group <group-name> to Inspect compartments IN TENANCY
```

These statements grant a user group the permissions to manage vaults and keys in the specified compartment.

Example

```
Allow group cckm-group to manage vaults in compartment cckm-compartment
Allow group cckm-group to manage keys in compartment cckm-compartment
Allow group cckm-group to read audit-events in us-ashburn-1
Allow group cckm-group to Inspect compartments IN TENANCY
```

These statements grant members of your user group (**cckm-group**) the access to manage vaults and keys in your compartment.

9. Click **Create**.

Create a User in the User Group

To create an OCI user:

1. Open the navigation menu and click **Identity & Security**. The Identity & Security page is displayed.
2. Under **Identity**, click **Users**. The Users page is displayed.
3. Click **Create User**. The Create User dialog box is displayed.
4. Enter a unique **Name** for the user, for example, **cckm-user**.

Note

The name cannot contain spaces.

5. Enter a **Description** for the user.
6. (Optional) Specify and confirm email of the user.
7. Click **Create**.

The OCI user, **cckm-user** is created and displayed on the Users page. Now, you can add this user to the user group you created earlier. The user will inherit the permissions granted to its user group.

To add the user to a group:

1. Click the **Name** link of your user. The user details are displayed.
2. Navigate to the **Groups** section.

3. Click **Add User to Group**. The Add User to Group dialog box is displayed.
4. From the Groups drop-down list, select the desired group (for example, **cckm-group**). The drop-down list shows the available user groups.
5. Click **Add**.

The OCI user, **cckm-user** is added to the selected user group. The user now inherits the permissions granted to the linked user group.

Create an OCI Vault

OCI keys reside inside vaults. Only the users of the group with permissions to manage vaults can create vaults.

To create an OCI vault:

1. Open the navigation menu.
2. Click **Identity & Security > Vault**.
3. In the left, under **List Scope**, in the **Compartment** list, click the name of the compartment where you want to create the vault.
4. Click **Create Vault**. The Create Vault dialog box is displayed.
5. Enter a display **Name** for the vault, for example, **cckm-vault**. Avoid entering confidential information.

Optionally, make the vault a virtual private vault by selecting the **Make it a virtual private vault** check box. For more information about vault types, refer to [Key and Secret Management Concepts](#).

Note

You cannot change the vault type after the vault is created.

6. If you have permissions to create a resource, then you also have permissions to apply free-form tags to that resource. To apply a defined tag, you must have permissions to use the tag namespace. For more information about tagging, refer to [Resource Tags](#). If you are not sure whether to apply tags, skip this option (you can apply tags later) or ask your administrator.
7. When you are finished, click **Create Vault**.

The newly created vault is displayed in the list of vaults.

Create an API Key for the User

Now, you need to create an API key for your user so that the user can add OCI connection and manage vaults and keys. When you create the key, you will get authentication information needed to add an OCI connection to

the CipherTrust Manager.

To add an API key for the user:

1. Open the navigation menu and click **Identity & Security**. The Identity & Security page is displayed.
2. Under **Identity**, click **Users**. The Users page is displayed.
3. Click the **Name** link of your user. The user details are displayed.

Alternatively, click the **Profile** icon in the top right corner, and click the user name link.

4. In the left pane, under **Resources**, click **API Keys**. The API Keys section is displayed under the user details.
5. In the **API Keys** section, click **Add API Key**. The Add API Key screen is displayed.

An API key is an RSA key pair in PEM format used for signing API requests. You can generate the key pair here and download the private key. If you already have a key pair, you can choose to upload or paste your public key file instead. In this section, you will generate a new API key pair.

6. Select **Generate API Key Pair**.
7. Click **Download Private Key**. A private key file is downloaded in the PEM format.
8. Click the **Download Public Key** link to download the associated public key.
9. Click **Add**. The Configuration File Preview dialog box is displayed.

This configuration file snippet includes the authentication information such as OCID of the user and tenancy, fingerprint, and region. You will need this information when adding an OCI connection to the CipherTrust Manager.

10. Click **Close** to close the screen.

The newly added API key is displayed under the API Keys section. To view the key details later, visit the API Keys section, click the overflow icon corresponding to the key, and click **View Configuration File**.

Add OCI Connection on CipherTrust Manager

Before you can add an Oracle vault to the CCKM, a connection to your OCI account must exist on the CipherTrust Manager. After you have created an API key for the user, add a connection to the OCI account.

A CipherTrust Manager administrator manages connections to external resources on the **Access Management > Connections Management** page of the CipherTrust Manager GUI. Refer to [Connection Manager](#) for details.

What Next?

After completing the prerequisites, you can view linked Oracle vaults and manage keys on the CipherTrust Manager.

- [Managing Oracle Vaults](#)

-
- [Managing Oracle Keys](#)
 - [Managing Oracle Tenancies](#)
 - [Scheduling Operations](#)
 - [Managing Oracle Reports](#)
 - [Managing Oracle Compartments](#)
 - [Downloading Oracle Metadata](#)

Managing Oracle Vaults

This section describes how to manage Oracle vaults on CCKM. Before proceeding, a connection to your Oracle account must exist on the CipherTrust Manager. Refer to [Oracle Cloud Infrastructure \(OCI\)](#) for details.

After the connection is configured, you can add vaults to the CipherTrust Manager. Oracle vaults can be added, viewed, modified, or deleted on the Vaults tab of the Oracle Vaults page.

Adding Existing Oracle Vaults

You can add existing vaults linked to an Oracle connection to the CipherTrust Manager. An existing vault can only be added just once.

To add an existing Oracle vault:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**. The Vaults tab of the Oracle Vaults page is displayed.
3. Click **Add Existing Vault**. The Select Connection Information and Vaults screen of the Add Existing Vault wizard is displayed.

Select Connection Information and Vaults

Filter vaults based on Oracle compartments and then by regions.

1. **Select a Connection** from the drop-down list. The list of existing compartments linked with the selected connection is displayed.
2. **Select a Compartment** from the drop-down list. The list of regions where the selected compartment is available is displayed.
3. **Select Region** from the drop-down list. The list of key vaults in the selected region is displayed.
4. Under **Key Vault Name**, select the desired vaults.
5. Click **Next**. The Add Bucket Name, Bucket Namespace screen is displayed.

Add Bucket Name, Bucket Namespace

The bucket name and namespace are required for creating key backups of HSM-protected keys for Virtual Private Vaults. Without these parameters, the Key Backup functionality while syncing vaults will cease.

1. Specify the **Bucket Name**.
2. Specify the **Bucket Namespace**.
3. Click **Add**.

The selected vault is displayed on the Vaults tab of the Oracle Vaults page. Now, you can manage the vault from CCKM on the CipherTrust Manager.

The vault is available to [upload Oracle keys](#) and [view Oracle reports](#).

Viewing Oracle Vaults

The Vaults tab of the Oracle Vaults page shows the list of vaults added to the CipherTrust Manager. Search for the vaults by Vault Name, Tenant, or Compartment.

To view the list of vaults added to the CipherTrust Manager:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**. The Vaults tab of the Oracle Vaults page shows the list of vaults added to the CipherTrust Manager.

The page displays the following details:

Column	Description
Vault Name	Name of the vault.
Tenancy	Name of the Oracle tenancy.
Compartment	Name of the Oracle compartment.
Region	Region of the vault.
Virtual Private	Whether the vault is a Virtual Private vault.
State	State of the vault.
Connection	Name of the Oracle connection added to the CipherTrust Manager.
Last Refreshed	Date and time when the vault was refreshed the last.

To view/hide columns, click the Customize View () icon, select/clear the desired option, and click **OK** to display the column.

Refreshing Oracle Vaults

Refreshing is the process to download keys created in Oracle vaults to the CCKM. You can refresh keys from individual or all Oracle vaults.

- [Refreshing Specific Oracle Vaults](#)
- [Refreshing All Oracle Vaults](#)

The backup of Oracle keys is created only when the vaults are refreshed. The backup can only be created for keys that:

- Are stored in Virtual Private Vaults (VPVs)
- Are stored in vaults that have associated bucket credentials
- Have the HSM protection mode

Refreshing Specific Oracle Vaults

To refresh a vault:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. On the Vaults tab, click the overflow icon () corresponding to the desired vault and click **Refresh Now**.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Oracle > Oracle Keys** page. Refer to [Viewing Oracle Keys](#) for details.

Refreshing All Oracle Vaults

To refresh all Oracle vaults:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. On the Vaults tab, click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all vaults is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

-
4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Oracle > Oracle Keys** page. Refer to [Viewing Oracle Keys](#) for details.

Viewing Details of a Vault

To view the details of a vault on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. On the Vaults tab, click the **Name** link of the desired vault.

Alternatively, click the overflow icon () corresponding to the desired vault, and click **View/Edit Details**.

The edit view of the Oracle Vaults page shows additional details of the selected vault under the GENERAL INFO and ACCESS CONTROL sections. Expand each section to view more details.

The GENERAL INFO section provides details of linked Oracle connection, Bucket Name, and Bucket Namespace. If needed, you can modify these settings, as appropriate.

Changing the Oracle Connection

To change the Oracle connection of a vault:

1. Expand **GENERAL INFO**.
2. From the **Connection** drop-down list, select the desired Oracle connection.
3. Click **Update**.

The connection of the Oracle vault is changed.

Modifying the Bucket Details

To modify the bucket details of a vault:

1. Expand **GENERAL INFO**.
2. Update the **Bucket Name**.
3. Update the **Bucket Namespace**.
4. Click **Update**.

The bucket name and bucket namespace of the Oracle vault is modified.

Managing User Permissions on Oracle Vaults

To work with the Oracle cloud, users/groups must have the minimum set of permissions that allow them to use the Oracle resources such as keys and vaults. Initially, the CCKM user only has permission to view the keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform operations on Oracle vaults.

Users with the following characteristics can perform operations for Oracle keys and vaults:

- Users in the **CCKM Admins** group
- Users in the **Admin** group
- Users who are administrators for a **domain**
- Users who are in the **CCKM Users** group and which have had a CCKM Admin assign permissions through the UI or the `/v1/cckm/oci/vaults/{id}/update-acls` endpoint in the REST API.

Adding Permissions for a User/Group

To add permissions for a user/group:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. On the Vaults tab, click the **Name** link of the desired vault.

Alternatively, click the overflow icon () corresponding to the desired vault, and click **View/Edit Details**.

4. Expand the **ACCESS CONTROL** section.
5. Click **Assign User/Group**. The **Assign User/Group** dialog box is displayed.
6. Select the desired user or group from the **User/Group** drop-down list.
7. Click **Save**.

The newly added user/group is displayed under **Name** in the **ACCESS CONTROL** section. You can now grant additional permissions to the user/group, as appropriate. Refer to [Granting Permission to Perform an Operation](#) for details.

Allowed Operations

CCKM allows the following operations on Oracle vaults:

-
- View Keys, Add Native Key, Add BYOK Key, Edit Key
 - Rotate Key, Delete Key, Cancel Delete, Remove Key Backup, Remove Key
 - Rotate to Native Key, Rotate to BYOK Key, Refresh Key

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. In the ACCESS CONTROL section, select the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

To revoke permissions from a user/group, refer to [Removing a Permission](#) for details.

Removing a Permission

To remove a permission assigned to a user or group:

1. In the ACCESS CONTROL section, clear the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. In the ACCESS CONTROL section, under **Unassign**, click the **X** button corresponding to the desired user/group.
2. On the **Remove User / Remove Group** screen, click **Remove**.

Note

Removing this user/group will remove all permissions currently assigned to the user/group.

3. Click **Remove** to confirm the action. To cancel the action, click **Keep It**.

A success message is displayed on the screen.

Removing Oracle Vaults

Oracle vaults can be removed on the Oracle Vaults page. Search for existing vaults using Vault Name, Tenant, or Compartment.

To remove a vault from CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. On the Vaults tab, click the overflow icon () corresponding to the vault you want to remove.
4. Click **Remove Vault**.
5. Select **I wish to delete the vault**.
6. Click **Delete**.

The Oracle vault is deleted successfully. The vault is removed from the list of Oracle vaults.

Managing Oracle Keys

This section describes how to manage Oracle keys on CCKM. Before proceeding, you must have an Oracle vault added to the CCKM. Refer to [Managing Oracle Vaults](#) for details.

- [Key Creation Methods and Sources](#)
- [Adding Key Material Using External \(BYOK\) Source](#)
- [Adding Key Material Using Oracle External \(HYOK\) Source](#)
- [Creating Native Key Material](#)
- [Viewing Oracle Keys](#)
- [Viewing Key Versions](#)
- [Viewing or Editing Details of Oracle Keys](#)
- [Refreshing Oracle Keys](#)
- [Disabling an Oracle Key](#)
- [Enabling an Oracle Key](#)
- [Moving Keys to Another Compartment](#)
- [Adding a Key Version](#)
- [Scheduling Deletion of a Key](#)

- [Canceling Deletion of a Key](#)
- [Removing a Key Backup](#)
- [Restoring a Deleted Key](#)

Key Creation Methods and Sources

Key material for new OCI keys can be added by:

- Creating and uploading a new source key, or by creating a new Oracle native key.
- Copying key material from an existing source key to create a new key.

For adding OCI keys, CCKM supports the following key material sources:

- **External (BYOK):** External (Bring Your Own Key). Add key material by creating or uploading new source key from an external source. Refer to [Adding Key Material Using External \(BYOK\) Source](#) for details. You can select CipherTrust Manager (External), CipherTrust Manager (Local), Luna HSM, or Vormetric (DSM) as an external key source.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

- **Oracle External (HYOK):** External KMS (Hold Your Own Key). Add an HYOK key tied to key material stored in an external key vault. This option is available for external key vaults only. [Adding Key Material Using Oracle External \(HYOK\) Source](#) for details. You can select CipherTrust Manager (Local) as an external key source.
- **Oracle Native:** Create Oracle key material directly with native Oracle application. Refer to [Creating Native Key Material](#) for details.

Adding Key Material Using External (BYOK) Source

To add key material using external BYOK as a key source:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click **Add Key**. The Key Material Origin screen of the Add Oracle Key wizard is displayed.

Key Material Origin

1. Select **External (BYOK)** as the **Origin Type**. This is the default option.

2. Select the desired **Vault** from the drop-down list. The list shows available external and default vaults.
3. Select an **Oracle Compartment** from the drop-down list. The list shows the available compartments based on the selected vault. This field is mandatory for default vaults, and is read-only for external vaults.
4. Select the **Source**. The options are:
 - **CipherTrust (External)**: Add key by creating or uploading an external CipherTrust key as the source key. Refer to [Adding Key Using External CipherTrust as External \(BYOK\) Source](#) for details.
 - **CipherTrust (Local)**: Add key by creating or uploading a CipherTrust key as the source key. Refer to [Adding Key Using Local CipherTrust as External \(BYOK\) Source](#) for details.
 - **Luna HSM**: Add key by creating or uploading a Luna HSM key as the source key. Refer to [Adding Key Using Luna HSM as External \(BYOK\) Source](#) for details.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

- **Vormetric DSM**: Add key by creating or uploading a Vormetric DSM key as the source key. Refer to [Adding Key Using Vormetric DSM as External \(BYOK\) Source](#) for details.

Adding Key Using External CipherTrust as External (BYOK) Source

To add a key by creating or uploading an external CipherTrust key as the source key:

Key Material Origin

1. Select **CipherTrust (External)** as the **Source**.
2. Click **Next**. The Configure CipherTrust (External) Key screen is displayed.

Configure CipherTrust (External) Key

1. Select the **Source Key Material**. This specifies how to create the key. The options are:
 - **Create New Key**: Click to create a fresh key.
 - a. **Select Domain** for the key. The drop-down list shows the domains of the external CipherTrust Manager linked with the configured connection.
 - b. Enter a **Key Name**.
 - c. Select the **Algorithm** from the available options. The supported algorithms are AES and RSA.

d. Select the **Key Size** from the available options.

- **Copy Existing Key:** Click to create a new key by copying an existing key. **Select Domain**, then select an existing **Key** from the drop-down list.

2. Click **Next**. The Configure Oracle Key screen is displayed.

Configure Oracle Key

1. Enter a unique, user-friendly alias as the **Oracle Key Name**. This will be the key name on Oracle cloud. This name helps uniquely identify an Oracle key.
2. Select the **Protection Mode**. The options are Software and HSM.
3. (Optional) Specify the tags.

To add a new tag:

a. Select a **Tag Namespace**. The options are:

- **Free Form:** Allows adding free form tags.
- **Oracle Tags:** Allows adding tags based on created on and created by.

b. Specify a **Tag Key**.

c. Specify a **Tag Value**.

d. Click **+**.

Similarly, add as many tags as required.

4. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, and BYOK KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** section and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY section becomes Complete and the Key ID link is displayed, the

key is created successfully.

3. Click **Close**. The Add Oracle Key wizard is closed.

The newly created key is displayed in the list of Oracle keys.

Adding Key Using Local CipherTrust as External (BYOK) Source

To add a key by creating or uploading a CipherTrust key as the source key:

Key Material Origin

1. Select **CipherTrust (Local)** as the **Source**.
2. Click **Next**. The Configure CipherTrust (Local) Key screen is displayed.

Configure CipherTrust (Local) Key

1. Select the **Source Key Material**. This specifies how to create the key. The options are:
 - **Create New Key**: Click to create a fresh key.
 - a. Enter a **Key Name**.
 - b. Select the **Key Type** from the available options. The supported key types are AES and RSA.
 - c. Select the **Key Size** from the available options.
 - **Copy Existing Key**: Click to create a new key by copying an existing key. Select **Key Type**, **Key Size***, and an existing key from the CipherTrust Key Name** drop-down list.
2. Click **Next**. The Configure Oracle Key screen is displayed.

Configure Oracle Key

1. Enter a unique, user-friendly alias as the **Oracle Key Name**. This will be the key name on Oracle cloud. This name helps uniquely identify an Oracle key. By default, the Key Name you specified on the previous screen is populated.
2. Select the **Protection Mode**. The options are Software and HSM.
3. (Optional) Specify the tags.

To add a new tag:

- a. Select a **Tag Namespace**. The options are:
 - **Free Form**: Allows adding free form tags.
 - **Oracle Tags**: Allows adding tags based on created on and created by.
- b. Specify a **Tag Key**.

c. Specify a **Tag Value**.

d. Click **+**.

Similarly, add as many tags as required.

4. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, and BYOK KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **NATIVE KEY** section and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is complete.

When the status next to the BYOK KEY section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Oracle Key wizard is closed.

The newly created key is displayed in the list of Oracle keys.

Adding Key Using Vormetric DSM as External (BYOK) Source

To add a key by creating or uploading a Vormetric DSM key as the source key:

Key Material Origin

1. Select **Vormetric DSM** as the **Source**.
2. Click **Next**. The Configure DSM Key screen is displayed.

Configure DSM Key

1. Select the **Source Key Material**. This specifies how to create the key. The options are:
 - **Create New Key**
 - a. Click to create a fresh key.
 - b. Specify the **DSM Key Name** for the new key.
 - c. (Optional) Add a **Description** for the new key.
-

-
- d. Select the desired **DSM Domain** for the key. This drop-down list shows the DSM domains based on the DSM connection added to the CipherTrust Manager.
 - e. Select the **Key Type** from the available options. The supported key types are AES and RSA.
 - f. Select the **Key Size** from the available options.
- **Copy Existing Key:** Click to create a new key by copying an existing key.
 - a. Select the **Key Type** from the available options. The supported key types are AES and RSA.
 - b. Select the **Key Size** from the available options.
 - c. Select an existing key from the **DSM Key Name** drop-down list. This field shows the available keys based on the DSM connection added to the CipherTrust Manager.
2. Click **Next**. The Configure Oracle Key screen is displayed.

Configure Oracle Key

1. Enter a unique, user-friendly alias as the **Oracle Key Name**. This will be the key name on Oracle cloud. This name helps uniquely identify an Oracle key. By default, the Key Name you specified on the previous screen is populated.
2. Select the **Protection Mode**. The options are Software and HSM.
3. (Optional) Specify the tags.

To add a new tag:

- a. Select a **Tag Namespace**. The options are:
 - **Free Form:** Allows adding free form tags.
 - **Oracle Tags:** Allows adding tags based on created on and created by.
- b. Specify a **Tag Key**.
- c. Specify a **Tag Value**.
- d. Click **+**.

Similarly, add as many tags as required.

4. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, and BYOK KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** and **BYOK KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is complete.

When the status next to the SOURCE KEY and BYOK KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Oracle Key wizard is closed.

The newly created key is displayed in the list of Oracle keys.

Adding Key Using Luna HSM as External (BYOK) Source

To add a key by creating or uploading a Luna HSM key as the source key:

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Key Material Origin

1. Select **Luna HSM** as the **Source**.
2. Click **Next**. The Configure HSM Key screen is displayed.

Configure HSM Key

1. Select the **Source Key Material**. This specifies how to create the key. The options are:
 - **Create New Key**
 - a. Click to create a fresh key.
 - b. Select the desired **Partition ID** for the key. This drop-down list shows the Luna HSM partitions based on the Luna HSM connection added to the CipherTrust Manager. The new Luna HSM key will be created in this partition.
 - c. Specify the **HSM Key Name** for the new key.
 - d. Select the **Key Type** from the available options. The supported key types are AES and RSA.
 - e. Select the **Key Size** from the available options.

-
- f. Select a key **Mechanism** from the available options.
 - g. Select the **Key Attributes**. The options are:
 - Modifiable, Extractable, Sensitive (all *three* are selected for a **BYOK Compatible** key)
 - Encrypt, Decrypt, Wrap, Unwrap
 - Sign, Verify, Derive
 - **Copy Existing Key**: Click to create a new key by copying an existing key.
 - a. Select the **Key Type** from the available options. The supported key types are AES and RSA.
 - b. Select the **Key Size** from the available options.
 - c. Select an existing key from the **HSM Key Name** drop-down list. This field shows the available keys based on the Luna HSM connection added to the CipherTrust Manager.

2. Click **Next**. The Configure Oracle Key screen is displayed.

Configure Oracle Key

1. Enter a unique, user-friendly alias as the **Oracle Key Name**. This will be the key name on Oracle cloud. This name helps uniquely identify an Oracle key. By default, the Key Name you specified on the previous screen is populated.
2. Select the **Protection Mode**. The options are Software and HSM.
3. (Optional) Specify the tags.

To add a new tag:

- a. Select a **Tag Namespace**. The options are:
 - **Free Form**: Allows adding free form tags.
 - **Oracle Tags**: Allows adding tags based on created on and created by.
- b. Specify a **Tag Key**.
- c. Specify a **Tag Value**.
- d. Click **+**.

Similarly, add as many tags as required.

4. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN,

SOURCE KEY, and BYOK KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** and **BYOK KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is complete.

When the status next to the SOURCE KEY and BYOK KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Oracle Key wizard is closed.

The newly created key is displayed in the list of Oracle keys.

Adding Key Material Using Oracle External (HYOK) Source

To add key material using external Oracle HYOK as a key source:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click **Add Key**. The Key Material Origin screen of the Add Oracle Key wizard is displayed.

Key Material Origin

1. Select **Oracle External (HYOK)** as the **Origin Type**.
2. Select the external **Vault** from the drop-down list. The list shows available external and default vaults.

The **Oracle Compartment** field is unavailable (read-only) for external vaults.

3. Select the **Source**. The options are:
 - **CipherTrust (Local)**: Add key by creating or uploading a CipherTrust key as the source key. Refer to [Adding Key Material Using Local CipherTrust as External \(HYOK\) Source](#) for details.
 - **Luna HSM**: Add key by creating or uploading a Luna HSM key as the source key. Refer to [Adding Key Material Using Luna HSM as External \(HYOK\) Source](#) for details.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Adding Key Material Using Local CipherTrust as External (HYOK) Source

To add a key by creating or uploading a CipherTrust key as the source key:

Key Material Origin

1. Select **CipherTrust (Local)** as the **Source**.
2. Click **Next**. The Configure CipherTrust (Local) Key screen is displayed.

Source Key

Caution

Modify the Oracle External Vaults (HYOK) only through the CipherTrust Cloud Key Management (CCKM) GUI, API, and CLI. Do not modify the source key using the **Keys** menu, the `ksctl keys` command, or the `/v1/vault/keys2` endpoint. Such modifications might make the Oracle (HYOK) key unavailable to the Oracle cloud services.

1. Select an option from **Source Key Material** to specify how to create the key. The options are:
 - **Create New Key**: Click to create a fresh key. Specify the Oracle Key Name for the new key.
 - **Select Existing Key**: Click to create a new key by copying an existing key. Select a source key from the available options.
2. Click **Next**. The Configure Oracle Key screen is displayed.

Configure Oracle Key

1. Enter a unique and user-friendly alias in the **Oracle Key Name** field. This will be the key name on Oracle cloud to help identify the Oracle key. By default, the Oracle Key Name specified previously, if creating a new key, is populated.
2. Specify a Rego policy for the key.
3. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the specified key details. These details are divided into **MATERIAL ORIGIN**, **SOURCE KEY**,

and **POLICY** sections.

Before adding the key, review all the details. After the key is added, certain features cannot be edited.

1. Review the key details displayed on the screen.

If you want to make any changes, click **Edit** adjacent to the **SOURCE KEY** and **HYOK KEY** sections. Alternatively, click **Back** and make changes.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is complete.

The key is created successfully when the status next to the **SOURCE KEY** and **HYOK KEY** sections changes to **Complete** and the Key ID links are displayed.

3. Click **Close**. The Add Oracle Key wizard is closed.

The newly created key is displayed in the Oracle keys list.

Adding Key Material Using Luna HSM as External (HYOK) Source

To add a key by creating or uploading a Luna HSM key as the source key:

Key Material Origin

1. Select **Luna HSM** as the **Source**.
2. Click **Next**. The Configure HSM Key screen is displayed.

Configure HSM Key

Caution

Modify the Oracle External Vaults (HYOK) only through the CipherTrust Cloud Key Management (CCKM) GUI, API, or CLI. Do not modify the source key using the **Keys** menu, the `ksctl keys` command, or the `/v1/vault/keys2` endpoint. Such modifications might make the Oracle (HYOK) key unavailable to the Oracle cloud services.

1. Select the **Source Key Material**. This specifies how to create the key. The options are:
 - **Create New Key**
 - a. Select the desired **Partition ID** for the key. This drop-down list shows the Luna HSM partitions based on the Luna HSM connection added to the CipherTrust Manager. The new Luna HSM key is created in this partition.
 - b. Specify the **HSM Key Name** for the new key.

Note

The values of the **Key Type**, **Key Size**, and **Mechanism** fields are predefined and cannot be modified. The defined values of Key Type and Key Size are set to AES and 256, respectively.

c. Select the **Key Usage**. The options are:

- Modifiable, Sensitive: By default, these are selected for a **OCI HYOK Compatible** key.
- Extractable, Encrypt, Decrypt, Wrap, Unwrap

Note

By default, the Extractable option is inactive and cannot be modified.

- Sign, Verify
- **Select Existing Key:** Click to create a new key by selecting an existing key.
 - a. Select the desired **Partition ID** for the key. This drop-down list shows the Luna HSM partitions based on the Luna HSM connection added to the CipherTrust Manager.

Note

The values of the **Key Type**, **Key Size**, and **Mechanism** fields are predefined and cannot be modified. The defined values of Key Type and Key Size are set to AES and 256, respectively.

- b. Select an existing key from the **Select HSM Key** list. Alternatively, you can search for a key name using the **Key Name** field. This field shows the available keys based on the Luna HSM connection added to the CipherTrust Manager.

2. Click **Next**. The Configure Oracle Key screen is displayed.

Configure Oracle Key

1. Enter a unique and user-friendly alias in the **Oracle Key Name** field. This will be the key name on Oracle cloud to help identify the Oracle key. By default, the Oracle Key Name specified previously, if creating a new key, is populated.
2. Specify the **Policy**.

-
3. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the specified key details. These details are divided into **MATERIAL ORIGIN**, **SOURCE KEY**, and **HYOK KEY** sections.

Before adding the key, review all details. After the key is added, certain features cannot be edited.

1. Review the key details displayed on the screen.

If you want to make any changes, click **Edit** adjacent to the **SOURCE KEY** and **HYOK KEY** sections. Alternatively, click **Back** and make changes.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Make sure to leave the window open until the process is complete.

The key is created successfully when the status next to the **SOURCE KEY** and **HYOK KEY** sections changes to **Complete** and the Key ID links are displayed.

3. Click **Close**. The Add Oracle Key wizard is closed.

The newly created key is displayed in the Oracle keys list.

Creating Native Key Material

To create Oracle key using the native Oracle application:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click **Add Key**. The Key Material Origin screen of the Add Oracle Key wizard is displayed.

Key Material Origin

1. Select **Oracle Native** as the **Origin Type**.
2. Select the desired **Vault** from the drop-down list. The list shows available external and default vaults.
3. Select an **Oracle Compartment** from the drop-down list. The list shows the available compartments based on the selected vault. This field is mandatory for default vaults, and is read-only for external vaults.
4. Click **Next**. The Configure Oracle Key screen is displayed.

Configure Oracle Key

1. Enter a unique and user-friendly alias in the **Oracle Key Name** field. This will be the key name on Oracle cloud to help identify the Oracle key.

-
2. Select the **Key Algorithm**. The options are **AES**, **RSA**, and **ECDSA**.
 3. Select the **Key Size** based on the key type:
 - For an **AES** key, the options are 16, 24, and 32 bytes.
 - For an **RSA** key, the options are 256, 384, and 512 bytes.
 - For an **ECDSA** key, the options are 32, 48, and 66 bytes.
 4. Select the **Protection Mode**. The options are Software and HSM.
 5. (Optional) Specify the tags.

To add a new tag:

- a. Select a **Tag Namespace**. The options are:
 - **Free Form**: Allows adding free form tags.
 - **Oracle Tags**: Allows adding tags based on created on and created by.
- b. Specify a **Tag Key**.
- c. Specify a **Tag Value**.
- d. Click **+**.

Similarly, add as many tags as required.

6. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN and NATIVE KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **NATIVE KEY** section and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is complete.

When the status next to the NATIVE KEY section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Oracle Key wizard is closed.

The newly created key is displayed in the list of Oracle keys.

Viewing Oracle Keys

The Oracle Keys page shows the list of Oracle keys available on the CipherTrust Manager.

To view the Oracle keys:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**. The list of available Oracle keys is displayed. The Oracle Keys page displays the following details:

Field	Description
Name	Unique, user-friendly name of the Oracle key. Click the link to view additional details of the key or edit the key. Refer to Viewing or Editing Details of Oracle Keys . This name is useful in searching for specific keys.
Key ID	ID of the Oracle key.
Tenancy	Name of the Oracle tenancy.
Compartment	Name of the OCI compartment where the key resides.
Vault	Name of the OCI vault where the key resides.
Protection Mode	Protection mode for the key - HSM or Software.
Algorithm	Algorithm of the Oracle key. AES, RSA, and ECDSA algorithms with different keys sizes are supported.
State	State of the Oracle key.
Region	Region of the key.
Version	Version of the key.
Creation Date	Date and time when the Oracle key is created.
Deletion Date	Date and time when the Oracle key is deleted.
Origin	Source of the key material used for the version. The origin can be: • CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud. • External (Unknown): Source of the key material is

Field	Description
	unknown. It is different than CCKM and the native cloud. Refer to Key Creation Methods and Sources for details.

The **Region**, **Version**, **Creation Date**, and **Origin** columns are hidden by default. To show/hide a column, click the custom view icon (⚙️), select/clear the desired column, and click **OK**.

Viewing Key Versions

To view the versions of a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**. The list of available Oracle keys is displayed.
3. Click the expand icon (▶) corresponding to the desired key. The mini detail view shows the list of key versions with their details.

Field	Description
Version ID	ID of the key version.
State	State of the key version. The state can be enabled or disabled.
Origin	Source of the key material. The origin of the key can be: • CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud. • External (Unknown): Source of the key material is unknown. It is different than CCKM and the native cloud.
Creation Date	Date and time when the Oracle key is created.

Alternatively, you can view the versions of a key in the details view of the key. Refer to [Viewing Key Version Details](#) for details.

Viewing or Editing Details of Oracle Keys

After a key is created, you can add tags to it, schedule their rotation, and view its versions.

In the edit view of a key, you can view all the key details such as its ID, compartment, vault, state, algorithm, and region etc.

To view or edit an Oracle key:

-
1. Open the **Cloud Key Manager** application.
 2. In the left pane, click **Cloud Keys > Oracle**. The list of available Oracle keys is displayed.
 3. Click the overflow icon () corresponding to the desired key and click **View/Edit Details**. Alternatively, you can click the key name link. The edit view of the key is displayed. The edit view is divided into:
 - **GENERAL INFO**: View the key ID and add tags. Refer to [Adding/Removing Tags](#) for details.
 - **KEY SCHEDULE**: Add, update, and disable a key rotation schedule. Refer to [Adding or Changing Key Rotation Schedule](#) and [Disabling Key Rotation Schedule](#).
 - **KEY VERSIONS**: View details of key versions. Refer to [Viewing Key Version Details](#).

Adding/Removing Tags

To add tags to a key:

1. Expand the **GENERAL INFO** section, if needed.
2. Select a **Tag Namespace**. The options are:
 - **Free Form**: Allows adding free form tags.
 - **Oracle Tags**: Allows adding tags based on created on and created by.
3. Specify a **Tag Key**.
4. Specify a **Tag Value**.
5. Click **+**. Similarly, add as many tags as required.
6. Click **Update**.

The tag is added to the key.

To remove a tag:

1. Expand the **GENERAL INFO** section, if needed.
2. Click the close icon in the added tags.
3. Click **Update**.

Adding or Changing Key Rotation Schedule

To add or update a key rotation schedule:

1. Expand the **KEY SCHEDULE** section.
2. From the **Select Rotation Schedule** drop-down list, select the desired schedule.

3. Select the **Key Origin**. The options are:

- CipherTrust (Local)
- CipherTrust (External), also **Select Domain** of the CipherTrust Manager.
- Native
- Luna, also select the Luna HSM **Partition**.
- DSM, also select the DSM **Domain**.

For keys based on ECDSA algorithms, only Native is available as the key origin.

4. Click **Update**.

The key rotation schedule is added/updated. The selected schedule is now assigned to the key. To view all the keys assigned to a schedule, refer to [Viewing Keys Assigned to Schedules](#).

Disabling Key Rotation Schedule

To disable a key rotation schedule:

1. Expand the KEY SCHEDULE section.
2. Next to the **Select Rotation Schedule** drop-down list, click the close icon (.

Auto key rotation is disabled.

Viewing Key Version Details

To view the details of key versions, expand the KEY VERSIONS section. The key version details are displayed. Refer to [Viewing Key Versions](#) for details.

Refreshing Oracle Keys

Refreshing is the process of downloading keys created in Oracle vaults to CCKM. You can refresh individual keys or all keys from all Oracle vaults at once.

Refreshing All Keys

To refresh all keys:

1. Open the Cloud Key Manager application.
2. In the left pane, click **Cloud Keys > Oracle**. The Oracle Keys tab is displayed. This tab displays the list of Oracle keys.
3. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all keys is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message Refresh started... is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Oracle > Oracle Keys** page.

Refreshing Individual Keys

To refresh individual keys and their versions:

1. Open the Cloud Key Manager application.
2. In the left pane, click **Cloud Keys > Oracle**. The Oracle Keys page is displayed. This page displays the list of Oracle keys.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Refresh**. The Refresh Key dialog box is displayed. The key and all its versions will be refreshed.
5. Click **Refresh** to confirm the action.

The key with its versions is refreshed successfully.

The refreshed key and its versions are listed on the **Cloud Keys > Oracle > Oracle Keys** page.

Disabling an Oracle Key

This section is applicable to Oracle BYOK keys.

If required, you can disable an enabled key. A disabled key cannot operate on data. Disabling a key disables all versions of the key.

To disable a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Disable**. The Disable Key dialog box is displayed.
5. Click **Disable** to confirm the action.

The state of the key changes to Disabling, finally to Disabled.

Enabling an Oracle Key

This section is applicable to Oracle BYOK keys.

If required, you can enable a disabled key. Enabling a key enables all versions of the key.

To enable a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Enable**. The Enable Key dialog box is displayed.
5. Click **Enable** to confirm the action.

The state of the key changes to Enabling, finally to Enabled.

Moving Keys to Another Compartment

This section is applicable to Oracle BYOK keys.

If needed, you can move a key to another Oracle compartment.

To move a key to another compartment:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Move Resource**. The Move Resource dialog box is displayed.
5. **Select Compartment** from the drop-down list. The drop-down list shows the list of Oracle compartments linked with the added Oracle vaults.
6. Click **Save**.

The key is moved to the selected compartment.

Adding a Key Version for the Software or HSM Protection Mode

To add a key version for the Software or HSM Protection Modes in CCKM:

1. Open the **Cloud Key Manager** application.
 2. In the left pane, click **Cloud Keys > Oracle**.
-

3. Click the overflow icon () corresponding to the key with the HSM or Software protection mode.
4. Click **Add Version**. The Add Version dialog box is displayed.
5. Select a method from the **Select Method** field. The options are:
 - **Create/Upload New Key Material**: Refer to [Adding Key Version by Creating/Uploading Key Material](#).
 - **Clone Existing Key Material**: Refer to [Adding Key Version by Cloning Existing Key Material](#).

Refer to [Key Creation Methods and Sources](#) for details on key creation methods and key sources.

Adding Key Version by Creating New Key Material

To create a new key version for the HSM Protection Mode in CCKM :

1. Select **Create/Upload New Key Material** as the method.
2. Select a source from the **Select Source** field. The options are:
 - **CipherTrust (External)**: In this option, select the **Domain** and specify **Key Name** for the new key version.
 - **CipherTrust (Local)**: In this option, specify **Key Name** for the new key version.
 - **Oracle (Native)**: Use this option to create a new native Oracle key.
 - **Vormetric DSM**: In this option, specify **Key Name** for the new key version and select the DSM **Domain**.
 - **Luna HSM**: In this option, select the **Partition ID**, **Mechanism**, and **Key Attributes**, and specify **Key Name** for the new key version.

The key attributes **Modifiable** and **Sensitive** are selected for a **HYOK Compatible** key. By default, the **Extractable** option is inactive and cannot be modified.

3. Click **Add Version**.

A new version is added to the key. The Version Count increases by one on the Oracle Keys page.

Adding Key Version by Cloning Existing Key Material

To clone the existing key version for the HSM Protection Mode in CCKM :

1. Select the **Clone Existing Key Material** method.
2. Select a source from the **Select Source** field. The options are:
 - **CipherTrust (External)**
 - **CipherTrust (Local)**
 - **Vormetric DSM**

- **Luna HSM**

For the above options, you need to select a value from the **Select a source key** drop-down menu.

3. Click **Add Version**.

A new version is added to the key. The Version Count increases by one on the Oracle Keys page.

Adding a Key Version for the External Protection Mode

To add a key version for the External Protection Mode in CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the key with the External protection mode.
4. Click **Add Version**. The Add Version dialog box is displayed.
5. Select a method from the **Select Method** field. The options are:
 - **Create New Key**: Refer to [Adding Key Version by Creating New Key](#).
 - **Select Existing Key**: Refer to [Adding Key Version by Selecting Existing Key](#).

Adding Key Version by Creating New Key

To create a new key version for the External Protection Mode in CCKM :

1. Select the **Create New Key** method.
2. From the **Select Source** field, select the source of the key version. Either of the following options is displayed depending on the source of the parent key. For example, if the parent key source is Luna HSM, then only Luna HSM is listed for you to select.
 - **CipherTrust (Local)**: In this option, specify **Key Name** for the new key version.
 - **Luna HSM**: In this option, select the **Partition ID**, and specify **HSM Key Name** for the new key version. The values of the **Key Type**, **Key Size**, and **Mechanism** fields are predefined and cannot be modified. The defined values of **Key Type** and **Key Size** are set to **AES** and **256**, respectively.

The key attributes **Modifiable** and **Sensitive** are selected for a **HYOK Compatible** key. By default, the **Extractable** option is inactive and cannot be modified.

3. Specify **Key Name**.
4. Click **Add Version**.

A new version is added to the key. The Version Count increases by one on the Oracle Keys page.

Adding Key Version by Selecting Existing Key

To select an existing key version for the External Protection Mode in CCKM :

1. Select the **Select Existing Key** method.
2. From the **Select Source** field, select the source of the key version. Either of the following options is displayed depending on the source of the parent key. For example, if the parent key source is Luna HSM, then only Luna HSM is listed for you to select.
 - **CipherTrust (Local)**
 - **Luna HSM**
3. Click **Next**. The **Select Source Key and Add version** screen is displayed.
4. Select the desired key from the given list. Optionally, if **CipherTrust (Local)** is selected as source, you can select the **Fetch Latest Version Only** option to view the latest key versions.

Note

You can modify the number of key IDs displayed per page by clicking the  icon. This icon is located at the bottom of the key name list.

5. Click **Add Version**.

A new version is added to the key. The Version Count increases by one on the Oracle Keys page.

Scheduling Deletion of a Key

This section is applicable to Oracle BYOK keys.

With CCKM, you can schedule deletion of an Oracle BYOK key. The key is removed from Oracle at the specified time. Oracle enforces a waiting period of 7 to 30 days. After a key is deleted, it cannot be restored and the data encrypted with the key is unrecoverable. Before the waiting period ends, schedule key deletion can be cancelled.

Note

An external key (HYOK) resides on the CipherTrust Manager, it does not require a deletion schedule. The key can be deleted directly from the CipherTrust Manager. Refer to [Deleting External Keys](#).

To schedule the deletion of a key:

1. Open the **Cloud Key Manager** application.

-
2. In the left pane, click **Cloud Keys > Oracle**.
 3. Click the overflow icon () corresponding to the desired key.
 4. Click **Schedule Key Deletion**. The Schedule Key Deletion dialog box shows the name and ID of the key.
 5. Select **I wish to delete this key**.
 6. Specify the **Waiting period (in Days)**. The default waiting period is 30 days.
 7. Click **Schedule Deletion**.

A message stating that the key is scheduled for deletion is displayed. The key state changes to SCHEDULING_DELETION, then to PENDING_DELETION until the waiting period is over. After the waiting period is over, the key state becomes DELETED.

You can cancel the scheduled deletion of a key before the waiting period expires. Refer to [Canceling Deletion of a Key](#) for details.

Canceling Deletion of a Key

This section is applicable to Oracle BYOK keys.

Before the waiting period ends, the scheduled deletion of a key with the state PENDING_DELETION can be cancelled.

To cancel the scheduled deletion of a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the desired key with the state PENDING_DELETION.
4. Click **Cancel Key Deletion**. The Cancel Key Deletion dialog box shows the name and ID of the key.
5. Select **I wish to cancel key deletion**.
6. Click **Cancel Schedule Deletion**.

A message stating that the key deletion is canceled is displayed. The key state changes to CANCELLING_DELETION, then to ENABLED.

Removing a Key

When an Oracle key is deleted from Oracle cloud, its status on the CipherTrust Manager becomes DELETED. You can remove such keys with their versions and backup from the CipherTrust Manager.

Note

An external key (HYOK) resides on the CipherTrust Manager, it does not require a deletion schedule. The key can be deleted directly from the CipherTrust Manager. Refer to [Deleting External Keys](#).

To remove an Oracle key from the CipherTrust Manager:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the desired key with the state DELETED.
4. Click **Remove Key**. The Remove Key dialog box is displayed.
5. Click **Remove Key** to confirm the action.

A message stating that the key is removed successfully is displayed.

Removing a Key Backup

This section is applicable to Oracle BYOK keys.

When the synchronization is initiated, Oracle cloud allows backup of keys that:

- Are stored in Virtual Private Vaults (VPVs)
- Are stored in vaults that have associated bucket credentials
- Have the HSM protection mode

When an Oracle key is deleted from Oracle cloud, its status on the CipherTrust Manager becomes DELETED. You can remove the backup of such keys from the CipherTrust Manager.

To remove the backup of an Oracle key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the desired key with the state DELETED.
4. Click **Remove Key Backup**. The Remove Key Backup dialog box is displayed.
5. Click **Remove Key Backup** to confirm the action.

A message stating that the key backup is removed successfully is displayed.

Restoring a Deleted Key

This section is applicable to Oracle BYOK keys.

Only the keys with backup on the CipherTrust Manager can be restored.

To restore a deleted key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Restore Key**. The Restore Key dialog box shows is displayed.
5. Click **Restore** to confirm the action.

A message stating that the key is restored successfully is displayed.

Managing Oracle Tenancies

An Oracle tenancy is a secure and isolated partition of Oracle Cloud Infrastructure (OCI) to create, organize, and administer your cloud resources. You can add OCI tenancies manually (without Oracle connection) or using an [Oracle](#) connection configured on the CipherTrust Manager.

On the CipherTrust Manager, licensing for Oracle cloud is implemented based on OCI tenancies.

This section describes how to manage Oracle tenancies on CCKM.

- [Adding Oracle Tenancies](#)
- [Viewing Oracle Tenancies](#)
- [Removing Oracle Tenancies](#)

Adding Oracle Tenancies

To add an Oracle tenancy:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**. The Vaults tab of the Oracle Vaults page is displayed.
3. Click the **Tenancy** tab.
4. Click **Add Tenancy**. The Add Tenancy wizard is displayed.
5. Select one of the **Methods for choosing Oracle Tenancy**. You can select **Oracle Connection** or **Oracle Tenancy (no connection)**.

6. Click the desired tab to view the instructions.

Oracle Connection

Select a Connection.

Oracle Tenancy (no connection)

- a. Enter the **Tenancy** name.
 - b. Enter the **Tenancy OCID**.
7. Click **Add**. A message **Tenancy added successfully** is displayed on the screen.
 8. Click **Close**.

The newly created tenancy is displayed in the list of tenancies.

Viewing Oracle Tenancies

The Tenancy tab of the Oracle Vaults page shows the list of tenancies added to the CipherTrust Manager. Search for the tenancies by Tenancy Name or Tenancy OCID.

To view the list of tenancies added to the CipherTrust Manager:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**. The Vaults tab of the Oracle Vaults page is displayed.
3. Click the **Tenancy** tab.

The page displays the following details:

Column	Description
Tenancy	Name of the tenancy.
Tenancy OCID	OCID of the tenancy.
Date Added	Date and time when the tenancy was added.

To view/hide columns, click the Customize View (⚙️) icon, select/clear the desired option, and click **OK** to display the column.

Removing Oracle Tenancies

To remove an Oracle tenancy:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **KMS Containers > Oracle Vaults**. The Vaults tab of the Oracle Vaults page is displayed.
3. Click the **Tenancy** tab.
4. Click the overflow icon () corresponding to the desired tenancy.
5. Click **Remove Tenancy**. The Remove Tenancy dialog box is displayed.
6. Select **I wish to delete the tenancy**.
7. Click **Remove**.

A message **Tenancy deleted** is displayed on the screen.

Warning

Be extremely careful when deleting a tenancy. Once the tenancy is deleted, it will no longer be available for use.

Scheduling Operations

CCKM allows you to periodically refresh or rotate keys in the background. To refresh or rotate keys, you need to create schedule configuration. A schedule configuration defines when to run a refresh or rotation job. You can use either basic format or cron format to specify the time when the rotation or refresh job will run.

Specify basic format in the following order:

[Repeats, at]

Where,

- Repeats: frequency of the scheduler. Possible values are daily, weekly, monthly, and yearly.
- at: specific time at which the rotation or refresh operation will be performed. Possible value lies between 12:00-11:00 AM/PM UTC.

Specify cron format in the following order:

" * * * * * "

These five values indicate when the job should be executed. These values are mandatory and must be specified in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

Field	Allowed values	Allowed special characters
Minute	0-59	* / , -
Hour	0-23	* / , -

Field	Allowed values	Allowed special characters
Day of month	1-31	* / , - ?
Month	1-12 or JAN-DEC	* / , -
Day of week	0-6 or SUN-SAT	* / , - ?

Adding Key Rotation Schedule

A key rotation schedule replaces selected keys with new cryptographic keys at the specified time.

To add key rotation schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. **Select Schedule Type as Key Rotation.**
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **Oracle** from the **Cloud Name** drop-down list.
 - Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. On the **Schedule Config** screen, enter or select the following details.
 - a. Configure **DURATION** for the scheduler. Specify the schedule start and end time:
 - **Schedule Starts**: Specify time when the schedule starts.
 - **Schedule Ends**: Unavailable by default, that is, the scheduler never expires.
 - **Never**: Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.
 - b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
 - c. Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created schedule is displayed in the schedules list.

Adding Key Refresh Schedule

A key refresh schedule synchronizes keys from linked Oracle vaults to Cloud Key Manager.

To add a key refresh schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. **Select Schedule Type as Key Refresh.**
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **Oracle** from the **Cloud Name** drop-down list.
 - Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. On the "**Schedule Config**" screen, enter or select the following details.
 - a. Configure **DURATION** for the scheduler. Specify the schedule start and end time:
 - **Schedule Starts**: Specify time when the schedule starts.
 - **Schedule Ends**: Unavailable by default, that is, the scheduler never expires.
 - **Never**: Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.
 - b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
 - c. Click **Next**.
8. On the **Add Vaults** screen:
 - a. Select the desired vault from the **Refresh Keys from Selected Vault(s)** drop-down list.

- b. Click the **+** button. Similarly, you can add more vaults. To remove a vault, click the close (X) icon in the name of the vault.

Alternatively, if you want to refresh all the existing vaults, select the **Refresh All Vaults** check box. When selected, the **Refresh Keys from Selected Vault(s)** field becomes unavailable.

- c. Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created scheduled is displayed in the schedules list.

Viewing/Editing Schedules

To view/edit a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The Schedules page displays the following details:

Field	Description
Name	Unique name of the scheduler configuration.
Schedule Type	Type of the schedule. Possible types are: • Key Rotation • Key Refresh • Add KMS Container (GCP and Azure only)
Cloud Name	Name of the cloud.
Last Modified	Time when the schedule is modified.
Frequency	Frequency of the scheduler configuration.
Start Date	Start time of the scheduler configuration.
End Date	Expiry time of the scheduler configuration. Never is displayed if the schedule is set to never expire.
Status	Status of the scheduler configuration. Possible values are: • Enabled • Disabled

3. Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.

Alternatively, click the **Name** link corresponding to the desired schedule.

4. Edit or configure the following information:

- **GENERAL INFO:**

- Basic description of the schedule.
- Status of the scheduler configuration. Select **Enable Schedule** to enable the schedule, clear to disable.
- **SCHEDULE CONFIG**: Scheduler configuration parameters such as duration and frequency of the schedule. Refer to the key rotation [Schedule Config](#) screen or key refresh [Schedule Config](#) screen, as appropriate for details.
- **ORACLE VAULTS**: (For key refresh schedules only) Select vaults for the key refresh schedule.

To add a vault:

- a. Select the desired vault from the **Refresh Keys from Selected Oracle Vault(s)** drop-down list.
- b. Click **+** button. Similarly, you can add more vaults. To remove a vault, click the close (X) icon in the name of the vault.

Alternatively, if you want to refresh keys in all the existing vaults, select the **Refresh All Oracle Vaults** check box. When selected, the **Refresh Keys from Selected Oracle Group(s)** field becomes unavailable.

5. Click **Update** to save the changes.

The Schedules page also contains a section named **JOB HISTORY**. This section displays information related to a schedule such as **Run Date**, **Job ID**, **Status**, and **Errors**.

Viewing Keys Assigned to Schedules

Schedules for a key can be added, updated, and deleted on the KEY SCHEDULE section of the edit view of a key. Refer to [Adding or Changing Key Rotation Schedule](#) for details.

To view the keys assigned to a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The Schedules page is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.

Alternatively, click the **Name** link corresponding to the desired schedule.

The mini detail view of the schedule shows a numbered link next to the **Keys Assigned** label. The number indicates the number of keys assigned to the schedule. If no key is assigned, linked number **0** is displayed.

4. Click the numbered link. The Oracle Keys for Schedule: <schedule-name> page is displayed. The page displays the following details:

Field	Description
Name	Unique, user-friendly name of the Oracle key. Click

Field	Description
	the link to view additional details of the key or edit the key. Refer to Viewing or Editing Details of Oracle Keys . This name is useful in searching for specific keys.
Key ID	ID of the Oracle key.
Tenancy	Name of the Oracle tenancy.
Compartment	Name of the OCI compartment where the key resides.
Vault	Name of the OCI vault where the key resides.
Protection Mode	Protection mode for the key - HSM or Software.
Algorithm	Algorithm of the Oracle key. AES, RSA, and ECDSA algorithms with different keys sizes are supported.
State	State of the Oracle key.
Region	Region of the key.
Version	Version of the key.
Creation Date	Date and time when the Oracle key is created.
Deletion Date	Date and time when the Oracle key is deleted.
Origin	Source of the key material used for the version. The origin can be: • CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud. • External (Unknown): Source of the key material is unknown. It is different than CCKM and the native cloud. Refer to Key Creation Methods and Sources for details.

The **Region**, **Version**, **Creation Date**, and **Origin** columns are hidden by default. To show/hide a column, click the custom view icon () , select/clear the desired column, and click **OK**.

Disabling Schedules

To disable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.

-
3. Click the overflow icon () corresponding to the desired schedule and click **Disable**. The Disable Schedule message is displayed.
 4. Click **Disable Schedule**.

A message **Successfully disabled the schedule** is displayed on the screen.

Enabling Schedules

To enable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Enable**. The Enable Schedule message is displayed.
4. Click **Enable Schedule**.

A message **Successfully enabled the schedule** is displayed on the screen.

Manually Running Schedules

To manually run a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Run Now**.

The Run Now schedule is started in the background. A message **Schedule is running now. It will take a few seconds to finish.** is displayed on the screen. After the schedule is run successfully, a message **Successfully finished running the schedule** is displayed.

Deleting Schedules

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Delete Schedule**. The Delete Schedule message is displayed.
4. Click **Delete**.

A message **Successfully deleted schedule** is displayed on the screen.

Managing Oracle Reports

CCKM provides options to generate key visibility reports based on key-related activities between CCKM and Oracle vaults and rotation schedules. Use the reports to track keys by their expiration dates.

Oracle reports are categorized as:

- **Key Activity Report:** Inspect individual Oracle key histories by operations, for example, when they were refreshed, rotated, edited, or deleted. Also, use this report to compare key activities between CCKM and Oracle vaults.
- **Key Rotation Report:** Track keys that have been rotated or will be rotated between the selected period.
- **Key Aging Report:** Track Oracle keys by their expiration dates. Audit a range of dates, from past material deletions to future scheduled deletions, within selected Oracle vaults.

Creating Oracle Reports

Note

OCI supports key rotation reports for a period of maximum 14 days. When generating key rotation reports, make sure that the period between the start and end time is not more than 14 days.

To create an Oracle report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Oracle**. The Oracle Reports page is displayed.
3. Click **Add Report**. The Choose Report Type and Name screen of the Add Oracle Report wizard is displayed.
4. Select the report type. The options are **Key Activity Report**, **Key Rotation Report**, and **Key Aging Report**.
5. Specify a **Report Name**. This is a mandatory field.
6. Click **Next**. The Select Oracle Vault screen of the wizard is displayed.
7. Select a vault from the **Oracle Vaults** drop-down list. The drop-down list contains the list of Oracle vaults added to the CipherTrust Manager.
8. Click **Add**. The selected vault appears under the Selected Oracle Parameters section.

Similarly, you can add as many vaults as you want.

9. Click **Next**. The Set Start and End Dates screen of the wizard is displayed.
10. In the **Include Entries From** field, specify the start date and time for the report.

- Click the field and select the date and time on the on-screen calendar.
- Alternatively, enter the time in **MM/DD/YYYYHH:MM** format.

11. In the **To** field, specify the end date and time for the report.

12. Click **Save**.

A success message **Created report** is displayed. The newly created report is displayed under the Oracle reports list. The report Run Date becomes *Running Now* and Status becomes *In Progress*. After the run is complete, the date and time of the run is displayed and the Status becomes Completed.

Viewing Oracle Reports

The Oracle Reports page displays the list of existing reports. Filter the reports by their names.

To view existing Oracle reports:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Oracle**. The list of available reports is displayed. The following details of the reports are displayed:

Column Name	Description
Report Name	Name of the report.
Run Date	When the report is run. If a report is running currently, Running Now is displayed.
Type	Type of the report. The type can be: • Key Activity • Key Rotation • Key Aging
Start Date	Start date from when the report is generated.
End Date	End date for the report.
Status	Status of the report. The status can be: • Completed • In Progress

Viewing Details of an Oracle Report

To view the details of an Oracle report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Oracle**. The list of available reports is displayed.

- Click the desired **Report Name** link.

Alternatively, click the overflow icon () corresponding to the desired report and click **View**.

For a Key Aging report, the REPORT DETAILS section shows the following details of the reports:

Column Name	Description
Key Name	Name of the key.
Oracle Key ID	ID of the Oracle key.
Vault ID	ID of the Oracle vault where the key resides.
Time of Deletion	Date and time when the key is/will be deleted.

For a Key Activity report, the REPORT DETAILS section shows the following details of the reports:

Column Name	Description
Key Name	Name of the key.
Oracle Key ID	ID of the Oracle key.
Vault ID	ID of the Oracle vault where the key resides.
Key Activity	Activity performed on the key.
Oracle User	Name of the Oracle user who created the report.
Modified Time	Date and time when the key is modified.

Downloading Oracle Reports

To download an Oracle report as a CSV file:

- Open the **Cloud Key Manager** application.
- In the left pane, click **Reports > Oracle**. The list of available reports is displayed.
- Click the overflow icon () corresponding to the desired report and click **Download**.

Alternatively, click the desired report under **Report Name**, and click **Download CSV** in the detail view.

The report is downloaded as a CSV file.

Deleting Oracle Reports

To delete an Oracle report:

- Open the **Cloud Key Manager** application.

-
2. In the left pane, click **Reports > Oracle**. The list of available reports is displayed.
 3. Click the overflow icon () corresponding to the desired report and click **Delete**. The Delete Report message is displayed.
 4. Click **Delete**.

The report is removed from the list of reports.

Managing Oracle Compartments

An Oracle key belongs to a compartment and a vault at the same time. When an Oracle vault is added to the CipherTrust Manager, linked Oracle compartments are automatically added. You can also add compartments to CCKM manually. One compartment can have multiple Oracle vaults associated with it.

When the last vault linked with an Oracle compartment is deleted from the CipherTrust Manager, the linked compartment is also deleted.

The Compartments tab of the Oracle Vaults page shows the list of compartments linked with the vaults added to the CipherTrust Manager. Search for the compartments by Tenant.

Adding Oracle Compartments

To add an Oracle compartment:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **Compartments** tab.
4. Click **Add Existing Compartment**.
5. Select the desired **Connection**.
6. Select the desired **Compartment ID** from the list. The list shows the Oracle compartments associated with the selected connection.
7. Expand the compartment ID to view the compartment hierarchy.
8. Select the desired compartments from the hierarchy.
9. Click **Save**.

The newly added compartments are displayed in the list of Oracle compartments.

Viewing Oracle Compartments

To view the list of compartments:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **Compartments** tab. The tab displays the following details:

Column	Description
Compartment	Name of the Oracle compartment.
Tenant	Name of the Oracle tenant.
Date Added	Date and time when the compartment is added to the CipherTrust Manager.

Deleting Oracle Compartments

To delete an Oracle compartment:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **Compartments** tab. The tab displays the list of Oracle compartments added to CCKM.
4. Click the overflow icon () corresponding to the compartment you want to delete.
5. Click **Remove Compartment**.
6. Select **I wish to delete the compartment**.
7. Click **Delete**.

The compartment is deleted successfully.

Downloading Oracle Metadata

Downloading Oracle Keys Metadata

CCKM users with permissions to view the Oracle keys can export and download the keys metadata displayed on the CCKM GUI. Filter the keys metadata to display on the GUI based on your requirements, for example, all keys created in the past 12 months, all keys residing in a specific Oracle vault, and all keys based on a specific algorithm. You can then download the filtered metadata as a CSV file.

To download the Oracle keys metadata visible:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**. The Oracle Keys page is displayed.
3. Filter the keys metadata based on your requirements.

Use the Search box to filter the keys. To further filter the metadata, show/hide the desired columns and use the column-level filters.

4. Click **Download as CSV**.

The filtered Oracle keys metadata is downloaded as a CSV file.

OCI External Resources

Oracle Cloud Infrastructure (OCI) Key Management Service (KMS) with external vault capability allows you to use a third-party on-premises key management system like CipherTrust Manager. This provides greater control over encryption keys that protect your data in the Oracle cloud.

CipherTrust Cloud Key Manager (CCKM) on the CipherTrust Manager uses external keys for performing cryptographic (encrypt/decrypt) operations on demand, while preserving end-user control to manage external keys. The lifecycle of external keys on the CipherTrust Manager is managed by CCKM only. The OCI KMS does not have any control over these keys.

The CipherTrust Manager exposes a network endpoint. This allows the OCI KMS to forward any cryptographic requests to the CipherTrust Manager for processing.

Prerequisites

Before integrating the CipherTrust Manager as an External Key Manager (EKM), make sure the following.

CipherTrust Manager

- A CipherTrust Manager is deployed. Refer to [Getting Started](#) for details.
- The NTP server is added. Refer to [Add an NTP Server](#) for details.
- The web interface must have a TLS certificate signed by an external or local Certificate Authority (CA). The CA bundle should be provided while creating a [private endpoint](#) on OCI.
- The Subject Alternative Name (SAN) in the web interface certificate must have specific IP Addresses.
 - A good default value for the SAN is the IP address of the CipherTrust Manager.
 - If you are using a load balancer, and it is set to do SSL passthrough, include the IP address of the load balancer in the SAN.
 - If you are using FQDN, SAN should be the FQDN of the CipherTrust Manager. Refer to [Adding the web certificate for FQDN connectivity](#) for details.

Refer to [SSL/TLS server certificate \(NAE, KMIP, and WEB\)](#) for details.

Note

- OCI does not provide any option to update the IP address or the CA bundle in private endpoints. If you change the CA of the web interface certificate or the IP address of the CipherTrust Manager or Load Balancer, the network connectivity is disrupted. As a result, OCI faces network connectivity issues and all operations with the CipherTrust Manager stop working. In this case, you need to reconfigure the complete setup on OCI.
- It is recommended to use the IAM Domain version of Oracle Identity Cloud Service (IDCS) for managing OCI vaults under EKM and keys (OCI HYOK). To use OCI HYOK with the tenancy that doesn't support the IAM Domain version, please contact [Thales Customer Support](#).

Oracle Cloud Infrastructure (OCI)

1. A Virtual Cloud Network (VCN) is created in OCI Networking. Refer to the [Creating a Virtual Cloud Network](#) for details.
2. (Optional, recommended) FastConnect (Co-lo) is created in OCI Networking. This is required to have direct connectivity between the OCI VCN and the CipherTrust Manager on premises. Refer to the [FastConnect](#) documentation for details.
3. Policies are created in OCI Identity.

A policy is a document that specifies who can access which OCI resources that your company has, and how. A policy simply allows a group to work in certain ways with specific types of resources in a particular compartment.

Refer to [Getting Started with Policies](#) for details.

4. A private endpoint is created. Refer to [Creating a Private Endpoint](#) for details.

Note

For FQDN connectivity:

Configuration using FQDN requires a private API Gateway in a private subnet. The private subnet must have a NAT Gateway to route requests from the API Gateway to EKM.

While creating a private endpoint, you need to provide the IP Address of OCI API Gateway in the External Key Management Private IP address field. To get the IP address.

- a. [Create an OCI API gateway with FQDN](#).

- b. Deploy the API gateway with FQDN.

After ensuring the prerequisites, perform the steps described in the subsequent sections.

Note

CCKM supports both commercial and dedicated OCI regions. Thales has validated only the commercial regions, while Oracle has validated the dedicated regions. For any issues related to the dedicated regions, contact the Oracle support team.

Add OCI Tenancy on CipherTrust Manager

On the CipherTrust Manager, add an OCI tenancy. You can add OCI tenancies manually (without Oracle connection) or using an [Oracle](#) connection configured on the CipherTrust Manager. Refer to [Adding Oracle Tenancies](#) for details.

After adding the OCI tenancy, register the CipherTrust Manager with OCI Identity Domain as a confidential application.

Register CipherTrust Manager with OCI Identity Domain

Register the CipherTrust Manager (an External Key Manager) as a confidential resource application with Oracle Identity Domain and enable the client credentials grant, if the Identity Domain is protected.

While registering the CipherTrust Manager, add the `oci_ekms` scope. Refer to [Creating Confidential Resource App](#) for details.

After registration, the CipherTrust Manager application's client credentials are generated if the Identity Domain is protected. Note down the credentials and the Identity Domain URL. These will be required for registering the JWT issuer (identity provider).

Register Identity Provider with CipherTrust Manager

Register an identity provider on the CipherTrust Manager. Specify the JWKS URI or OpenID configuration URL (the Identity Domain URL and the suffix string, `/.well-known/openid-configuration`) and CipherTrust Manager application's client credentials (if the OCI Identity Domain is protected). These details are generated during the step [Register CipherTrust Manager with OCI Identity Domain](#).

To register the identity provider with the CipherTrust Manager:

1. Log on to the CipherTrust Manager GUI.
2. Add the identity provider. Refer to [Creating Identity Providers](#) for details.

CipherTrust Manager verifies the provided JWKS URI and credentials from the OCI Identity Domain and saves the credentials securely on successful verification.

A JWT issuer ID is generated. This ID is required when creating an external vault on the CipherTrust Manager.

Register OCI KMS Application

Register the OCI KMS application as a confidential client application and enable the client credentials grant. Next, bind the CipherTrust Manager, which you registered [as a confidential resource application](#). This binding is required to associate the OCI KMS client to the CCKM authorized resource (CipherTrust Manager).

Refer to the Oracle documentation for details.

On OCI:

1. Register the OCI KMS Client application as a confidential client application.
2. Bind the OCI KMS Client application with the CipherTrust Manager application that you registered [as a confidential resource application](#).

After the KMS client application registration and binding, a KMS client ID is generated. This ID is required for creating an external vault on the CipherTrust Manager.

Creating External Vaults

To create an external vault:

1. On the **CipherTrust Manager**, create an external vault using the KMS client ID created in [Register OCI KMS Client Application](#) and the JWT issuer ID generated in [Register Identity Provider with CipherTrust Manager](#).

During this step, an external vault is created locally on the CipherTrust Manager and an *external vault endpoint URI* is generated. This URI can be found in the details of the external vault resource.

Refer to [Adding External Vaults](#) for details.

2. On **OCI**, create a vault under EKM using the *external vault endpoint URI* generated in the previous step. Refer to the [External Key Management Service](#) documentation for details.

Note

If the CipherTrust Manager's web interface is configured on a port other than 443, the port number should be added to the *external vault endpoint URI* while creating a vault under EKM on OCI.

The vault created under EKM on the OCI KMS is mapped with the external vault created on the CipherTrust Manager.

Creating External Keys

To create an external key:

1. On the **CipherTrust Manager**, create a key in the external vault created on the CipherTrust Manager (refer to the previous step). You can select an existing source key or create a new key.

An *external key ID* is generated. This external key ID can be found in the details of the external key resource.

Refer to [Adding External Keys](#) for details.

2. On **OCI**, create an external key reference using the *external key ID* generated in the previous step. Refer to the [External Key Management Service](#) documentation for details.

The external key reference created on the OCI KMS is mapped with the external key created on the CipherTrust Manager.

Rotating External Keys

To rotate an external key:

1. On the **CipherTrust Manager**, create a new key version in the external vault created on the CipherTrust Manager (refer to the previous step). You can select an existing source key or create a new key.

A *version ID* is generated. This version ID can be found in the details of the external key resource.

Refer to [Adding a Key Version](#) for details.

2. On **OCI**, rotate an external key reference using the *version ID* generated in the previous step. Refer to the [External Key Management Service](#) documentation for details.

The external key reference created on the OCI KMS is mapped with the version ID created on the CipherTrust Manager.

Deployment Tutorials

- [Deploy Thales CCKM in OCI and Set Up a Cluster with CA Role](#)
- [Set Up Oracle Hold Your Own Key \(HYOK\) with CipherTrust Manager \(No API Gateway\)](#)
- [Set Up Oracle Hold Your Own Key \(HYOK\) with CipherTrust Manager \(With API Gateway\)](#)
- [Enable OCI HYOK with Thales CipherTrust Manager Using OCI Load Balancer for High Availability](#)

What Next?

After completing the prerequisites, you can manage external vaults and keys on the CipherTrust Manager.

- [Managing Identity Providers](#)
- [Managing External Vaults](#)
- [Managing External Keys](#)

- Performance Summary

Managing Identity Providers

This section describes how to manage identity providers on CCKM.

Identity providers can be added, viewed, modified, or deleted on the External Vaults tab of the Oracle Vaults page. Identity providers are required when adding the external vaults, refer to [Managing External Vaults](#) for details.

Creating Identity Providers

An issuer or an openId configuration URL is required to create an identity provider.

To create an identity provider:

1. Open the **Cloud Key Manager** Application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **External Vaults** tab.
4. Scroll down the page, go to the **IDENTITY PROVIDERS** section, and click **Add Identity Provider**. The General Info tab of the Add Identity Provider screen is displayed.

General Info

1. Specify a unique **Name** for the provider. This is a mandatory field.
2. **Select Provider Verifier**. This is a mandatory field. The verifier can be **Issuer** or **OpenID Configuration URL**.

Note

- **OpenID Configuration URL** is the recommended option.
- If you select the **Issuer** option, you need to specify a combination of **Issuer** and **jwksURL**.
- If you select the **OpenID Configuration URL** option, you need to specify a **OpenID Configuration URL**.

Make sure to provide the exact CipherTrust Manager application's credentials, as received from Oracle; otherwise, external APIs might cease to work as expected. Currently, only Oracle IDCS is supported as identity provider and the issuer should be `https://identity.oraclecloud.com/`.

3. (If Oracle IDCS is configured in the protected mode), select **jwks Protected URL** and specify the following:
 - **Client ID**: Client ID of the CipherTrust Manager application as registered on third-party IDP.

- **Client Secret:** Client secret of the CipherTrust Manager application as registered on third-party IDP.

4. Click **Next**. The Regional Settings (Optional) tab of the Add Identity Provider screen is displayed.

Regional Settings (Optional)

1. Enter **Regional jwks URL(s)** or **Regional OpenID Config URL(s)**, depending on the Provider Verifier you have selected in the General Info tab.
2. Click **Add Regional URL(s)**.

Note

You can add multiple regional URLs.

3. Click **Add**.
4. Click **Close**.

The newly created identity provider appears in the providers list. Similarly, add as many identity providers as required.

To enable multi-region functionality with FQDN connectivity:

1. In the Oracle Cloud Console, create an API Gateway and a private endpoint in the **destination region**. Ensure both are in the same Virtual Cloud Network (VCN).
2. In the IDCS domain for the destination region, create **two new resource applications**:
 - **First application:** Set the primary audience to the static IP address of the API Gateway in the destination region.
 - **Second application:** Set the primary audience to the FQDN of your application CipherTrust Manager.

Add a scope to each application, and include both scopes in the resources field of the existing **client application**.

Note

This results in a configuration with three resource applications and one client application that includes the scopes from all three.

3. In your application (CipherTrust Manager), add the regional IDCS URLs for the **source** and **destination** regions to the Identity Provider (IDP), and the primary OCI region.

4. On the **Replicate Vault** page, replicate the vault from the source region. You will need to select the destination region, IDCS domain name, and private endpoint (PE).

Viewing Identity Providers

The Oracle Vaults page shows the available identity providers. The IDENTITY PROVIDERS section shows the Name, Issuer, OpenID Configuration URL, and jwksURL.

To view identity providers:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **External Vaults** tab.
4. Navigate to **IDENTITY PROVIDERS**. The list of identity providers is displayed.

The following details are shown:

Column	Description
Name	Name of the identity provider.
Issuer	Issuer string from the identity provider JWT.
OpenID Configuration URL	URL of the OpenID configuration.
jwksURL	URL of JWKS.

Editing Identity Providers

To edit identity providers:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **External Vaults** tab.
4. Navigate to **IDENTITY PROVIDERS**. Under Name, click the desired profile. The details screen of the Identity Provider is displayed.

Alternatively, click the overflow icon  corresponding to the desired profile and click **View/Edit**.

Under GENERAL INFORMATION

1. (Optional) Update **Name**.
2. (Optional) Update **jwks Protected URL**.

- a. Enter a new **Client ID**.
- b. Enter a new **Client Secret**.

Note

Do not leave the Client ID and Client Secret empty.

3. Click **Update**.

Under **REGIONAL SETTINGS**

1. (Optional) Add new regional URL(s).
 - a. Enter new **Regional jwks URL(s)** or **Regional OpenID Config URL(s)**, depending on the Provider Verifier that has been selected for this identity provider.
 - b. Click **Add Regional URL(s)**.
2. (Optional) Edit existing regional URL(s).
 - a. Click the **Edit** icon corresponding to the desired regional URL.
 - b. Update **regional URL**.
 - c. Click **Save**.
3. (Optional) Deleting the existing regional URL(s).

Click the **Delete** icon corresponding to the desired regional URL.
4. Click **Update**.

Deleting Identity Providers

When an identity provider is no longer needed, delete it from the CipherTrust Manager. Before deleting the provider, ensure that it is not in use by any external vaults.

To delete an identify provider:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **External Vaults** tab.
4. Under **IDENTITY PROVIDERS**, click the overflow icon  corresponding to the to the desired provider and click **Delete**. The **Delete Identity Provider** dialog box appeared on the screen.

Are you sure you want to delete? message is displayed in the dialog box.

- Click **Delete Identity Provider** to confirm the deletion.

Note

If the issuer (identity provider) is associated with an existing external vault, you will see an **Error in deleting identity provider** message. You need to modify the association to delete the issuer.

A success message is displayed on the screen. The identity provider is removed from the providers list.

Managing External Vaults

This section describes how to manage external vaults on CCKM. Before proceeding, [an identity provider](#) must be created.

External vaults can be added, viewed, modified, or deleted on the External Vaults tab of the Oracle Vaults page of the CipherTrust Cloud Key Manager (CCKM) GUI.

Adding External Vaults

You can add external vaults linked to an Oracle connection to the CipherTrust Manager. An external vault can only be added just once.

To add an external vault:

- Open the **Cloud Key Manager** application.
- In the left pane, click **KMS Containers > Oracle Vaults**. The Vaults tab of the Oracle Vaults page is displayed.
- Click the **External Vaults** tab.
- Click **Add External Vault**. The Add External Vault wizard is displayed.
- Enter a unique **Vault Name**.
- From **Methods for choosing Oracle Tenancy**, select one of the methods. The options are:
 - **Oracle Connection:** Select to specify an Oracle connection. The **Select a Connection** drop-down list is displayed.

Refer to [Connection Manager](#) to create an Oracle Connection if needed.
 - **Oracle Tenancy (no connection):** Select to specify a tenancy. The **Tenancy** drop-down list is displayed.
- Depending on the method selected in the previous step, **Select a Connection** or **Tenancy** from the drop-

down list.

8. (Optional) Switch the **Enable success audit events** toggle on or off to control the audit recording of success logs for the specified endpoint. By default, this toggle is set to active to troubleshoot performance issues.

Note

For improved performance, it is recommended to switch off the **Enable success audit events** toggle.

9. (Optional) Select a **Source Key Tier**. The options are:

- CipherTrust
- Luna HSM, also **Select a Luna Partition**. FM-enabled Luna HSM is not supported as a key source.

If you want to clear the selected Source Key Tier, click **Clear**.

10. **Select an Issuer** from the drop-down list.
11. Enter **Client Application ID** of the OCI KMS application as registered in [Register OCI KMS Application](#).
12. In **Endpoint URL Hostname**, specify the IP address for the OCI external vault endpoint URL. Specify the IP address of the CipherTrust Manager or Load Balancer. In case of FDQN connectivity, specify the IP address of the OCI API gateway.
13. (Optional) In **Endpoint URL Port**, specify the port number for the OCI external vault endpoint URL. Specify the port of the CipherTrust Manager or Load Balancer. The default port is **443**.

Note

The web interface port within CipherTrust Manager can be changed from the default port of **443** to another port. If you plan to change the default port for the CipherTrust Manager web interface, ensure to change it **before** configuring the OCI EKMS (HYOK) on CCKM. Also, reflect this port change when creating an external vault on CCKM. Changing the default port **after** configuring OCI EKMS is not supported. Refer to [Support for Changing the Default Port of Web Interface Setting](#) for instructions on changing the port when configuring OCI EKMS.

14. (Optional) Enter **Policy**.
15. Click **Add**.
16. Click **Close**.

The selected vault is displayed on the External Vaults tab of the Oracle Vaults page.

The vault is available for [adding external keys](#).

Viewing External Vaults

To view the list of the external vaults added to the CipherTrust Manager:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **External Vaults** tab. The External Vaults tab of the Oracle Vaults page shows the list of external vaults added to the CipherTrust Manager.

The page displays the following details:

Column	Description
Name	Name of the external vault.
External Vault URL	URL of the external vault.
Status	Status of the external vault, BLOCKED or UNBLOCKED.
Tenancy	Name of the Oracle tenancy.
Identity Provider	Name of the identity provider.
Connection	Name of the Oracle connection added to the CipherTrust Manager.
Source Key Tier	Name of the Source Key Tier, Luna HSM and CipherTrust.
Date Added	Date and time when the external vault was added.

To view/hide columns, click the Customize View () icon, select/clear the desired option, and click **OK** to display the column.

Viewing or Editing Details of an External Vault

To view or edit the details of an external vault on CCKM:

1. Open the **Cloud Key Manager** application.
 2. In the left pane, click **KMS Containers > Oracle Vaults**.
 3. Click the **External Vaults** tab.
 4. On the External Vaults tab, click the **Name** link of the desired vault.
-

Alternatively, click the overflow icon () corresponding to the desired vault, and click **View/Edit Details**.

The ORACLE VAULT DETAILS page shows additional details of the selected vault under the GENERAL INFO and ACCESS CONTROL sections. Expand each section to view and edit their details.

The GENERAL INFO section provides details of Endpoint URL Hostname, Identity Provider, and Policy. If needed, you can edit these details, as appropriate.

Editing Details of an External Vault

You can change the endpoint URL hostname, identity provider, and policy of an external vault on the CipherTrust Manager.

1. Expand **GENERAL INFO**.
2. Update the **Endpoint URL Hostname**. Specify the IP address of the CipherTrust Manager or Load Balancer. *FQDNs are not supported.*
3. Update the **Endpoint URL Port**. Specify the port number for the OCI external vault endpoint URL. Specify the port of the CipherTrust Manager or Load Balancer. The default port is 443.
4. Update the **Identity Provider** from the drop-down list.
5. Update the **Policy**. Specify a valid Rego policy.
6. Click **Update**.

Managing User Permissions on External Vaults

To work with the external resources, users/groups must have the minimum set of permissions that allow them to use the external resources such as external keys and vaults. Initially, the CCKM user only has permission to view the external keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform operations on external vaults.

Users with the following characteristics can perform operations on external keys and vaults:

- Users in the **CCKM Admins** group
- Users in the **Admin** group
- Users who are administrators for a **domain**
- Users who are in the **CCKM Users** group and which have had a CCKM Admin assign permissions through the UI or the `/v1/cckm/oci/vaults/{id}/update-acls` endpoint in the REST API.

Adding Permissions for a User/Group

To add permissions for a user/group:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **External Vaults** tab.
4. On the External Vaults tab, click the **Name** link of the desired vault.

Alternatively, click the overflow icon () corresponding to the desired vault, and click **View/Edit Details**.

5. Expand the ACCESS CONTROL section.
6. Click **Assign User/Group**. The **Assign User/Group** dialog box is displayed.
7. Select the desired user or group from the **User/Group** drop-down list.
8. Click **Save**.

The newly added user/group is displayed under **Name** in the ACCESS CONTROL section. You can now grant additional permissions to the user/group, as appropriate. Refer to [Granting Permission to Perform an Operation](#) for details.

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. In the ACCESS CONTROL section, select the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

To revoke permissions from a user/group, refer to [Removing a Permission](#) for details.

Removing a Permission

To remove a permission assigned to a user or group:

1. In the ACCESS CONTROL section, clear the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. In the ACCESS CONTROL section, under **Unassign**, click the **X** button corresponding to the desired user/group.
2. On the **Remove User / Remove Group** screen, click **Remove**.

Note

Removing this user/group will remove all permissions currently assigned to the user/group.

3. Click **Remove** to confirm the action. To cancel the action, click **Keep It**.

A success message is displayed on the screen.

Removing External Vaults

External vaults can be removed on the Oracle Vaults page. Search for existing external vaults using Vault Name, Tenant, or Compartment.

Note

- Removing an external vault is an irreversible operation which permanently deletes unique identifiers required by the corresponding OCI vaults under External Key Manager (EKM).
- Before deleting an external vault from the CipherTrust Manager, ensure that the vault contains no external keys. Delete all external keys stored in the vault.

To remove an external vault from the CipherTrust Manager:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **External Vaults** tab.
4. On the External Vaults tab, click the overflow icon () corresponding to the vault you want to remove.
5. Click **Remove Vault**.
6. Select **I wish to delete the vault**.

-
7. Click **Delete**.

The external vault is deleted successfully. It is removed from the list of external vaults.

Blocking External Vaults

External vaults can be blocked on the Oracle Vaults page. Search for existing external vaults using Vault Name, Tenant, or Compartment.

When you block an external vault, all the external keys in the vault will be blocked. Any subsequent requests to the keys will fail. Blocking an external vault will also prevent any cryptographic operations on the key from the OCI KMS as well.

To block an external vault from CipherTrust Manager:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **External Vaults** tab.
4. On the External Vaults tab, click the overflow icon () corresponding to the external vault you want to block.
5. Click **Block**.

All keys in the vault will be blocked. Any subsequent requests to the keys will fail.

6. Click **Confirm**.

The external vault is blocked successfully. The status of external vault changes to BLOCKED.

Unblocking External Vaults

Blocked external vaults can be unblocked on the Oracle Vaults page. Search for blocked external vaults using Vault Name, Tenant, or Compartment.

To unblock an external vault from the CipherTrust Manager:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Oracle Vaults**.
3. Click the **External Vaults** tab.
4. On the External Vaults tab, click the overflow icon () corresponding to the external vault you want to unblock.
5. Click **Unblock**.

All keys in the vault will be unblocked.

6. Click **Confirm**.

The external vault is unblocked successfully. The status of external vault changes to UNBLOCKED.

Managing External Keys

This section describes how to manage external keys on CCKM. Before proceeding, you must have an external vault added to the CCKM. Refer to [Managing External Vaults](#) for details.

- [Adding External Keys](#)
- [Viewing or Editing Details of External Keys](#)
- [Adding a Key Version](#)
- [Blocking External Keys](#)
- [Unblocking External Keys](#)
- [Deleting External Keys](#)

Note

- This section describes the operations specific to the external keys.
- External keys do not support the enable key, disable key, schedule key deletion, move resource, and restore deleted keys operations.
- All other operations that can be performed on external keys are similar to those for Oracle BYOK keys, as described in [Managing Oracle Keys](#).
- CCKM doesn't support FM-enabled Luna HSM as a key source.

Adding External Keys

To add an external key.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click **Add Key**. The Key Material Origin screen of the Add Oracle Key wizard is displayed.

Key Material Origin

1. Select **Oracle External (HYOK)** as the **Origin Type**.
2. Select an external **Vault** from the drop-down list. The list shows available external and default vaults.

The **Oracle Compartment** field is unavailable (read-only) for external vaults.

3. Select **CipherTrust (Local)** or **Luna HSM** as the **Source**.

Note

- You can select Luna HSM as the source, if the selected external vault is not CipherTrust.
- FM-enabled Luna HSM is not supported as a key source.

- a. Click **Next**. The Source Key screen is displayed.

Source Key

1. Click the desired source for instructions.

CipherTrust (Local)

Select the **Source Key Material**. This specifies how to create the key. The options are:

- **Create New Key**: Click to create a fresh key. Specify an **Oracle Key Name**.
- **Select Existing Key**: Click to create a new key by selecting an existing key. Select an existing key from the **Select a source key** drop-down list.

Luna HSM

- a. Select the **Source Key Material**, the options are **Create New Key** and **Select Existing Key**.
- b. Click the desired tab to view the instructions.

Create New Key

- a. **Create New Key**: Click to create a fresh key.
- b. Select a **Partition** from the drop-down list.
- c. Enter the **HSM Key Name**.
- d. Update the **Key Usage** from the available options.

Select Existing Key

- a. **Select Existing Key**: Click to create a new key by selecting an existing key.
- b. Select a **Partition** from the drop-down list.
- c. Select a **HSM Key Name** from the drop-down list.

-
2. Click **Next**. The Configure Oracle Key screen is displayed.

Configure Oracle Key

1. Enter a unique, user-friendly alias as the **Oracle Key Name**. This will be the key name on the Oracle cloud. This name helps uniquely identify an external key.
2. Enter a Rego policy to be associated with the external key.
3. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, Source Key, and HYOK KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **Source Key** and **HYOK KEY** sections and update the details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the HYOK KEY section becomes Complete and the Oracle Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Oracle Key wizard is closed.

The newly created key is displayed in the list of Oracle keys.

Viewing or Editing Details of External Keys

After a key is created, you can add tags to it, schedule their rotation, and view its versions.

In the edit view of a key, you can view all the key details such as its ID, compartment, vault, state, algorithm, and region etc.

To view or edit an external key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**. The list of available keys is displayed.
3. Click the overflow icon () corresponding to the desired key and click **View/Edit Details**. Alternatively, you can click the key name link. The edit view of the key is displayed. The edit view is divided into:
 - **GENERAL INFO**: View and update the key name and policy. Refer to [Updating Key Name and Policy](#) for

details.

- **KEY SCHEDULE:** Add, update, and disable a key rotation schedule. Refer to [Adding or Changing Key Rotation Schedule](#) and [Disabling Key Rotation Schedule](#).
- **KEY VERSIONS:** View details of key versions. Refer to [Viewing Key Version Details](#).

Updating Key Name and Policy

To update the key name and policy.

1. Edit **Key Name**.
2. Edit **Policy**.
3. Click **Update**.

Adding or Changing Key Rotation Schedule

To add or update a key rotation schedule:

1. Expand the **KEY SCHEDULE** section.
2. From the **Select Rotation Schedule** drop-down list, select the desired schedule.
3. Select the **Key Origin**. The options are:
 - CipherTrust (Local)
 - Luna, also select the Luna HSM **Partition**. FM-enabled Luna HSM is not supported as a key source.
4. Click **Update**.

The key rotation schedule is added/updated. The selected schedule is now assigned to the key. To view all the keys assigned to a schedule, refer to [Viewing Keys Assigned to Schedules](#).

Disabling Key Rotation Schedule

To disable a key rotation schedule:

1. Expand the **KEY SCHEDULE** section.
2. Next to the **Select Rotation Schedule** drop-down list, click the close icon (.

Auto key rotation is disabled.

Viewing Key Version Details

To view the details of key versions, expand the **KEY VERSIONS** section. The key version details are displayed.

Adding a Key Version

To add a new key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Add Version**. The Add Version dialog box is displayed.
5. Select a **Method**, the options are **Create New Key** and **Select Existing Key**.
6. Click the desired tab to view the instructions.

Create New Key

- a. Select a **Source**, the options are **CipherTrust (Local)** and **Luna HSM**.
- b. Click the desired tab to view the instructions.

CipherTrust (Local)

Enter **Key Name**.

Luna HSM

- a. Select a **Partition** from the drop-down list.
- b. Enter the **HSM Key Name**.
- c. Select the **Key Type** from the available options. The supported key types are AES and RSA.
- d. Select the **Key Size** from the available options.
- e. Select the **Key Usage** from the available options.

Select Existing Key

- a. Select a **Source**, the options are **CipherTrust (Local)** and **Luna HSM**.
- b. Click the desired tab to view the instructions.

CipherTrust (Local)

Select the desired **Source Key** from the drop-down list.

Luna HSM

- a. Select the **Key Type** from the available options. The supported key types are AES and RSA.

- b. Select the **Key Size** from the available options.
- c. Enter the **HSM Key Name**.

Note

It is not recommended to delete a source key that is being used for cryptographic operations.

7. Click **Add Version**.

Blocking Oracle Keys

You can block a key that belongs to external vaults. To block a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Block**. The Block Key dialog box is displayed.
5. Click **Block Key** to confirm the action.

The key is blocked. The state of the external key changes to Blocked.

Unblocking Oracle Keys

You can unblock a blocked key that belongs to external vaults. To unblock a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Unblock**. The Unblock Key dialog box is displayed.
5. Click **Unblock Key** to confirm the action.

The key is unblocked. The state of the external key changes to Unblocked.

Deleting External Keys

An external key (HYOK) resides on the CipherTrust Manager, it does not require a deletion schedule. The key can be deleted directly from the CipherTrust Manager.

Caution

Before deleting an external key, ensure that the key is not being referred to by the OCI KMS.

To delete an external key from the CipherTrust Manager:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Oracle**.
3. Click the overflow icon () corresponding to the key to be deleted.
4. Click **Remove Key**. The Remove Key dialog box is displayed.
5. Click **Remove Key** to confirm the action.

A message stating that the key is removed successfully is displayed.

OCI EKMS Performance Summary

We have tested different environments to capture performance metrics for the CipherTrust Cloud Key Manager (CCKM) Oracle Cloud Infrastructure (OCI) External Key Manager Service (EKMS) integration. The performance results provided within this document demonstrate the effects of deployment choices on throughput for OCI HYOK key encrypt operations for one key. This can help you plan your CipherTrust Manager deployment to meet your performance needs for OCI EKMS integration.

All environments used an open-source [k6.io](#) tool as the REST client to simulate encrypt requests from OCI External Vault. The client ran on a virtual machine with 50 GB system volume, 8 CPUs, and 64 GB of memory.

Note

Actual performance numbers in your environment may be different. The results can vary based on factors, such as how and where the CipherTrust Manager is deployed, CipherTrust Manager resources, the location of cloud KMS, the key source of your choice, the network connectivity between the CipherTrust Manager, OCI Cloud and your key source, as well as how the traffic is load balanced.

Tested Environments

The following deployments were tested:

- CipherTrust Manager as a Key Source
 - Single Node of CipherTrust Manager on Premises, K6 Client on Premises
 - Single Node of CipherTrust Manager on Oracle, K6 Client on Oracle

- Two Clustered CipherTrust Manager Nodes on OCI, K6 Client on OCI
- Single Node of CipherTrust Manager on AWS, K6 Client on OCI
- Single Node of CipherTrust Manager on Premises, K6 Client on CipherTrust Manager, Zero Network Latency
- Luna HSM as a Key Source
 - Single Node of CipherTrust Manager on Premises, K6 client on Premises, Single Node of HSM
 - Single Node of CipherTrust Manager on Premises, K6 Client on Sanjose Lab, HSM in HA Mode
 - Single Node of CipherTrust Manager on Premises, K6 Client on the CipherTrust Manager, Zero Network Latency

The following CipherTrust Manager release version was employed in the tested deployments:

Description	Value
Release version	2.15.0.10938 RC1

Test Process

The test consisted of starting a given number of virtual users to perform encrypt operations on the OCI EKMS/HYOK key. Each user simulated a separate thread.

The total test duration was 40 seconds for each reading. The test was divided into the following increments:

- The ramp-up time was 5 seconds. Virtual users were started.
- The test duration was 30 seconds for each reading. Virtual users make wrap requests during this time.
- Ramp-down time was 5 seconds. Virtual users were stopped until there were zero active virtual users.
- In all environments, the acceptable response latency threshold by OCI is assumed as 100ms. But, for a zero network latency environment, it is assumed as 30ms.

Repeating the test in your environment

We have published the scripts used with `k6 .io` on [Github](#), for you to repeat the tests in your own environments as desired.

OCI EKMS Deployment Results

- CipherTrust Manager as a Key Source
- Luna HSM as a Key Source

CipherTrust Manager as a Key Source

Single Node of CipherTrust Manager on Premises, K6 Client on Premises

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on premises

4 CPU 16 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	4-CPU, 16GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	20299	507.43	25
20	21617	540.34	52
30	21849	546.12	79.81
40	21714	542.46	104.91
50	18213	555.26	154.67
60	18507	556.65	182.96

8 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	8-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	32889	822.66	14.55
20	38082	951.89	26.87
30	39502	987.31	40.58
40	40241	1005.79	54.06
50	40667	1016.58	68.00
60	41614	1040.12	80.52
70	41550	1038.59	95.76
80	41552	1036.18	111.07

16 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	16-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	41634	1040.58	11.10
20	55337	1383.14	17.91
30	58867	1471.27	27.95
40	59082	1477.00	39.54
50	59776	1494.32	50.40
60	59661	1491.23	61.90
70	60120	1502.92	73.85
80	60102	1502.30	84.47

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
90	60473	1511.00	95.54
100	59855	1495.95	102.70

32 CPU 64 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	32-CPU, 64GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	44120	1102.86	10.41
20	60385	1509.16	17.77
30	64535	1612.94	27.39
40	66207	1654.89	37.03
50	67923	1697.88	46.33
60	68621	1715.02	56.63
70	69952	1748.73	65.69
80	70635	1765.76	75.25
90	71525	1787.66	83.40
100	71518	1787.92	92.98
110	72039	1800.63	101.30
120	72860	1820.92	109.51

Single Node of CipherTrust Manager on Oracle, K6 Client on Oracle

This deployment includes the following:

- Single node of CipherTrust Manager on Oracle
- K6 client on Oracle

4 CPU 16 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	OCI (India South)	Single Instance	4-CPU, 16GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	27611	690.20	16.96
20	30591	764.37	34.47
30	31314	782.84	52.81
40	31404	785.04	71.65
50	31283	782.02	91.73
55	31488	787.16	102.42
60	31326	783.04	112.23

8 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	OCI (India South)	Single Instance	8-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	48837	1220.88	9.35

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
20	52493	1312.27	19.97
30	52886	1322.13	32.40
40	54573	1364.24	43.91
50	56799	1419.83	51.95
60	56789	1419.67	67.59
70	56789	1451.70	75.79
80	58073	1431.24	90.20
90	57256	1467.96	95.49
95	58567	1464.04	100.91
100	58073	1451.70	109.91

16 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	OCI (India South)	Single Instance	16-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	60413	1510.16	7.2
20	65990	1649.62	17.71
30	66412	1660.12	29.84
40	67571	1689.26	39.53
50	68477	1711.87	49.40
60	70366	1759.05	57.75
70	70760	1768.85	68.54
80	71873	1796.73	77.10
90	72263	1806.35	86.17

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
100	72772	1819.25	94.04
110	73346	1833.49	102.59

32 CPU 64 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	OCI (India South)	Single Instance	32-CPU, 64GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	71518	1787.82	5.98
20	88173	2204.21	12.77
30	86032	2150.61	22.71
40	85805	2145.10	31.78
50	85583	2139.42	40.24
60	85821	2145.43	49.55
70	86075	2151.65	57.82
80	86951	2173.62	64.35
90	87175	2179.26	71.86
100	87318	2182.88	80.24
110	87690	2192.10	88.22
120	88433	2210.77	95.18
130	88332	2208.09	104.37

Two Clustered CipherTrust Manager Nodes on OCI, K6 Client on OCI

This deployment includes the following:

- Two clustered nodes of CipherTrust Manager on OCI

- K6 client on OCI

4 CPU 16 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	OCI (India South)	Two Node Cluster	4-CPU, 16GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	46542	1163.53	9.15
20	55006	1375.10	17.95
30	61006	1525.08	24.97
40	60945	1523.55	35.06
50	61603	1539.96	43.86
60	61142	1528.50	54.07
70	62901	1572.46	61.98
80	55531	1388.17	82.79
90	61666	1541.51	81.94
100	64201	1604.96	89.90
110	63181	1579.36	100.99
120	64013	1600.14	109.66

8 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	OCI (India South)	Two Node Cluster	8-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
20	89998	2249.73	10.48
40	99252	2481.26	21.72
60	101581	2539.43	34.71
80	102691	2565.39	44.81
100	104688	2616.97	57.19
120	107151	2678.72	68.64
140	108174	2704.26	80.55
160	106732	2668.24	95.14
180	107432	2685.62	103.87

16 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	OCI (India South)	Two Node Cluster	16-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
80	136847	3420.89	38.81
100	139770	3494.15	47.48
120	142564	3563.92	56.84
140	143618	3590.36	66.54
160	144336	3608.04	76.48
180	146378	3659.17	84.24

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
200	142915	3572.83	97.54
220	148554	3713.52	109.64
240	148365	3708.96	118.59

32 CPU 64 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	OCI (India South)	Two Node Cluster	32-CPU, 64GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
100	171167	4279.05	40.82
120	170280	4256.91	49.54
140	171716	4292.44	57.35
160	172525	4312.68	65.15
180	174033	4350.56	72.40
200	175320	4382.55	79.39
220	176698	4417.03	87.81
240	176681	4416.70	95.28
260	177756	4443.83	102.95

Single Node of CipherTrust Manager on AWS, K6 Client on OCI

This deployment includes the following:

- Single node of CipherTrust Manager on AWS
- K6 client on OCI

4 CPU 16 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	AWS (us-east-1)	Single Instance	4-CPU, 16GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
2	286	250.95	7.14
4	545	260.74	13.59
6	826	258.63	20.51
8	1097	259.73	27.36
10	1341	264.73	33.49
20	2705	265.30	67.54
30	4065	265.53	101.44
40	5374	263.71	133.69
50	6699	265.47	166.81

8 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	AWS (us-east-1)	Single Instance	8-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
2	272	270.79	6.75
4	517	271.81	12.80
6	799	269.18	19.88
8	1041	273.10	25.95
10	1300	272.14	32.32
20	2630	272.74	65.73
30	3982	270.08	99.42
40	5246	273.78	130.50
50	6533	273.62	162.56

16 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	AWS (us-east-1)	Single Instance	16-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
2	281	258.18	7.02
4	558	258.66	13.88
6	831	261.11	20.71
8	1071	267.11	26.70
10	1367	257.12	34.10
20	2722	260.90	67.72
30	4100	260.75	102.34
40	5477	259.92	136.07
50	6811	260.77	169.62

32 CPU 64 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	AWS (us-east-1)	Single Instance	32-CPU, 64GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	OCI (India South)	8-CPU, 64GB RAM, 50GB HDD

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
2	286	255.92	7.11
4	564	260.81	14.01
6	1099	261.07	27.36
8	829	260.70	20.63
10	1376	261.41	34.31
20	2736	260.08	68.01
30	4093	261.50	102.10
40	5444	260.41	135.01
50	6838	258.46	140.56

Single Node of CipherTrust Manager on Premises, K6 Client on CipherTrust Manager, Zero Network Latency

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on CipherTrust Manager

4 CPU 16 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	4-CPU, 16GB RAM	CipherTrust Manager

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
1	5175	129.35	10.01
2	8783	219.55	11.03
5	13768	344.16	18.30
7	14861	371.50	24.52
10	16229	405.69	33.04
15	16610	415.24	50.43
20	16617	415.41	68.97

8 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	8-CPU, 32GB RAM	CipherTrust Manager

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
2	8827	220.61	10.58
5	17738	443.38	13.66
7	22478	561.93	15.15
10	26744	668.58	18.49
15	30688	767.15	24.85
20	32169	804.29	32.10
25	32814	820.29	40.10

16 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	16-CPU, 32GB RAM	CipherTrust Manager

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
5	23123	577.92	9.38
10	37631	940.72	12.57
15	45749	1143.57	16.07
20	44827	1120.52	22.23
25	48074	1201.65	26.35
30	52257	1306.33	30.88
35	53415	1335.24	35.38

32 CPU 64 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	32-CPU, 64GB RAM	CipherTrust Manager

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
5	12954	323.54	16.46
10	28286	707.02	15.53
15	39067	976.20	17.64
20	45122	1127.82	20.72
25	48757	1218.77	24.31
30	50909	1272.30	28.54
35	51572	1288.99	33.41

Luna HSM as a Key Source

Single Node of CipherTrust Manager on Premises, K6 client on Premises, Single Node of HSM

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on premises
- Single node of HSM on premises

4 CPU 16 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	4-CPU, 16GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Luna Details

Location: SanJose Lab

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	11225	280.50	45.38
20	11775	294.36	93.55
25	12109	302.69	113.76
30	12114	302.82	138.11
40	12272	306.80	183.82
50	12411	310.23	230.62
60	12225	305.57	276.57

8 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	8-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Luna Details

Location: SanJose Lab

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	18436	460.69	25.07
20	22858	571.26	43.26
30	24602	614.82	62.91
40	24966	623.92	84.56
45	25387	634.59	94.60
50	25493	637.13	107.42
60	25642	640.94	129.93
70	25800	644.84	154.20

16 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	16-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Luna Details

Location: SanJose Lab

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	37136	928.17	12.61
20	51297	1282.04	19.17

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
30	53242	1330.81	30.54
40	54518	1362.76	40.99
50	54984	1374.58	52.54
60	55101	1377.31	64.34
70	56116	1402.79	76.03
80	55753	1396.67	87.48
90	55777	1394.21	100.01
100	56125	1402.76	111.71

32 CPU 64 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	16-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Luna Details

Location: SanJose Lab

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	28956	723.62	15.67
20	41891	1046.84	23.14
30	47908	1197.52	31.51
40	50683	1266.91	41.27
50	52991	1324.65	49.40
60	56606	1414.90	59.65
70	57283	1431.93	69.18
80	58169	1454.08	79.86
90	59790	1494.60	87.65

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
100	60458	1510.72	96.90
110	61227	1530.05	105.71

Single Node of CipherTrust Manager on Premises, K6 client on Premises, HSM in HA Mode

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on premises
- HSM in HA mode

4 CPU 16 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	4-CPU, 16GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Luna Details

Location: SanJose Lab

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	11234	280.83	45.37
20	11816	295.35	92.86
25	12102	302.52	113.20
30	12161	303.93	136.79
40	12164	304.04	184.84
50	12189	304.666	232.04
60	12043	301.03	284.89

8 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	8-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Luna Details

Location: SanJose Lab

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	17985	449.59	25.72
20	23564	588.82	41.57
30	24963	623.88	61.80
40	25467	636.57	83.33
50	26518	662.90	102.92
60	26507	662.56	126.52
70	26426	660.48	149.56

16 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	16-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Luna Details

Location: SanJose Lab

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	23799	594.90	18.53
20	36102	902.26	25.99
30	41629	1040.41	34.69
40	43610	1090.23	43.97
50	45497	1137.72	55.17
60	45341	1132.72	68.07
70	42385	1059.05	85.37
80	43175	1079.02	97.42
90	43658	1091.35	111.28
95	42935	1073.10	101.68

32 CPU 64 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	16-CPU, 32GB RAM	CipherTrust Manager

Client Details

OS Type	Location	Client Configuration
Ubuntu-22.04	SanJose Lab	8-CPU, 64GB RAM, 50GB HDD

Luna Details

Location: SanJose Lab

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
10	27354	683.48	15.83
20	45355	1133.54	20.56
30	49844	1245.68	29.37
40	52258	1305.86	39.04
50	55543	1387.86	47.99

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
60	57365	1433.90	57.45
70	58480	1461.91	66.59
80	60001	1499.55	75.99
90	60492	1511.70	85.38
100	61408	1534.53	95.36
110	60635	1515.60	105.31

Single Node of CipherTrust Manager on Premises, K6 Client on the CipherTrust Manager, Zero Network Latency

This deployment includes the following:

- Single node of CipherTrust Manager on premises
- K6 client on CipherTrust Manager

4 CPU 16 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	4-CPU, 16GB RAM	CipherTrust Manager

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
2	5432	135.78	17.8
5	8978	224.43	26.76
6	9752	243.74	30.01
7	10245	256.06	33.97
10	10952	273.78	46.84
15	11419	285.41	69.21
20	11403	285.07	95.21

8 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	8-CPU, 32GB RAM	CipherTrust Manager

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
4	10465	261.57	17.54
8	16066	401.46	23.60
12	20510	512.60	30.79
15	21416	535.34	34.05
20	22573	564.32	44.22
25	24195	604.76	52.81

16 CPU 32 GB

Here are the details of this deployment:

CM Details

CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	16-CPU, 32GB RAM	CipherTrust Manager

Results

Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
5	23583	589.55	9.63
10	33399	834.82	14.20
15	40881	1021.89	18.05
20	44951	1123.66	22.20
25	47429	1185.56	26.64
30	49079	1226.71	31.59
35	49384	1234.37	37.03

32 CPU 64 GB

Here are the details of this deployment:

CM Details

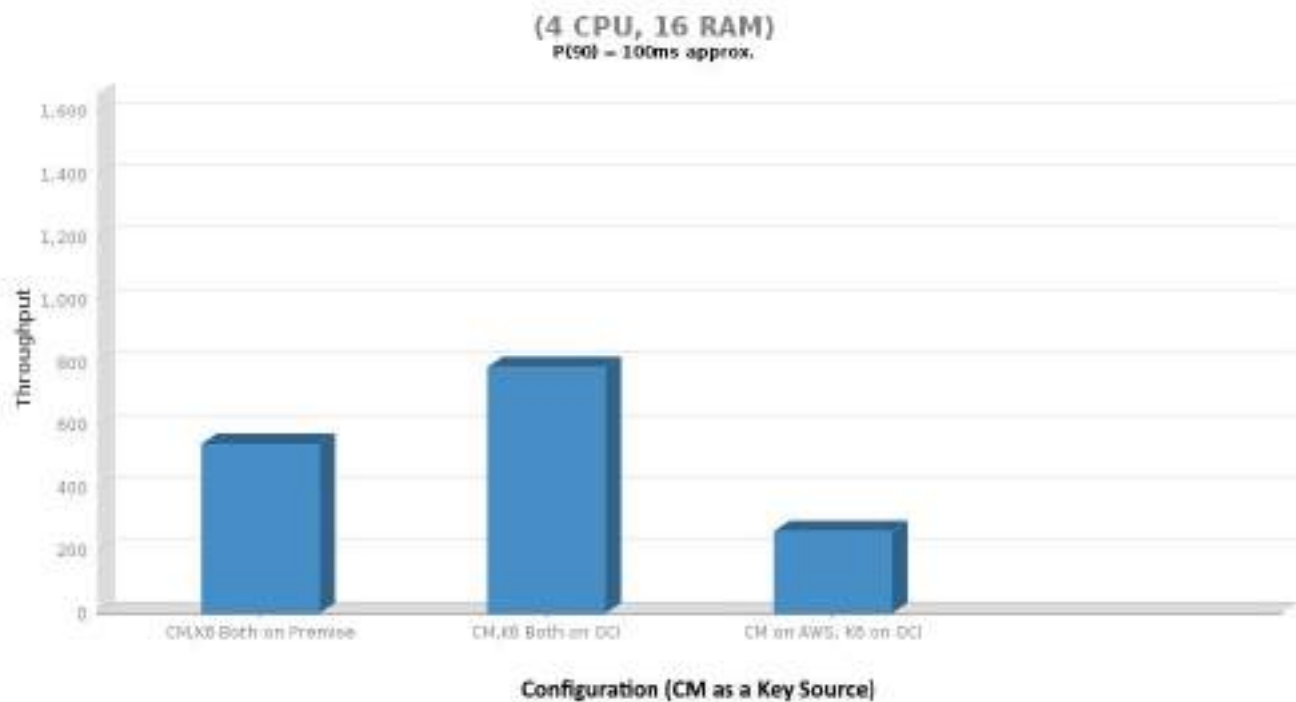
CM Version	CM Location	Instance Type	CM Configuration	Source
2.15.0.10938 RC1	SanJose Lab	Single Instance	32-CPU, 64GB RAM	CipherTrust Manager

Results

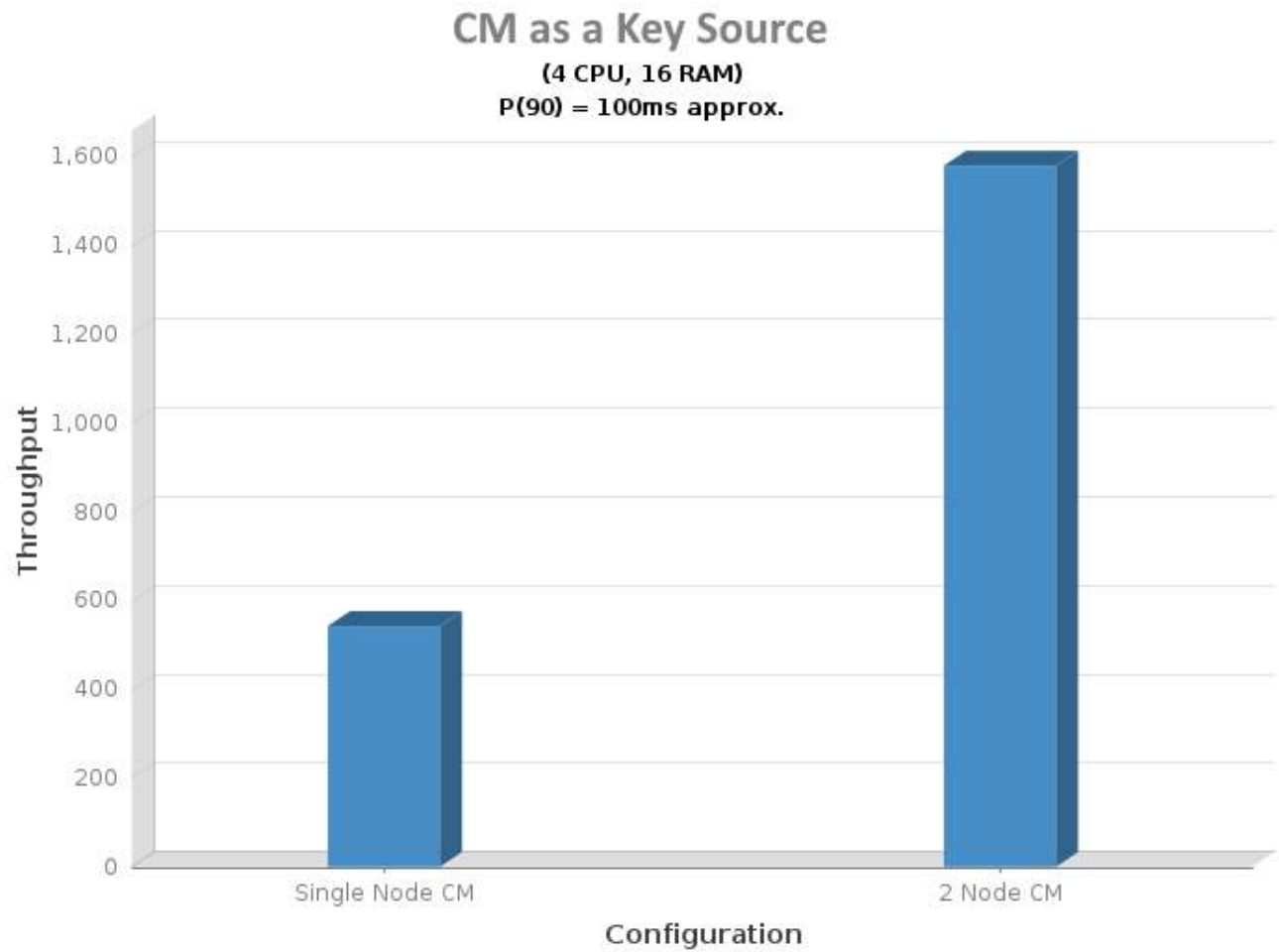
Number of Virtual Users	Total Operations	Throughput (Ops/Sec)	90% Time
5	12652	316.14	16.86
10	27958	698.76	15.79
15	38461	961.34	17.64
20	44388	1109.34	21.07
25	47964	1198.71	24.80
30	50565	1263.22	28.66
35	51783	1295.30	33.71

Conclusion

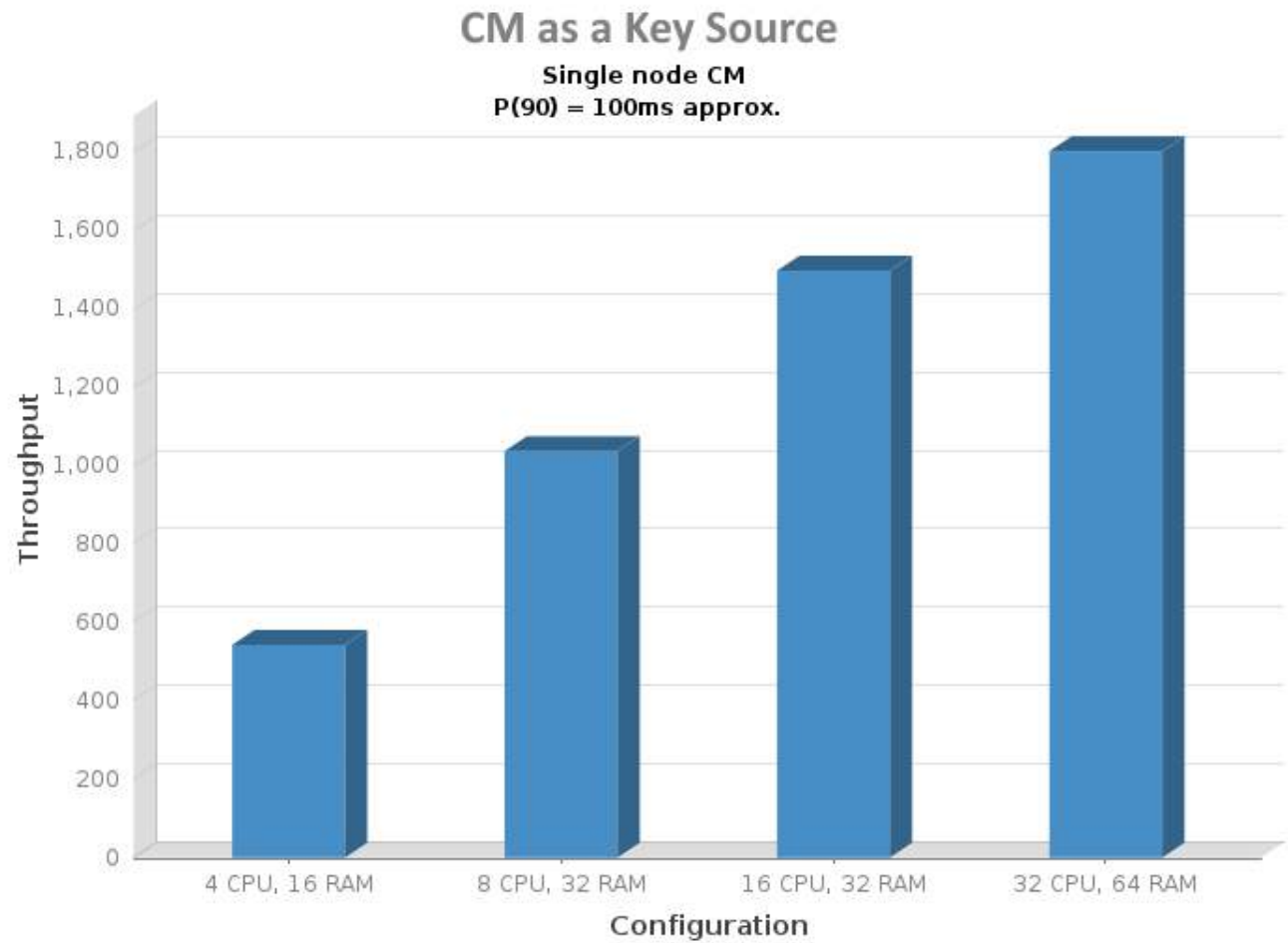
- The network latency between the K6 client which is simulating traffic from Oracle KMS to CipherTrust Manager affects the performance. In the environments, where the K6 client and the CM are closer geographically or are in the same network, the performance is higher.



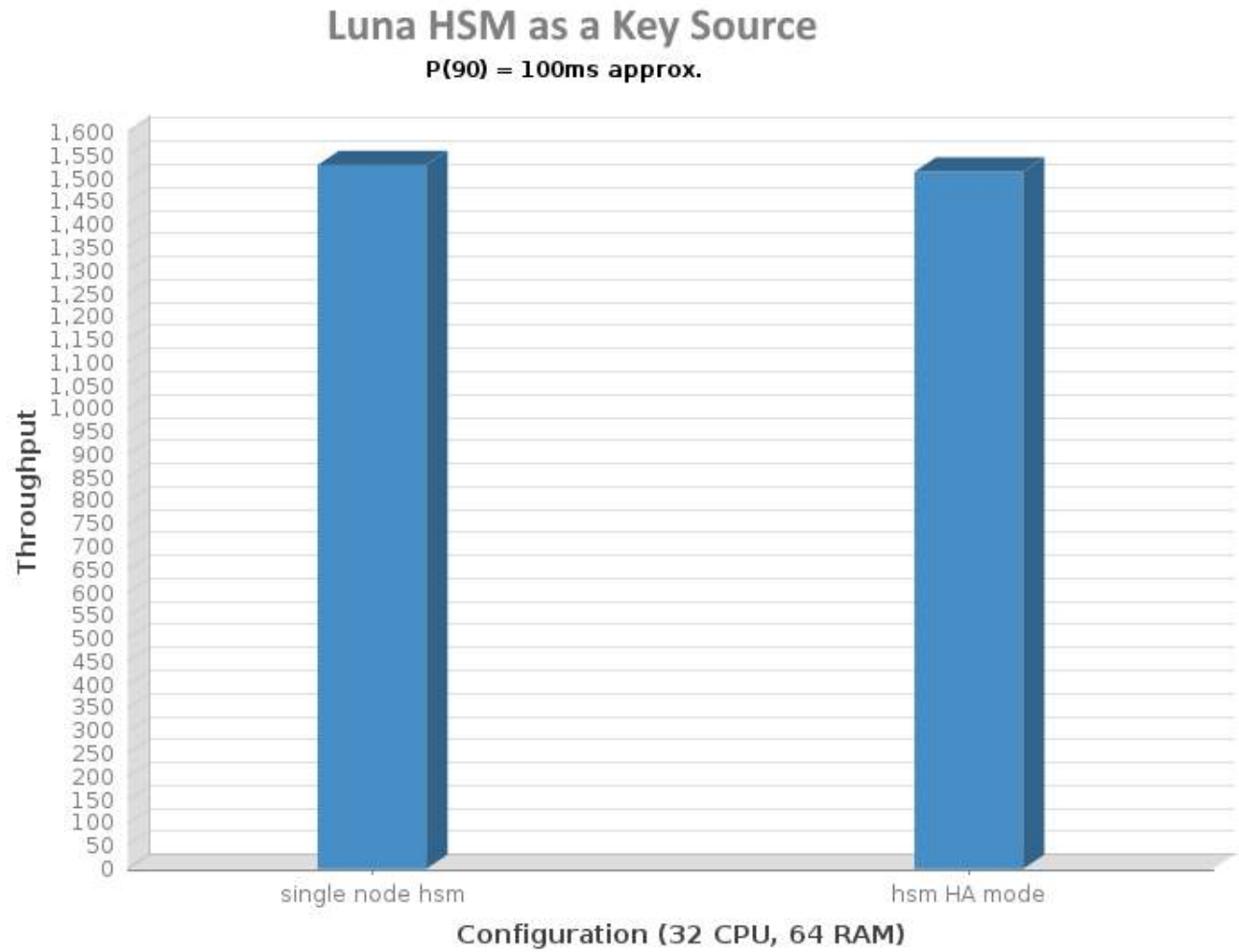
- Increasing the number of nodes enhances the performance.



- Increasing CM RAM and CPU increases the performance.



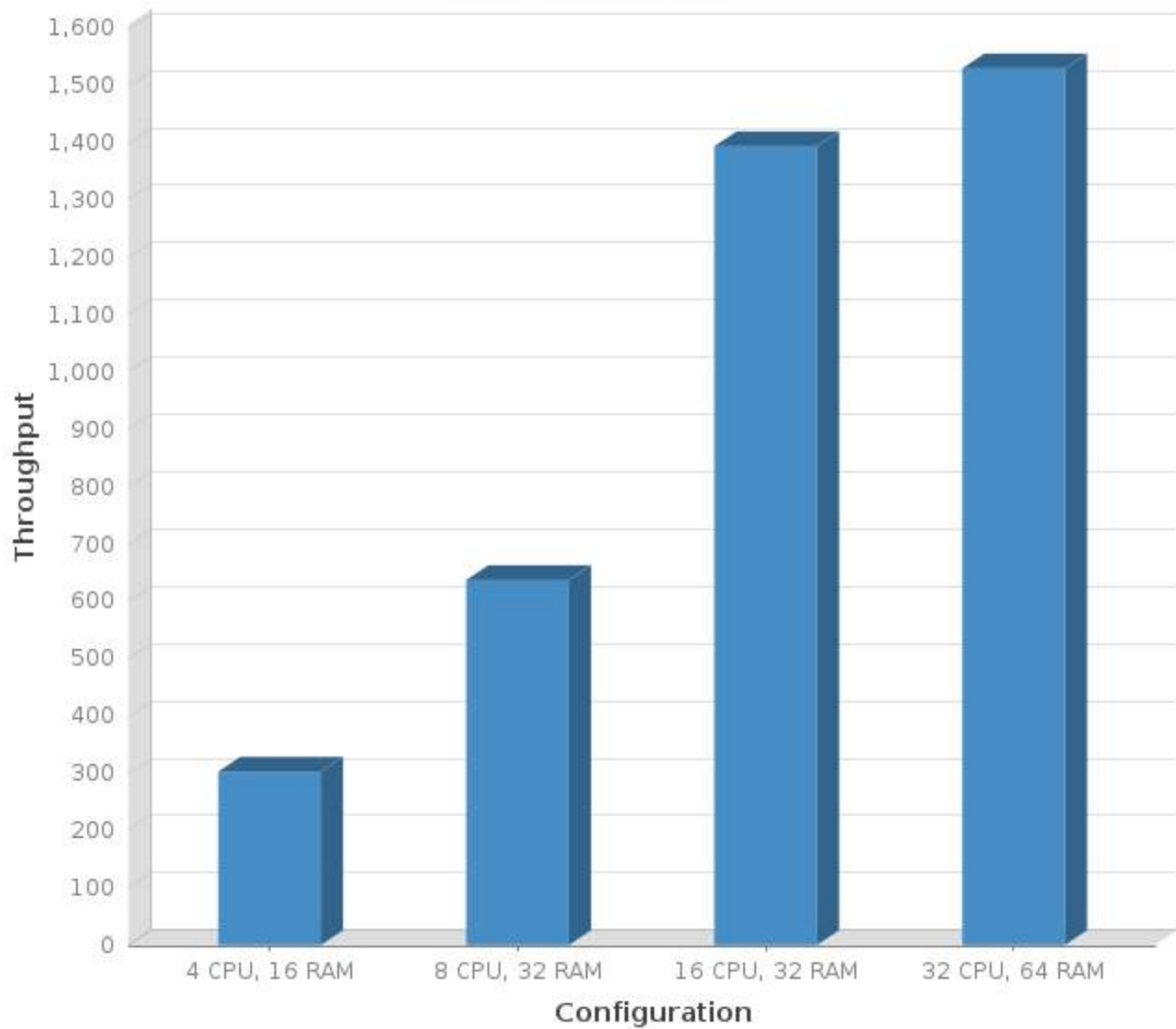
- The number of HSM nodes in HA doesn't affect the throughput.



- In the below two graphs, increasing the CM RAM and CPU improves the performance.

Luna HSM as a Key Source

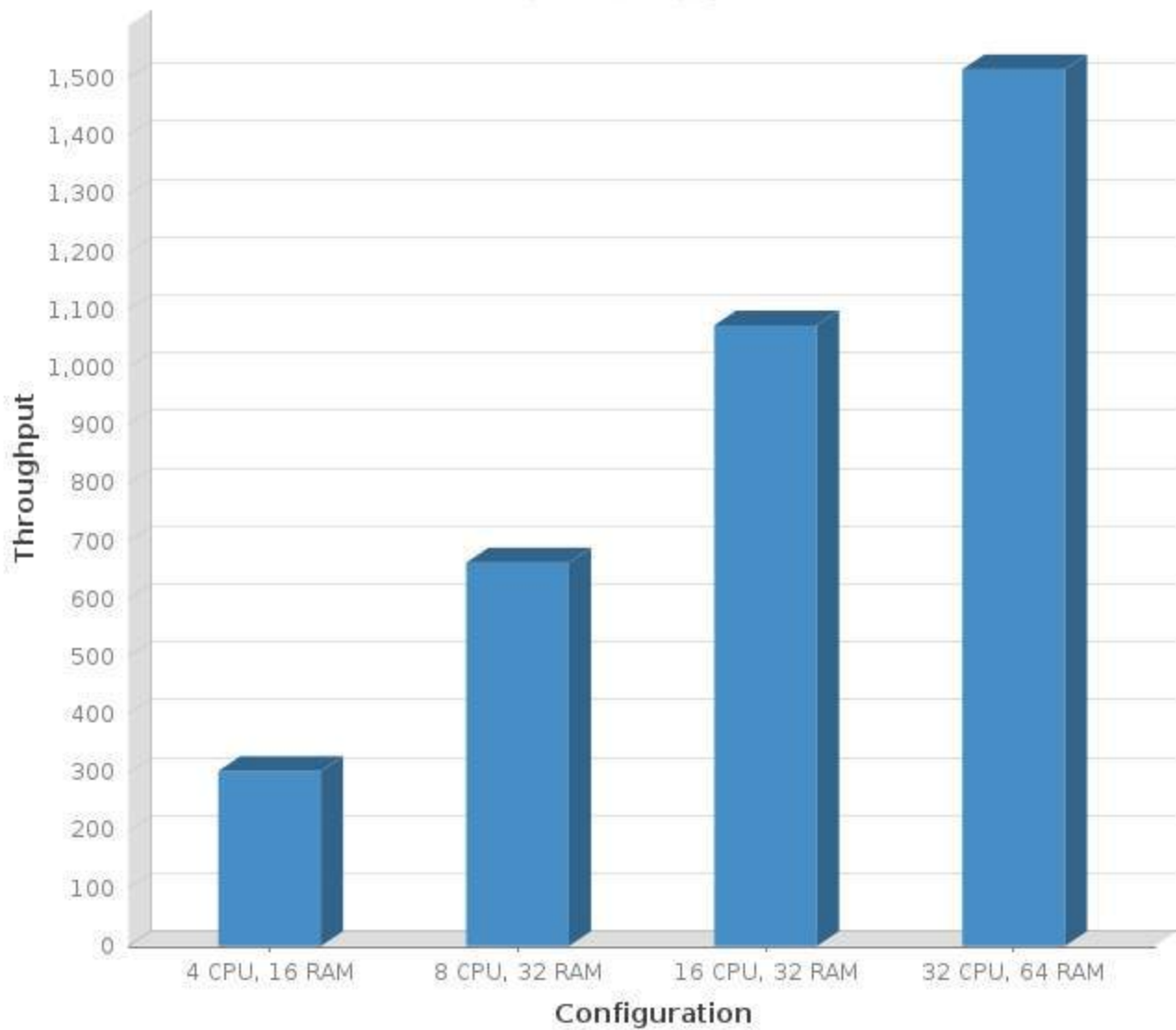
Single Node of HSM
P(90) = 100ms approx.



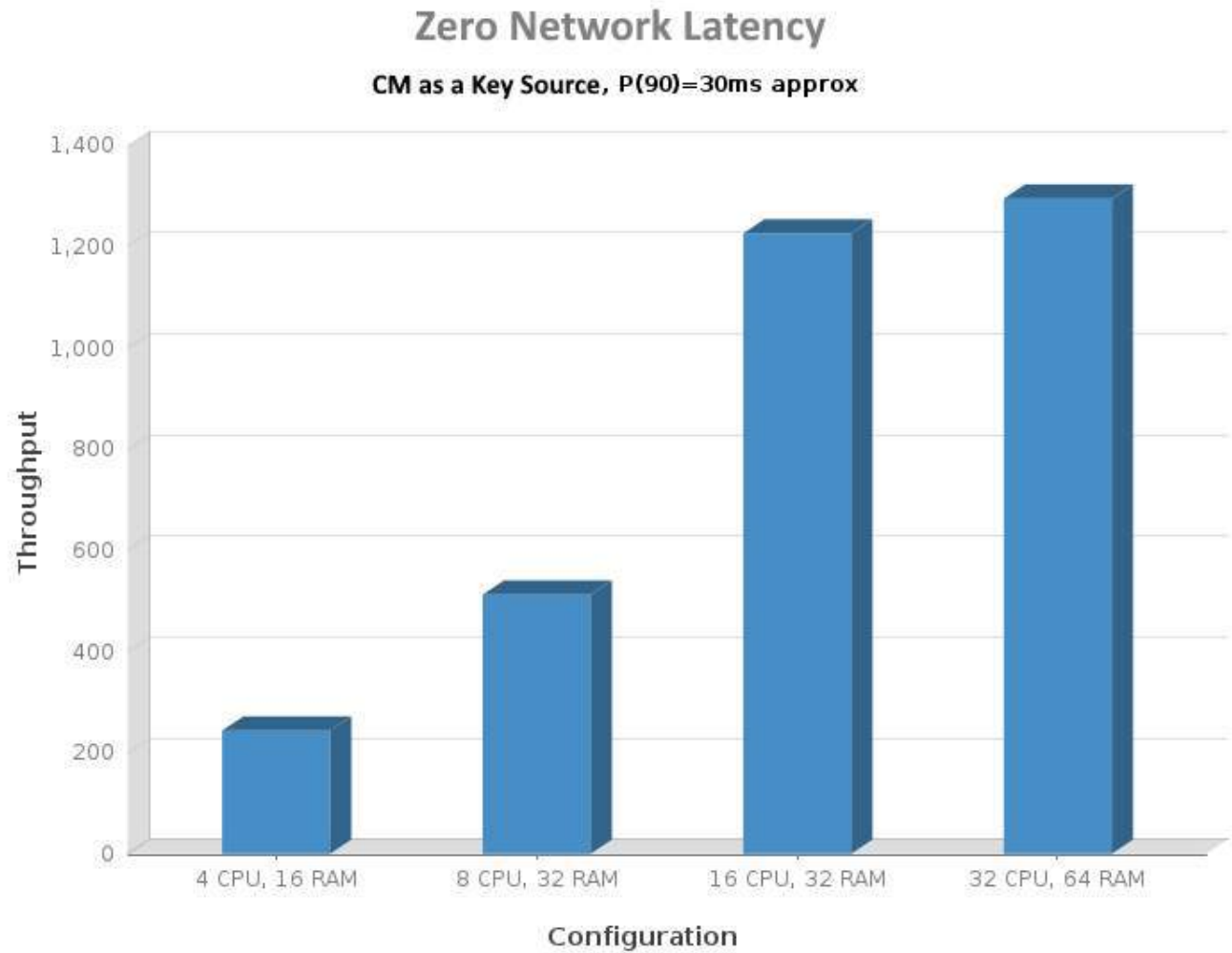
Luna HSM as a Key Source

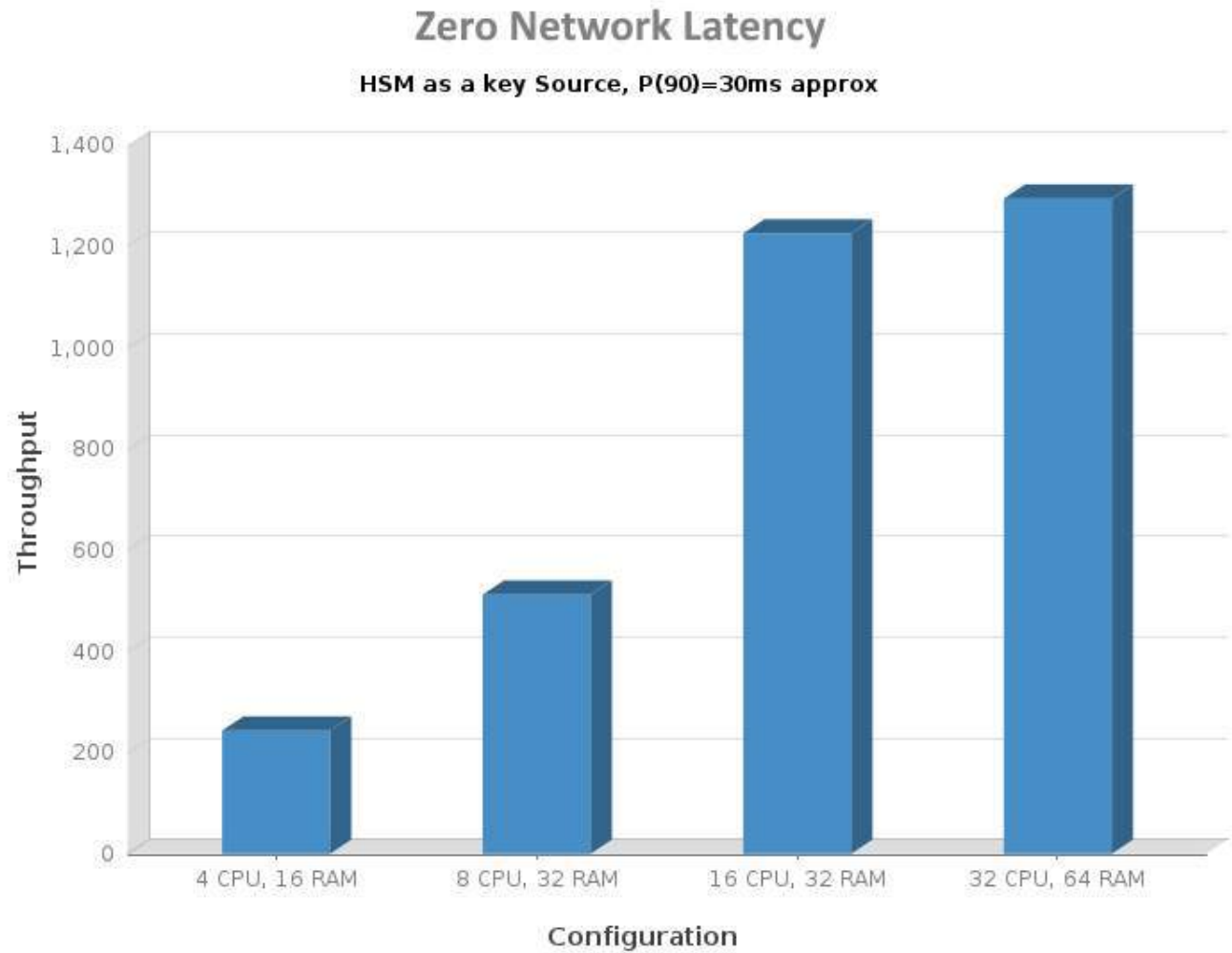
HSM in HA Mode

P(90) = 100ms approx.

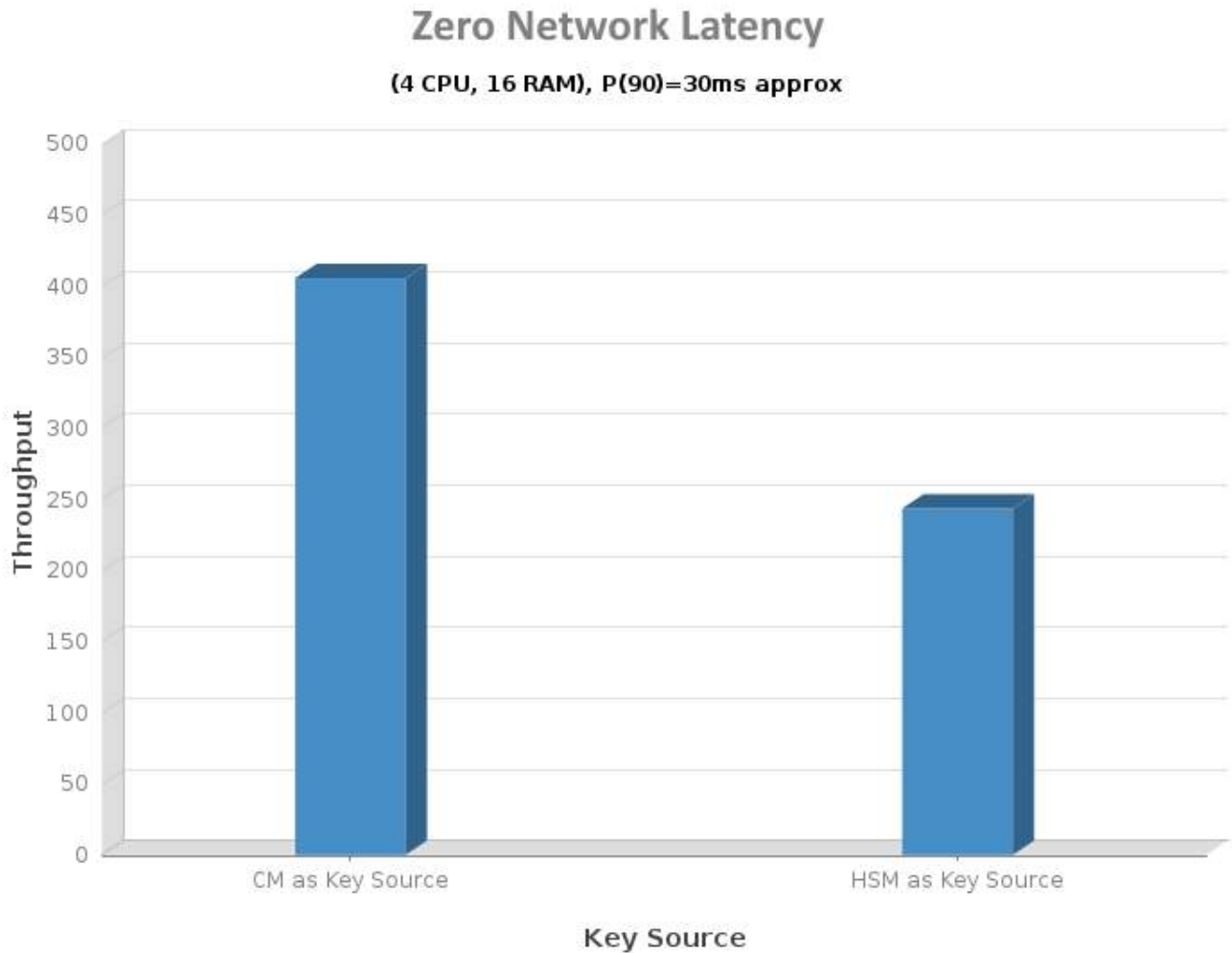


- In the below two graphs, increasing the CM RAM and CPU improves the performance. We have considered the threshold 30ms, where the K6 client is running on CM.





- The performance is better when the Key Source is CM as compared to the Luna HSM, where the K6 client is running on CM.



SAP Resources

CCKM supports the SAP Data Custodian cloud.

Prerequisites

Before you can manage SAP resources on CCKM:

1. [Get a Service Admin User Account](#)
2. [Create a New Group](#)
3. [Create an Application Technical User or a Standard User in ISM](#)
4. [Add the Standard User to Key Management Service](#)
5. [Add SAP Connection on CipherTrust Manager](#)

Click each link to view details.

Get a Service Admin User Account

Get a Service Admin user account from SAP. You will be provided service account credentials (a Data Custodian URL, tenant, and username.) These are required to access the Data Custodian system.

To log on to the SAP cloud as the Service Admin:

1. Open the URL for your SAP Data Custodian environment.
2. Enter your tenant name.
3. Click **Continue**.
4. Enter your **Username** and **Password**.
5. Click **Login**.

Next, you need to create a new group.

Create a New Group

Key Management Service groups are used to organize encryption keys and KMS users. Every group is associated with a specific application type, key store provider, key store type, and service tier.

A user with the Key Administrator role in KMS can create new groups.

To create a new group:

1. Log on to the SAP cloud as the Key Admin user.
2. Navigate to **Home > Dashboard**. The Key Management Service Dashboard is displayed.
3. On the **Key Management Service** tab, select an application, for example **General/IAAS Applications**.
4. Click **Create Group**. The Group Details screen is displayed.
5. Specify the group details:
 - a. Specify the **Group Name**.
 - b. Add the **Group Description**.
 - c. (Optional) If applicable, provide the **Landscape** type, **System ID**, and **Tenant**. These fields are optional.
6. Click **Step 2**. The Key Store Selection screen is displayed.
7. Specify the key store details:
 - a. Select **SAP Data Custodian Provided Key Store**.

- b. Select a **Key Store** (for example, AWS, ESK, Azure).
- c. Select the **Key Store Region** where your key store is located.
- d. Click **Review** to verify the group details. Click **Edit** to make any changes.
- e. Click **Create**.

The group is created and its details are displayed. After you have created the group, you need to add a new user to Data Custodian's Identity Service Management (ISM). New Data Custodian users must be added to ISM before they can be added to the Key Management Service.

Create an Application Technical User or a Standard User in ISM

Click the desired tab to view the instructions.

Note

- It is recommended to use Technical Users when creating new connections for SAP Data Custodian on the CipherTrust Manager.
- If you don't use Technical Users when the SAP Data Custodian KMS is integrated with the SAP Cloud Identity Services, the connection to SAP may not work.

User

Add a user to Data Custodian's Identity Service Management (ISM). SAP Data Custodian users with an administrator role in ISM can create new users and edit their service roles.

To add a user in ISM:

1. Log on to the SAP cloud as the Service Admin.
2. Go to top left **Menu > Tenant Management > Users**.
3. Click **Create**.
4. Enter the user's **First Name** and **Last Name**.
5. Enter the user's **Email Address**.
6. Select optional ISM administrator **Roles** to assign to the user.
7. Select an optional **Active Time Range** for the user.
 - a. **Active From**: Allows you to activate the user at a specific time.
 - b. **Active To**: Allows you to deactivate the user at a specific time.

-
8. Click **Review**. Verify the user details. Click **Edit** to make changes, if required.
 9. Click **Create**.

The user is created. Next, you need to add this user to the Key Management Service, as described below.

Technical User

Add an application technical user to Data Custodian's Identity Service Management (ISM). SAP Data Custodian users with a key administrator (Key Admin) role in ISM can create new technical users and edit their permissions.

For more details, refer to the SAP documentation on [Creating an Application Technical User](#).

To add a technical user in ISM:

1. Log on to the SAP cloud as the Key Admin.
2. Go to top left **Menu > Home > Dashboard**.
3. Click the desired application.
4. Click the desired group.
5. On the **Users** tab, click **Technical Users**.
6. Click **Create Technical User**.
7. Enter the **Username** and **Description** (optional), and click **Step 2**.
8. Select an **Active Time Range** and click **Step 3**.

To specify the range:

- a. **Active From**: Allows you to activate the technical user at a specific time. The technical user remains inactive until the **Active From** time is specified.
 - b. **Active To**: Allows you to deactivate the technical user at a specific time.
9. Select the following permissions.
 - View all audit logs for this technical user's group
 - Create Keys
 - Modify Keys
 - Delete Keys
 - Backup and restore keys
 - Manage Labels
 - Read all keys in group
-

- Crypto
- Rotate
- Export all keys in group
- Service Availability Check
- Read Dynamic Key References

10. Click **Review**.
11. Click **Create**.
12. Click **Generate Credential**.
13. Enter **Credential Name**, select **Expiry Date** (optional).
14. Ensure that **Activate Now** is set to **NO**.

Warning

If **Activate Now** is set to **YES**, a connection using this Technical User's credentials cannot be created. You will keep getting the error, "Credential must only be refreshed once".

15. Click **Download**.

The credentials (access key, secret key, and API endpoints) are downloaded in a ZIP file named **<credential-name>.zip**. These credentials are needed to create a SAP connection on the CipherTrust Manager. It is recommended to use the latest API endpoint URL listed in the extracted **API endpoints.txt** file. Refer to [Add SAP Connection on CipherTrust Manager](#) for details.

Note

Connection Test Behavior

When you use a Technical User's credentials to create the SAP connection on CipherTrust Manager:

1. The **Test Connection** during the creation process will fail. This is expected as the credential is not yet active.
2. Proceed to save the connection without a successful test.
3. After the connection is created, find it in the Connections list and perform a **Test Connection** on it. This second test will succeed and activate the Technical User credential.

Standard User

Add a standard user to Data Custodian's Identity Service Management (ISM). SAP Data Custodian users with an administrator role in ISM can create new users and edit their service roles.

To add a user in ISM:

1. Log on to the SAP cloud as the Service Admin.
2. Go to top left **Menu > Tenant Management > Users**.
3. Click **Create**.
4. Enter the user's **First Name** and **Last Name**.
5. Enter the user's **Email Address**.
6. Select optional ISM administrator **Roles** to assign to the user.
7. Select an optional **Active Time Range** for the user.
 - a. **Active From**: Allows you to activate the user at a specific time.
 - b. **Active To**: Allows you to deactivate the user at a specific time.
8. Click **Review**. Verify the user details. Click **Edit** to make changes, if required.
9. Click **Create**.

The user is created. Next, you need to add this user to the Key Management Service, as described below.

Add the Standard User to Key Management Service

Note

This section is applicable to standard (non-technical) SAP users. Skip the section when adding SAP Technical Users.

Add the user created in the previous step to the Key Management Service. Only the Key Management Service users with an Administrator role can add other users to KMS.

To add the user to the Key Administrator service:

1. Navigate to **Tenant Management > Users**.
2. From the **All Users** section, select the user (created [above](#)).
3. Click **Add to Service**. You might need to open the menu on the top-right, to see this button.

4. Select **Key Management Service (KMS)**.
5. Select the **Key Administrator** role to assign to the user. The user with the Key Administrator role is referred to as the Key Admin user in this document.

Note

Optionally, assign one or more existing groups to the user. Refer to [User Roles in the Key Management Service](#) in SAP documentation for more information.

6. Click **Add**.

After assigning the Key Administrator role to the user, add SAP connection on CipherTrust Manager.

Add SAP Connection on CipherTrust Manager

Before you can add a SAP group to the CCKM, a connection to your SAP account must already exist on the CipherTrust Manager. After you have created a SAP group, add a connection to the SAP cloud using your SAP tenant and the Key Admin username and password.

A CipherTrust Manager administrator manages connections to external resources on the **Access Management > Connections Management** page of the CipherTrust Manager GUI. Refer to [Connection Manager](#) for details.

What Next?

After completing the prerequisites, you can view linked SAP groups and manage keys on the CipherTrust Manager.

Refer to the following sections:

- [Managing SAP Groups](#)
- [Managing SAP Keys](#)
- [Scheduling Operations](#)
- [Managing SAP Reports](#)
- [Viewing Linked SAP Applications](#)
- [Downloading SAP Metadata](#)

Managing SAP Groups

This section describes how to manage SAP groups on CCKM. Before proceeding, a connection to your SAP account must exist on the CipherTrust Manager. Refer to [Connection Manager](#) for details.

After the connection is configured, you can add groups to the CipherTrust Manager. SAP groups can be added,

viewed, modified, or deleted on the SAP Data Custodian Groups page.

Adding Existing SAP Groups

You can add existing groups linked to a SAP connection to the CipherTrust Manager. An existing group can only be added just once.

To add an existing SAP group:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > SAP Groups**. The Groups tab of the SAP Data Custodian Groups page is displayed.
3. Click **Add Existing Groups**. The Select Connection tab of the Add SAP Data Custodian Group screen is displayed.
4. Select the desired **Connection** from the list.
5. Click **Next**. The Add SAP Data Custodian Group tab is displayed. You will see the list of existing groups linked with the selected connection.
6. Under **Group Name**, select the desired groups.
7. Click **Save**.

The selected group is displayed on the SAP Data Custodian Groups page. Now, you can manage the group from CCKM on the CipherTrust Manager.

The group is available to [upload SAP keys](#) and [view SAP reports](#).

Viewing SAP Data Custodian Groups

The Groups tab of the SAP Data Custodian Groups page shows the list of groups added to the CipherTrust Manager. Search for the groups by Group Name, Created By, or Region.

To view the list of groups added to the CipherTrust Manager:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > SAP Groups**. The Groups tab of the SAP Data Custodian Groups page shows the list of groups added to the CipherTrust Manager.

The page displays the following details:

Column	Description
Group Name	Name of the group.
Keystore Provider	Name of the keystore provider. The provider can be: • ESK

Column	Description
	• AWS • AZURE • THALES
Keystore	Name of the keystore.
Key Management Service	Name of the key management service. The service can be: • Software KMS • Software HYOK
Region	Region of the group.
Application	Name of the SAP application.
Connection	Name of the SAP connection added to the CipherTrust Manager.
Created By	Name of the SAP user who created the group.
Tenant	Name of the SAP tenant. For SAP Technical Users (TUs), the field remains blank.
Last Refreshed	Date and time when the group was refreshed the last.

To view/hide columns, click the Customize View (⚙️) icon, select/clear the desired option, and click **OK** to display the column.

Refreshing SAP Groups

Refreshing is the process to download keys created in SAP groups to the CCKM. You can refresh keys from individual or all SAP groups.

- Refreshing Specific SAP Groups
- Refreshing All SAP Groups

Refreshing Specific SAP Groups

To refresh a group:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > SAP Groups**.
3. On the Groups tab, click the overflow icon (⋮) corresponding to the desired group and click **Refresh Now**.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > SAP > SAP Keys** page. Refer to [Viewing SAP Keys](#) for details.

Refreshing All SAP Groups

To refresh all SAP groups:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > SAP Groups**.
3. On the Groups tab, click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all groups is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > SAP > SAP Keys** page. Refer to [Viewing SAP Keys](#) for details.

Viewing Details of a Group

To view the details of a group on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > SAP Groups**.
3. On the Groups tab, click the **Group Name** link of the desired group.

Alternatively, click the overflow icon () corresponding to the desired group, and click **View/Edit Details**.

The edit view of the SAP Data Custodian Groups page shows additional details of the selected group under the ACCESS CONTROL and GENERAL INFO sections. Expand each section to view more details.

Changing the SAP Connection

To change the SAP connection of a group:

1. Expand **GENERAL INFO**.
2. From the **Connection ID** drop-down list, select the SAP connection.

3. Click **Update**.

The connection of the SAP group is changed.

Managing User Permissions on SAP Groups

To work with the SAP cloud, users/groups must have the minimum set of permissions that allow them to use the SAP resources such as keys and groups. Initially, the CCKM user only has permission to view the keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform operations on SAP groups.

Users with the following characteristics can perform operations for SAP keys and SAP groups:

- Users in the **CCKM Admins** group
- Users in the **Admin** group
- Users who are administrators for a **domain**
- Users who are in the **CCKM Users** group and which have had a CCKM Admin assign permissions through the UI or the `/v1/cckm/sap/groups/{id}/update-acls` endpoint in the REST API.

Adding Permissions for a User/Group

To add permissions for a user/group:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > SAP Groups**.
3. On the Groups tab, click the **Group Name** link of the desired group.

Alternatively, click the overflow icon () corresponding to the desired group, and click **View/Edit**.

4. Expand the **ACCESS CONTROL** section.
5. Click **Assign User/Group**. The **Assign User/Group** dialog box is displayed.
6. Select the desired user or group from the **User/Group** drop-down list.
7. Click **Save**.

The newly added user/group is displayed under **Name** in the **ACCESS CONTROL** section. You can now grant additional permissions to the user/group, as appropriate. Refer to [Granting Permission to Perform an Operation](#) for details.

Allowed Operations

CCKM allows the following operations on SAP groups:

- View Keys, Add Native Key, Add BYOK Key, Edit Key
- Rotate Key, Delete Key, Delete Key Backup, Remove Key, Restore Key
- Rotate to Native Key, Rotate to BYOK Key, Synchronize Key

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. In the ACCESS CONTROL section, select the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

To revoke permissions from a user/group, refer to [Removing a Permission](#) for details.

Removing a Permission

To remove a permission assigned to a user or group:

1. In the ACCESS CONTROL section, clear the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. In the ACCESS CONTROL section, under **Unassign**, click the **X** button corresponding to the desired user/group.
2. On the **Remove User / Remove Group** screen, click **Remove**.

Note

Removing this user/group will remove all permissions currently assigned to the user/group.

3. Click **Remove** to confirm the action. To cancel the action, click **Keep It**.

A success message is displayed on the screen.

Removing SAP Groups

SAP groups can be removed on the SAP Data Custodian Groups page. Search for existing groups using Group Name or Created By.

To remove a group from CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > SAP Groups**.
3. On the Groups tab, click the overflow icon () corresponding to the group you want to remove.
4. Click **Delete**.
5. Select **I wish to delete the SAP Group**.
6. Click **Delete**.

The SAP group is deleted successfully. The group is removed from the list of SAP Data Custodian groups.

Managing SAP Keys

This section describes how to manage SAP keys on CCKM. Before proceeding, you must have a SAP Data Custodian group added to the CCKM. Refer to [Managing SAP Groups](#) for details.

- [Key Creation Methods and Sources](#)
- [Creating/Uploading New Key Material](#)
- [Cloning Existing Key Material](#)
- [Viewing SAP Keys](#)
- [Refreshing SAP Keys](#)
- [Enabling a SAP Key](#)
- [Disabling a SAP Key](#)
- [Deleting a SAP Key](#)
- [Removing a SAP Key](#)
- [Adding a Key Version](#)
- [Viewing or Editing Details of SAP Keys](#)
- [Changing Key Details](#)

-
- Changing Key Exportability
 - Changing Key Attributes
 - Adding or Changing Key Rotation Schedule
 - Disabling Key Rotation Schedule
 - Viewing Key Version Details

Key Creation Methods and Sources

Methods to create SAP cloud keys using CCKM are:

- **Creating/Uploading New Key Material:** Add key material by creating and uploading new source key or creating new native key. The key source can be:
 - **CipherTrust (External):** A new key is first created on the external CipherTrust Manager. Then, this key material is uploaded to SAP cloud to create a new SAP key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CipherTrust (External)**.
 - **CipherTrust (Local):** A new key is first created on the CipherTrust Manager. Then, this key material is uploaded to SAP cloud to create a new SAP key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.
 - **SAP (Native):** A new key is directly created on SAP cloud using a native SAP application. The key origin is **NATIVE**.
 - **Vormetric DSM:** A new DSM key is first created on the CipherTrust Manager. Then, this key material is uploaded to SAP cloud to create a new SAP key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.
 - **Luna HSM:** A new Luna HSM key is first created on the CipherTrust Manager. Then, this key material is uploaded to SAP cloud to create a new SAP key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

- **Cloning Existing Key Material:** Clone key material from an existing key to create a new key. The key source can be:
 - **CipherTrust (Local):** An existing local CipherTrust Manager key is first cloned on the CipherTrust Manager.

Then, the cloned key material is uploaded to SAP cloud to create a new SAP key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.

- **CipherTrust (External):** An existing key is first cloned on the external CipherTrust Manager. Then, this key material is uploaded to SAP cloud to create a new SAP key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CipherTrust (External)**.
- **Vormetric DSM:** An existing DSM key is first cloned on the CipherTrust Manager. Then, the key material is uploaded to SAP cloud to create a new SAP key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.
- **Luna HSM:** An existing Luna HSM key is first cloned on the CipherTrust Manager. Then, the key material is uploaded to SAP cloud to create a new SAP key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Supported Key Types and Sizes for Key Stores

The following key types and sizes/curves are supported for AWS keystore.

Size/Curve	Native	BYOK
AES	256	256, applicable to all the key materials.
RSA	2048, 3072, and 4096	2048, 3072, and 4096. Applicable to all the key materials.
EC	ECC_NIST_P256 (SECP256R1, PRIME256V1), ECC_NIST_P384 (SECP384R1), ECC_NIST_P521 (SECP521R1), and ECC_SECG_P256K1 (SECP256K1). Applicable to asymmetric keys only.	ECC_NIST_P256 (SECP256R1, PRIME256V1), ECC_NIST_P384 (SECP384R1), ECC_NIST_P521 (SECP521R1), and ECC_SECG_P256K1 (SECP256K1). Applicable to CipherTrust (External) and CipherTrust (Local) key materials.

If a SAP group has an AWS keystore, you cannot export the key.

The following key types and sizes/curves are supported for the ESK keystore.

Size/Curve	Native	BYOK
AES	128, 192, and 256	128, 192, and 256. 192 is not applicable to DSM key material.

Size/Curve	Native	BYOK
RSA	2048, 3072, 4096, and 8192	2048, 3072, and 4096. For Luna HSM, you can also select 8192.
EC	SECP192K1, SECP224K1, SECP256K1, NISTP192, NISTP224, NISTP256, NISTP384, and NISTP521	ECC_SECG_P224K1 (SECP224K1), ECC_SECG_P256K1 (SECP256K1), ECC_NIST_P224 (SECP224R1), ECC_NIST_P256 (PRIME256V1), ECC_NIST_P384 (SECP384R1), and ECC_NIST_P521 (SECP521R1). Applicable to CipherTrust (External) and CipherTrust (Local) key materials.

Creating/Uploading New Key Material

To add a SAP cloud key by creating/uploading new key material:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**.
3. Click **Add Key**. The Select Material Origin screen of the Add SAP Key wizard is displayed.
4. Under **Select Method**, select **Create/Upload New Key Material**. The **Select Source** section appears. Depending on your requirements, select from the following:
 - **CipherTrust (External)**: Refer to [Uploading CipherTrust \(External\) Key Material](#) for details.
 - **CipherTrust (Local)**: Refer to [Uploading CipherTrust \(Local\) Key Material](#) for details.
 - **SAP (Native)**: Refer to [Creating SAP \(Native\) Key Material](#) for details.
 - **Luna HSM**: Refer to [Uploading Luna HSM Key Material](#) for details.
 - **Vormetric DSM**: Refer to [Uploading Vormetric DSM Key Material](#) for details.

Refer to [Key Creation Methods and Sources](#) for details on key sources.

Refer to [Supported Key Types and Sizes for Key Stores](#) for key store support information.

Uploading CipherTrust (External) Key Material

Upload the key material of an external CipherTrust Manager key using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

1. Select **CipherTrust (External)**.

2. Click **Next**. The Select Group screen is displayed.

Select Group

1. Select a **Group** from the list.

Note

If you are creating a master/primary key, you must select a group that allows you to create a master/primary key. The following groups are supported.

- S4/HANA
- Integrated Business Planning
- SAP Commissions
- HANA Enterprise Cloud
- SAP Marketing Cloud
- SuccessFactors
- Analytics Cloud

2. Click **Next**. The Configure CipherTrust (External) Key screen is displayed.

Configure CipherTrust (External) Key

1. **Select Domain** from the drop-down list. The drop-down list shows the external CM domains linked with the configured external CM connection.
2. Enter a **Key Name**. A new key with this name will be created on the CipherTrust Manager and its key material will be uploaded to SAP cloud.
3. Select the **Algorithm** for the key. The options are:
 - **AES**: Creates and uploads an AES key.
 - **RSA**: Creates and uploads an RSA key pair.
 - **EC**: Creates and uploads an EC key.
4. Select the **Key Size / Elliptic Curve** based on the key type.

Note

- If you are creating a master/primary key, you must select RSA as the key type, and 3072 or 4096 as the key size.
- For AES as the algorithm, you do not need to select a key size.

5. Click **Next**. The Configure SAP Key screen is displayed.

Configure SAP Key

1. (Optional) Provide a basic **Description** for the key.
2. (To create a master/primary key) Select **Create as Master/Primary key**.

Note

You can create only one master/primary key for a group. If a master/primary key already exists for the selected group, you can't select **Create as Master/Primary key**.

3. (Optional) Select **Allow Key Export**. Selecting this check box allows the key to be exported.

Note

You can only export the keys in a group that belongs to an ESK keystore.

4. Select the **Key Attributes**. The options are:

- Encrypt, Decrypt, Sign
- Verify, Wrap, Unwrap

Note

- For *EC* keys, only Sign and Verify are available. For *AES* keys, Sign and Verify are not available.
- For a master/primary key, you must select Encrypt and Decrypt as the key attributes.

- Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SELECT GROUP, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

- Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SELECT GROUP**, **SOURCE KEY**, and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

- Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY and DESTINATION KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

- Click **Close**. The Add SAP Key wizard is closed.

The newly created key is displayed in the list of SAP keys.

Uploading CipherTrust (Local) Key Material

Upload the local key material using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

- Select **CipherTrust (Local)**.
- Click **Next**. The Select Group screen is displayed.

Select Group

- Select a **Group** from the list.

Note

If you are creating a master/primary key, you must select a group that allows you to create a master/primary key. The following groups are supported.

- S4/HANA
- Integrated Business Planning

- SAP Commissions
- HANA Enterprise Cloud
- SAP Marketing Cloud
- SuccessFactors
- Analytics Cloud

2. Click **Next**. The Configure CipherTrust Key screen is displayed.

Configure CipherTrust (Local) Key

1. Enter a **Key Name**. A new key with this name will be created on the CipherTrust Manager and its key material will be uploaded to SAP cloud.
2. Select **Key Type**. The options are:
 - **AES**: Creates and uploads an AES key.
 - **RSA**: Creates and uploads an RSA key pair.
 - **EC**: Creates and uploads an EC key.
3. Select the **Key Size / Elliptic Curve** based on the [key type](#).

Note

If you are creating a master/primary key, you must select RSA as the key type, and 3072 or 4096 as the key size.

4. Click **Next**. The Configure SAP Key screen is displayed.

Configure SAP Key

1. (Optional) Provide a basic **Description** for the key.
2. (To create a master/primary key) Select **Create as Master/Primary key**.

Note

You can create only one master/primary key for a group. If a master/primary key already exists for the selected group, you can't select **Create as Master/Primary key**.

3. (Optional) Select **Allow Key Export**. Selecting this check box allows the key to be exported.

Note

You can only export the keys in a group that belongs to an ESK keystore.

4. Select the **Key Attributes**. The options are:

- Encrypt, Decrypt, Sign
- Verify, Wrap, Unwrap

Note

- For *EC* keys, only Sign and Verify are available. For *AES* keys, Sign and Verify are not available.
- For a master/primary key, you must select Encrypt and Decrypt as the key attributes.

5. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SELECT GROUP, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SELECT GROUP**, **SOURCE KEY**, and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window

open until the process is completed.

When the status next to the SOURCE KEY and DESTINATION KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add SAP Key wizard is closed.

The newly created key is displayed in the list of SAP keys.

Creating SAP (Native) Key Material

Create the key material directly using a native SAP application.

Select Material Origin > Select Source

1. Select **SAP (Native)**.
2. Click **Next**. The Select Group screen is displayed.

Select Group

1. Select a **Group** from the list.

Note

If you are creating a master/primary key, you must select a group that allows you to create a master/primary key. The following groups are supported.

- S4/HANA
- Integrated Business Planning
- SAP Commissions
- HANA Enterprise Cloud
- SAP Marketing Cloud
- SuccessFactors
- Analytics Cloud

2. Click **Next**. The Configure SAP Key screen is displayed.

Configure SAP Key

1. Enter a unique, user-friendly alias as the **SAP Key Name**. This will be the key name on SAP cloud. This name helps uniquely identify a SAP key.

2. (Optional) Provide a basic **Description** for the key.
3. Select **Key Type**. The options are:
 - **AES**: Creates and uploads an AES key.
 - **RSA**: Creates and uploads an RSA key pair.
 - **EC**: Creates and uploads an EC key.
4. Select the **Key Size / Elliptic Curve** based on the [key type](#).

Note

The key size 8192 is applicable to a non-FIPS Luna HSM.

Note

If you are creating a master/primary key, you must select RSA as the key type, and 3072, 4096, or 8192 as the key size.

5. (To create a master/primary key) Select **Create as Master/Primary key**.

Note

You can create only one master/primary key for a group. If a master/primary key already exists for the selected group, you can't select **Create as Master/Primary key**.

6. (Optional) Select **Allow Key Export**. Selecting this check box allows the key to be exported.

Note

You can only export the keys in a group that belongs to an ESK keystore.

7. Select the **Key Attributes**. The options are:
 - Encrypt, Decrypt, Sign

- Verify, Wrap, Unwrap

Note

- For *EC* keys, only Sign and Verify are available. For *AES* keys, Sign and Verify are not available.
- For a master/primary key, you must select Encrypt and Decrypt as the key attributes.

8. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SELECT GROUP, and NATIVE KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **NATIVE KEY** or **SELECT GROUP** section and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the NATIVE KEY section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add SAP Key wizard is closed.

The newly created key is displayed in the list of SAP keys. The origin of the key is **NATIVE**.

Uploading Vormetric DSM Key Material

Upload the local key material using the Vormetric DSM to configure the source key.

Select Material Origin > Select Source

1. Select **Vormetric DSM**.
2. Click **Next**. The Select Group screen is displayed.

Select Group

1. Select a **Group** from the list.

Note

If you are creating a master/primary key, you must select a group that allows you to create a master/primary key. The following groups are supported.

- S4/HANA
- Integrated Business Planning
- SAP Commissions
- HANA Enterprise Cloud
- SAP Marketing Cloud
- SuccessFactors
- Analytics Cloud

2. Click **Next**. The Configure DSM Key screen is displayed.

Configure DSM Key

1. Enter a **DSM Key Name**. A new key with this name will be created on the DSM and its key material will be uploaded to SAP cloud.
2. (Optional) Provide a **Description** for the key.
3. Select a **DSM Domain** for the key. The drop-down list shows the DSM domains linked with the configured DSM connection.
4. Select **Key Type**. The options are:
 - **AES**: Creates and uploads an AES key.
 - **RSA**: Creates and uploads an RSA key pair.
5. Select the **Key Size** based on the [key type](#).
6. Click **Next**. The Configure SAP Key screen is displayed.

Configure SAP Key

1. (Optional) Provide a basic **Description** for the key.
2. (To create a master/primary key) Select **Create as Master/Primary key**.

Note

You can create only one master/primary key for a group. If a master/primary key already exists for the selected group, you can't select **Create as Master/Primary key**.

3. (Optional) Select **Allow Key Export**. Selecting this check box allows the key to be exported.

Note

You can only export the keys in a group that belongs to an ESK keystore.

4. Select the **Key Attributes**. The options are:

- Encrypt, Decrypt, Sign
- Verify, Wrap, Unwrap

Note

- For AES keys, Sign and Verify are not available.
- For a master/primary key, you must select Encrypt and Decrypt as the key attributes.

5. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SELECT GROUP, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SELECT GROUP**, **SOURCE KEY**, and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window

open until the process is completed.

When the status next to the SOURCE KEY and DESTINATION KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add SAP Key wizard is closed.

The newly created key is displayed in the list of SAP keys.

Uploading Luna HSM Key Material

Upload the local key material using the Luna HSM to configure the source key.

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Select Material Origin > Select Source

1. Select **CipherTrust Luna HSM**.
2. Click **Next**. The Select Group screen is displayed.

Select Group

1. Select a **Group** from the list.

Note

If you are creating a master/primary key, you must select a group that allows you to create a master/primary key. The following groups are supported.

- S4/HANA
- Integrated Business Planning
- SAP Commissions
- HANA Enterprise Cloud
- SAP Marketing Cloud
- SuccessFactors

- Analytics Cloud

2. Click **Next**. The Configure HSM Key screen is displayed.

Configure HSM Key

1. Select the **Partition ID** of the desired Luna HSM partition.
2. Enter an **HSM Key Name**. A new key with this name will be created on the Luna HSM and its key material will be uploaded to SAP cloud.
3. Select the **Key Type** for the key. The options are:
 - **AES**: Creates and uploads an AES key.
 - **RSA**: Creates and uploads an RSA key pair.
4. Select the **Key Size** based on the [key type](#).

Note

If you are creating a master/primary key, you must select RSA as the key type, and 3072, 4096, or 8192 as the key size.

5. (Applicable to RSA algorithm only) Select the key **Mechanism**. The supported key mechanisms are:
 - CKM_RSA_FIPS_186_3_AUX_PRIME_KEY_PAIR_GEN
 - CKM_RSA_X9_31_KEY_PAIR_GEN
 - CKM_RSA_FIPS_186_3_PRIME_KEY_PAIR_GEN
 - CKM_RSA_PKCS_KEY_PAIR_GEN

Note

The mechanism for the AES algorithm is CKM_AES_KEY_GEN.

6. Select the **Key Attributes**. The options are:
 - Modifiable, Extractable, Sensitive (all *three* are selected for a **BYOK Compatible** key)

- Encrypt, Decrypt, Wrap, Unwrap
- Sign, Verify, Derive

7. Click **Next**. The Configure SAP Key screen is displayed.

Configure SAP Key

1. (Optional) Provide a basic **Description** for the key.
2. (To create a master/primary key) Select **Create as Master/Primary key**.

Note

You can create only one master/primary key for a group. If a master/primary key already exists for the selected group, you can't select **Create as Master/Primary key**.

3. (Optional) Select **Allow Key Export**. Selecting this check box allows the key to be exported.

Note

You can only export the keys in a group that belongs to an ESK keystore.

4. Select the **Key Attributes**. The options are:

- Encrypt, Decrypt, Sign
- Verify, Wrap, Unwrap

Note

For a master/primary key, you must select Encrypt and Decrypt as the key attributes.

5. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY and DESTINATION KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add SAP Key wizard is closed.

The newly created key is displayed in the list of SAP keys.

Cloning Existing Key Material

To add a new SAP cloud key by cloning key material existing on the CipherTrust Manager (External), CipherTrust Manager (Local), DSM, or Luna HSM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**.
3. Click **Add Key**. The Select Material Origin screen of the Add SAP Key wizard is displayed.
4. Under **Select Method**, select **Clone Existing Key Material**. The **Select Source** section appears. Depending on your requirements, select from the following:
 - **CipherTrust (External)**: Refer to [Cloning CipherTrust \(External\) Key Material](#) for details.
 - **CipherTrust (Local)**: Refer to [Cloning CipherTrust \(Local\) Key Material](#) for details.
 - **Vormetric DSM**: Refer to [Cloning Vormetric DSM Key Material](#) for details.
 - **Luna HSM**: Refer to [Cloning Luna HSM Key Material](#) for details.

Refer to [Key Creation Methods and Sources](#) for details on these key sources.

Cloning CipherTrust (External) Key Material

Clone and upload the key material of an external CipherTrust Manager key using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

1. Select **CipherTrust (External)**.
2. Click **Next**. The Select Group screen is displayed.

Select Group

1. Select a **Group** from the list.

Note

If you are creating a master/primary key, you must select a group that allows you to create a master/primary key. The following groups are supported.

- S4/HANA
- Integrated Business Planning
- SAP Commissions
- HANA Enterprise Cloud
- SAP Marketing Cloud
- SuccessFactors
- Analytics Cloud

2. Click **Next**. The Select CipherTrust (External) Key screen is displayed.

Select CipherTrust (External) Key

1. **Select Domain** from the drop-down list. The drop-down list shows the external CM domains linked with the configured external CM connection.
2. Select the **Algorithm** for the key. The options are:
 - **AES**: Creates and uploads an AES key.
 - **RSA**: Creates and uploads an RSA key pair.
 - **EC**: Creates and uploads an EC key.
3. Select the **Key Size / Elliptic Curve** based on the [key type](#).
4. Select the desired key from the **Key Name** drop-down list. This field shows the available local CipherTrust Manager keys.
5. Click **Next**. The Configure SAP Key screen is displayed.

Configure SAP Key

1. Enter a unique, user-friendly alias as the **SAP Key Name**. This will be the key name on SAP cloud. This name

helps uniquely identify a SAP key. By default, the CipherTrust Key Name you specified on the previous screen is populated.

2. (Optional) Provide a basic **Description** for the key.
3. (To create a master/primary key) Select **Create as Master/Primary key**.

Note

You can create only one master/primary key for a group. If a master/primary key already exists for the selected group, you can't select **Create as Master/Primary key**.

4. (Optional) Select **Allow Key Export**. Selecting this check box allows the key to be exported.

Note

You can only export the keys in a group that belongs to an ESK keystore.

5. Select the **Key Attributes**. The options are:

- Encrypt, Decrypt, Sign
- Verify, Wrap, Unwrap

Note

For *EC* keys, only Sign and Verify are available. For *AES* keys, Sign and Verify are not available.

6. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SELECT GROUP, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SELECT GROUP**, **SOURCE KEY**, and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as

appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY section becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add SAP Key wizard is closed.

The newly created key is displayed in the list of SAP keys.

Cloning CipherTrust (Local) Key Material

Clone and upload the local key material using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

1. Select **CipherTrust (Local)**.
2. Click **Next**. The Select Group screen is displayed.

Select Group

1. Select a **Group** from the list.

Note

If you are creating a master/primary key, you must select a group that allows you to create a master/primary key. The following groups are supported.

- S4/HANA
- Integrated Business Planning
- SAP Commissions
- HANA Enterprise Cloud
- SAP Marketing Cloud
- SuccessFactors
- Analytics Cloud

2. Click **Next**. The Select CipherTrust (Local) Key is displayed.

Select CipherTrust (Local) Key

1. Select **Key Type**. The options are:
 - **AES**: Creates and uploads an AES key.
 - **RSA**: Creates and uploads an RSA key pair.
 - **EC**: Creates and uploads an EC key.
2. Select the **Key Size / Elliptic Curve** based on the [key type](#).
3. Select the desired key from the **Key Name** drop-down list. This field shows the available local CipherTrust Manager keys.
4. Click **Next**. The Configure SAP Key screen is displayed.

Configure SAP Key

1. Enter a unique, user-friendly alias as the **SAP Key Name**. This will be the key name on SAP cloud. This name helps uniquely identify a SAP key. By default, the CipherTrust Key Name you specified on the previous screen is populated.
2. Select the desired **SAP Group Name** from the drop-down list. The drop-down shows the list of SAP groups added to the CCKM.
3. (Optional) Provide a basic **Description** for the key.
4. (To create a master/primary key) Select **Create as Master/Primary key**.

Note

You can create only one master/primary key for a group. If a master/primary key already exists for the selected group, you can't select **Create as Master/Primary key**.

5. (Optional) Select **Allow Key Export**. Selecting this check box allows the key to be exported.

Note

You can only export the keys in a group that belongs to an ESK keystore.

6. Select the **Key Attributes**. The options are:
 - Encrypt, Decrypt, Sign

- Verify, Wrap, Unwrap

Note

For *EC* keys, only Sign and Verify are available. For *AES* keys, Sign and Verify are not available.

7. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SELECT GROUP, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SELECT GROUP**, **SOURCE KEY**, and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY section becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add SAP Key wizard is closed.

The newly created key is displayed in the list of SAP keys.

Cloning Vormetric DSM Key Material

Clone and upload the local key material using the Vormetric DSM to configure the source key.

Select Material Origin > Select Source

1. Select **Vormetric DSM**.
2. Click **Next**. The Select Group screen is displayed.

Select Group

1. Select a **Group** from the list.

Note

If you are creating a master/primary key, you must select a group that allows you to create a master/primary key. The following groups are supported.

- S4/HANA
- Integrated Business Planning
- SAP Commissions
- HANA Enterprise Cloud
- SAP Marketing Cloud
- SuccessFactors
- Analytics Cloud

2. Click **Next**. The Select DSM Key screen is displayed.

Select DSM Key

1. Select **Key Type**. The options are:
 - **AES**: Creates and uploads an AES key.
 - **RSA**: Creates and uploads an RSA key pair.
2. Select the **Key Size** based on the [key type](#).
3. Select the desired key from the **DSM Key Name** drop-down list. This field displays the available DSM keys.
4. Click **Next**. The Configure SAP Key screen is displayed.

Configure SAP Key

1. Enter a unique, user-friendly alias as the **SAP Key Name**. This will be the key name on SAP cloud. This name helps uniquely identify a SAP key. By default, the DSM Key Name you specified on the previous screen is populated.
2. (Optional) Provide a basic **Description** for the key.
3. (To create a master/primary key) Select **Create as Master/Primary key**.

Note

You can create only one master/primary key for a group. If a master/primary key already exists for the selected group, you can't select **Create as Master/Primary key**.

4. (Optional) Select **Allow Key Export**. Selecting this check box allows the key to be exported.

Note

You can only export the keys in a group that belongs to an ESK keystore.

5. Select the **Key Attributes**. The options are:

- Encrypt, Decrypt, Sign
- Verify, Wrap, Unwrap

Note

For AES keys, Sign and Verify are not available.

6. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SELECT GROUP, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SELECT GROUP**, **SOURCE KEY**, and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add SAP Key wizard is closed.

The newly created key is displayed in the list of SAP keys.

Cloning Luna HSM Key Material

Note

The Functionality Module (FM) configuration on the Luna Network HSM is not compatible with CipherTrust Manager integrations.

Clone and upload the local key material using the Luna HSM to configure the source key.

Select Material Origin > Select Source

1. Select **Luna HSM**.
2. Click **Next**. The Select Group screen is displayed.

Select Group

1. Select a **Group** from the list.

Note

If you are creating a master/primary key, you must select a group that allows you to create a master/primary key. The following groups are supported.

- S4/HANA
- Integrated Business Planning
- SAP Commissions
- HANA Enterprise Cloud
- SAP Marketing Cloud
- SuccessFactors
- Analytics Cloud

-
2. Click **Next**. The Select HSM Key screen is displayed.

Select HSM Key

1. Select the **Partition ID** of the desired Luna HSM partition.
2. Select the **Algorithm** (key type) for the key. The options are:
 - **AES**: Creates and uploads an AES key.
 - **RSA**: Creates and uploads an RSA key pair.
3. Select the **Key Size** based on the [key type](#).
4. Select the desired **HSM Key** key from the list.
5. Click **Next**. The Configure SAP Key screen is displayed.

Configure SAP Key

1. Enter a unique, user-friendly alias as the **SAP Key Name**. This will be the key name on SAP cloud. This name helps uniquely identify a SAP key. By default, the HSM Key Name you specified on the previous screen is populated.
2. (Optional) Provide a basic **Description** for the key.
3. (To create a master/primary key) Select **Create as Master/Primary key**.

Note

You can create only one master/primary key for a group. If a master/primary key already exists for the selected group, you can't select **Create as Master/Primary key**.

4. (Optional) Select **Allow Key Export**. Selecting this check box allows the key to be exported.

Note

You can only export the keys in a group that belongs to an ESK keystore.

5. Select the **Key Attributes**. The options are:
 - Encrypt, Decrypt, Sign
 - Verify, Wrap, Unwrap

6. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SELECT GROUP, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SELECT GROUP**, **SOURCE KEY**, and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY section becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add SAP Key wizard is closed.

The newly created key is displayed in the list of SAP keys.

Viewing SAP Keys

The SAP Keys page shows the list of SAP cloud keys available on the CipherTrust Manager.

To view the SAP keys:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**. The list of available SAP keys is displayed. The SAP Keys page displays the following details:

Field	Description
Key Name	Unique, user-friendly name of the SAP key. Click the link to view additional details of the key or edit the key. Refer to Viewing or Editing Details of SAP Keys . This name is useful in searching for specific keys.
Algorithm	Algorithm of the SAP key. AES, RSA, and EC algorithms with different keys sizes and elliptic curves are supported.
Status	State of the SAP key. The status can be: • Available

Field	Description
	• Not Available • Deleted
Key State	State of the SAP key. The status can be Enabled or Disabled.
Created By	Name of the user who created the key. Note: SAP excludes the creator identifiable fields from the API responses for the technical users. If the connection to the SAP Data Custodian is established by a technical user, this field will be blank.
Version Count	Number of key versions.
Creation Date	Date and time when the SAP key is created.
Operations	Operations allowed using the SAP key.
Group	SAP group where the SAP key resides.
Version	Version of the key.
Origin	Source of the key material used for the version. The origin can be: • CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud. • External (Unknown): Source of the key material is unknown. It is different than CCKM and the native cloud. Refer to Key Creation Methods and Sources for details.
Application	SAP application (for example, SAP S/4HANA Cloud) where the key is used.
Tenant	SAP tenant in which the key is created.
Allow Key Export	Whether the key export is allowed. The setting can be: • Enabled: The key export is allowed. • Disabled: The key export is not allowed.

The **Operations**, **Group**, **Version**, **Origin**, **Application**, **Tenant**, and **Allow Key Export** columns are hidden by default. To show/hide a column, click the custom view icon (⚙️), select/clear the desired column, and click **OK**.

Refreshing SAP Keys

Refreshing is the process of downloading keys created in SAP groups to CCKM. You can refresh keys from all SAP groups at once.

To refresh all keys:

1. Open the Cloud Key Manager application.
2. In the left pane, click **Cloud Keys > SAP**. The SAP Keys page is displayed. This page displays the list of SAP keys.
3. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all keys is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message Refresh started... is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > SAP > SAP Keys** page.

Viewing Versions of a Key

To view the versions of a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**.
3. Click the expand icon () to the left of the desired key. The key versions are displayed.

Disabling a SAP Key

If required, you can disable an enabled key. A disabled key cannot operate on data. Disabling a key disables all versions of the key.

To disable a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**.
3. Click the overflow icon () corresponding to the desired key.

-
4. Click **Disable**. The Disable Key dialog box is displayed.
 5. Click **Disable** to confirm the action.

The state of the key changes to Disabled.

Enabling a SAP Key

If required, you can enable a disabled key. Enabling a key enables all versions of the key.

To enable a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Enable**. The Enable Key dialog box is displayed.
5. Click **Enable** to confirm the action.

The state of the key changes to Enabled.

Adding a Key Version

CCKM provides two methods to add a new version to a key. Refer to [Key Creation Methods and Sources](#) for details on key creation methods and key sources.

To add a new key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Add Version**. The Add Version dialog box is displayed.
5. **Select Method**. The options are:
 - **Create/Upload New Key Material**: Refer to [Adding Key Version by Creating/Uploading Key Material](#).
 - **Clone Existing Key Material**: Refer to [Adding Key Version by Cloning Existing Key Material](#).

Adding Key Version by Creating New Key Material

1. Select **Create/Upload New Key Material** as the method.
2. **Select Source**. The options are:

- **CipherTrust (External):** Select this option, select the **Domain Name**, and specify a **Key Name** for the new key version.
- **CipherTrust (Local):** Select this option and specify **Key Name** for the new key version.
- **SAP (Native):** Select this option to create a new native SAP key.
- **Vormetric DSM:** Select this option, specify **Key Name** for the new key version and select the **DSM Domain**.
- **Luna HSM:** Select this option, select the **Partition ID**, select **Key Attributes**, and specify **Key Name** for the new key version.

The key attributes Modifiable, Extractable, and Sensitive are selected for a **BYOK Compatible** key.

3. Click **Add Version**.

A new version is added to the key. The Version Count increases by one on the SAP Keys page.

Adding Key Version by Cloning Existing Key Material

1. Select **Clone Existing Key Material** as the method.
2. **Select Source.** The options are:
 - **CipherTrust (External):** Select this option and **Select a source key** for the new key version.
 - **CipherTrust (Local):** Select this option and **Select a key source** for the new key version.
 - **Vormetric DSM:** Select this option and **Select a key source** for the new key version.
 - **Luna HSM:** Select this option and **Select a key source** for the new key version.
3. Click **Next**. The Select Source Key and Add Version screen is displayed.

Select Source Key and Add Version

1. (Optional, applicable to CipherTrust (Local) Key only) Enable **Fetch Latest Version Only**.
2. Select the desired key from the **Key Name** list.
3. Click **Add Version**.

A new version is added to the key. The Version Count increases by one on the SAP Keys page.

Deleting a SAP Key

If no longer required, you can delete a key. The delete operation deletes the key from SAP cloud.

Deleting a Non Exportable Key

Deletion of a non exportable key is irrecoverable. Because the key is not exportable, CCKM cannot back up this key, so it cannot be restored after deletion.

To delete a non exportable key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**.
3. Click the overflow icon () corresponding to the desired key.
4. Click **Delete**. The Remove dialog box is displayed.

Warning

This will delete the key from SAP. This operation is not recoverable. Because the key is not exportable, CCKM is unable to backup this key, so it cannot be restored after deletion. Are you sure you want to delete the key?

5. Click **Delete**. The Delete Key dialog box is displayed. It shows the name of the key being deleted in **bold**.

This dialog box acts as a secondary confirmation so that you are aware of the consequences of the delete operation.

6. Type the name of the key to be deleted. When the typed key name matches the given key name, the Delete button is enabled.
7. Click **Delete** to confirm the deletion. To cancel the key deletion, click **Cancel**.

A message stating that delete key is in progress is displayed. After the key is deleted successfully, it is removed from the list of SAP keys.

Deleting an Exportable Key

When deleting an exportable key, you have the option to proceed with the key deletion even if the key backup fails. If you select this option, CCKM will automatically take a fresh backup of the key before deleting it. If the backup process fails, it will proceed with deleting the key anyway. This may mean the key cannot be restored to its current state.

To delete an exportable key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**.
3. Click the overflow icon () corresponding to the desired key.

- Click **Delete**. The Delete Key dialog box is displayed.
- (Optional) Select **Delete even if backup fails**.

Note

If you select this option, CCKM will automatically take a fresh backup of the key before deleting it. If the backup process fails, it will proceed with deleting the key anyway. This may mean the key cannot be restored to its current state.

- Click **Delete**. The Delete Key dialog box is displayed. It shows the name of the key being deleted in **bold**.

This dialog box acts as a secondary confirmation so that you are aware of the consequences of the delete operation.

- Type the name of the key to be deleted. When the typed key name matches the given key name, the Delete button is enabled.
- Click **Delete** to confirm the deletion. To cancel the key deletion, click **Cancel**.

A message stating that delete key is in progress is displayed. After the key is deleted successfully, it is removed from the list of SAP keys.

Deleting a Master/Primary Key

To delete a master/primary key:

- Open the **Cloud Key Manager** application.
- In the left pane, click **Cloud Keys > SAP**.
- Click the overflow icon () corresponding to the master/primary key.
- Click **Delete**. The Delete Master/Primary Key dialog box is displayed.
- (Optional) Select **Delete even if backup fails**.

Note

If you select this option, CCKM will automatically take a fresh backup of the key before deleting it. If the backup process fails, it will proceed with deleting the key anyway. This may mean the key cannot be restored to its current state.

- Click **Yes**.

7. For confirmation, enter the **full name of the master/primary key** you want to delete.
8. Select the following.
 - Data encrypted with a key may no longer be accessible once the key is deleted.
 - Key deletion cannot be reversed unless the key has been fully backed up.
 - I have read and acknowledge the consequences of key deletion.
9. Click **Delete** to confirm the deletion. To cancel the key deletion, click **Cancel**.

A message stating that the delete key is in progress is displayed. After the key is deleted successfully, it is removed from the list of SAP keys.

Removing a SAP Key

After the key is deleted, you can remove it from the CCKM. To remove the key.

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**.
3. Click the overflow icon () corresponding to the desired deleted key.
4. Click **Remove**. The Remove dialog box is displayed.
5. (Applicable to master/primary keys) Select an option to remove the key. You can select either **Remove Key Backup** or **Remove Key Backup and Metadata**.
6. Click **Next**.
7. Enter the **key name**.
8. Click **Remove**.

You will see a message that the key is removed successfully.

Viewing or Editing Details of SAP Keys

After a key is created, you can update key name and description, change exportability, and modify key attributes.

In the edit view of a key, you can view all the key details such as its purpose, protection level, and location etc.

To view or edit an SAP key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**. The list of available SAP keys is displayed.
3. Click the overflow icon () corresponding to the desired key and click **View/Edit**. Alternatively, you can click

the key name link. The edit view of the key is displayed. The edit view is divided into:

- **GENERAL INFO:** View and update key name and its description (refer to [Changing Key Details](#)). Also, you can change the exportability of the key (refer to [Changing Key Exportability](#)) and key attributes (refer to [Changing Key Attributes](#)).
- **KEY SCHEDULE:** Add, update, and disable a key rotation schedule. Refer to [Adding or Changing Key Rotation Schedule](#) and [Disabling Key Rotation Schedule](#).
- **KEY VERSIONS:** View details of key versions. Refer to [Viewing Key Version Details](#).

Changing Key Details

To modify the key details:

1. Expand the **GENERAL INFO** section, if needed.
2. Update the **SAP Key Name**.
3. Add or update **Description**.
4. Click **Update**.

The key details are updated.

Changing Key Exportability

To change key exportability:

1. Expand the **GENERAL INFO** section, if needed.
2. Clear or select **Allow Key Export**.
 - Select the check box to allow the key export.
 - Clear the check box to prevent the key export.

When Allow Key Export is disabled, the CipherTrust Manager cannot back up the key.

3. Click **Update**.

The key exportability is changed.

Changing Key Attributes

To modify key attributes:

1. Expand the **GENERAL INFO** section.
2. Under **Key Attributes**, select or clear the desired attributes.

3. Click **Update**.

The key attributes are updated.

Adding or Changing Key Rotation Schedule

To add or update a key rotation schedule:

1. Expand the KEY SCHEDULE section.
2. From **Select Rotation Schedule** drop-down list, select the desired schedule.
3. Select the **Key Origin**. The options are:
 - CipherTrust (External)
 - CipherTrust (Local)
 - Native
 - Luna, also select the Luna HSM **Partition**.
 - DSM, also select the DSM **Domain**.
4. Click **Update**.

The key rotation schedule is added/updated. The selected schedule is now assigned to the key. To view all the keys assigned to a schedule, refer to [Viewing Keys Assigned to Schedules](#).

Disabling Key Rotation Schedule

To disable a key rotation schedule:

1. Expand the KEY SCHEDULE section.
2. Next to the **Key Rotation Schedule** drop-down list, click the close icon ().

Auto key rotation is disabled.

Viewing Key Version Details

To view the details of key versions, expand the KEY VERSIONS section. The key version details are displayed:

Field	Description
Version	Version number of the key.
Key State	State of the key version. The state can be enabled or disabled.
Created By	User who created the key.

Field	Description
Creation Date	Date and time when the SAP key is created.
Operations	Operations allowed using the SAP key.
Source Key	Source key for the version.
Origin	Source of the key material. The origin of the key can be: • CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud. • External (Unknown): Source of the key material is unknown. It is different than CCKM and the native cloud.

Scheduling Operations

CCKM allows you to periodically refresh or rotate keys in the background. To refresh or rotate keys, you need to create schedule configuration. A schedule configuration defines when to run a refresh or rotation job. You can use either basic format or cron format to specify the time when the rotation or refresh job will run.

Specify basic format in the following order:

[Repeats, at]

Where,

- Repeats: frequency of the scheduler. Possible values are daily, weekly, monthly, and yearly.
- at: specific time at which the rotation or refresh operation will be performed. Possible value lies between 12:00-11:00 AM/PM UTC.

Specify cron format in the following order:

" * * * * * "

These five values indicate when the job should be executed. These values are mandatory and must be specified in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

Field	Allowed values	Allowed special characters
Minute	0-59	* / , -
Hour	0-23	* / , -
Day of month	1-31	* / , - ?
Month	1-12 or JAN-DEC	* / , -
Day of week	0-6 or SUN-SAT	* / , - ?

Adding Key Rotation Schedule

A key rotation schedule replaces selected keys with new cryptographic keys at the specified time.

To add key rotation schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. **Select Schedule Type as Key Rotation.**
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **SAP** from the **Cloud Name** drop-down list.
 - (Optional) Add **Description** for the scheduler in a maximum of 250 characters.
 - (Optional) Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. On the **Schedule Config** screen, enter or select the following details.
 - a. Configure **DURATION** for the scheduler. Specify the schedule start and end time:
 - **Schedule Starts**: Specify time when the schedule starts.
 - **Schedule Ends**: Unavailable by default, that is, the scheduler never expires.
 - **Never**: Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.
 - b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
 - c. Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created schedule is displayed in the schedules list.

Adding Key Refresh Schedule

A key refresh schedule synchronizes keys from linked SAP groups to Cloud Key Manager.

To add a key refresh schedule:

1. Open the **Cloud Key Manager** application.
-

-
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
 3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
 4. **Select Schedule Type** as **Key Refresh**.
 5. Click **Next**.
 6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **SAP** from the **Cloud Name** drop-down list.
 - (Optional) Add **Description** for the scheduler in a maximum of 250 characters.
 - (Optional) Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
 7. On the **Schedule Config** screen, enter or select the following details.
 - a. Configure **DURATION** for the scheduler. Specify the schedule start and end time:
 - **Schedule Starts**: Specify time when the schedule starts.
 - **Schedule Ends**: Unavailable by default, that is, the scheduler never expires.
 - **Never**: Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.
 - b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
 - c. Click **Next**.
 8. On the **Add Groups** screen:
 - a. Select the desired group from the **Refresh Keys from Selected Group(s)** drop-down list.
 - b. Click **+** button. Similarly, you can add more groups. To remove an added group, click the close (X) icon in the name of the group.

Alternatively, if you want to refresh all the existing groups, select the **Refresh All Groups** check box. When selected, the **Refresh Keys from Selected Group(s)** field becomes unavailable.

 - c. Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created scheduled is displayed in the schedules list.

Viewing/Editing Schedules

To view/edit a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The Schedules page displays the following details:

Field	Description
Name	Unique name of the scheduler configuration.
Schedule Type	Type of the schedule. Possible types are: • Key Rotation • Key Refresh • Add KMS Container (GCP and Azure only)
Cloud Name	Name of the cloud.
Last Modified	Time when the schedule is modified.
Frequency	Frequency of the scheduler configuration.
Start Date	Start time of the scheduler configuration.
End Date	Expiry time of the scheduler configuration. Never is displayed if the schedule is set to never expire.
Status	Status of the scheduler configuration. Possible values are: • Enabled • Disabled

3. Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.

Alternatively, click the **Name** link corresponding to the desired schedule.

4. Edit or configure the following information:

- **GENERAL INFO:**

- Basic description of the schedule.
- Status of the scheduler configuration. Select **Enable Schedule** to enable the schedule, clear to disable.

- **SCHEDULE CONFIG:** Scheduler configuration parameters such as duration and frequency of the schedule. Refer to the [Key Rotation Schedule Config Screen](#) or [Key Refresh Schedule Config Screen](#), as appropriate for details.

- **SAP GROUPS:** (For key refresh schedules only) Refer to [Add Groups](#) for details.

5. Click **Update** to save the changes.

The Schedules page also contains a section named **JOB HISTORY**. This section displays information related to a schedule such as **Run Date**, **Job ID**, and **Status**.

Viewing Keys Assigned to Schedules

Schedules for a key can be added, updated, and deleted on the KEY SCHEDULE section of the edit view of a key. Refer to [Adding or Changing Key Rotation Schedule](#) for details.

To view the keys assigned to a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The Schedules page is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.

Alternatively, click the **Name** link corresponding to the desired schedule.

The mini detail view of the schedule shows a numbered link next to the **Keys Assigned** label. The number indicates the number of keys assigned to the schedule. If no key is assigned, linked number 0 is displayed.

4. Click the numbered link. The SAP Keys for Schedule: <schedule-name> page is displayed. The page displays the following details:

Field	Description
Key Name	Unique, user-friendly name of the SAP key.
Algorithm	Algorithm of the SAP key. AES, RSA, and EC algorithms with different keys sizes and elliptic curves are supported.
Status	State of the SAP key. The status can be: • Available • Not Available • Deleted
Created By	User who created the key.
Operations	Operations allowed using the SAP key.
Group	SAP group where the SAP key resides.
Version	Version of the key.
Origin	Source of the key material used for the version. The origin can be: • CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud.

Field	Description
	• External (Unknown): Source of the key material is unknown. It is different than CCKM and the native cloud. Refer to Key Creation Methods and Sources for details.
Application	SAP application (for example, SAP S/4HANA Cloud) where the key is used.
Creation Date	Date and time when the SAP key is created.

Disabling Schedules

To disable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Disable**. The Disable Schedule message is displayed.
4. Click **Disable Schedule**.

A message **Successfully disabled the schedule** is displayed on the screen.

Enabling Schedules

To enable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Enable**. The Enable Schedule message is displayed.
4. Click **Enable Schedule**.

A message **Successfully enabled the schedule** is displayed on the screen.

Manually Running Schedules

To manually run a schedule:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Run Now**.

The Run Now schedule is started in the background. A message **Schedule is running now. It will take a few seconds to finish.** is displayed on the screen. After the schedule is run successfully, a message **Successfully finished running the schedule** is displayed.

Deleting Schedules

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Delete Schedule**. The Delete Schedule message is displayed.
4. Click **Delete**.

A message **Successfully deleted schedule** is displayed on the screen.

Managing SAP Reports

CCKM provides options to generate key visibility reports based on key-related activities between CCKM and SAP groups and rotation schedules. SAP reports are categorized as:

- **Key Activity Report:** Inspect individual SAP key histories by operations, for example, when they were refreshed, rotated, edited, or deleted. Also, use this report to compare key activities between CCKM and SAP group.
- **Key Rotation Report:** Track keys that are scheduled for rotation.

Creating SAP Reports

To create a SAP report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > SAP**. The SAP Reports page is displayed.
3. Click **Add Report**. The Choose Report Type and Name screen of the Add SAP Report wizard is displayed.
4. Select the report type. The options are **Key Activity Report** and **Key Rotation Report**.
5. Specify a **Report Name**. This is a mandatory field.
6. Click **Next**. The Select SAP Group screen of the wizard is displayed.
7. Select a group from the **SAP Groups** drop-down list. The drop-down list contains the list of SAP groups added to the CipherTrust Manager.

8. Click **Add**. The selected group appears under the Selected SAP Groups section.

Similarly, you can add as many groups as you want.

9. Click **Next**. The Set Start and End Dates screen of the wizard is displayed.
10. In the **Include Entries From** field, specify the start date and time for the report.
 - Click the field and select the date and time on the on-screen calendar.
 - Alternatively, enter the time in **MM/DD/YYYYHH:MM** format.
11. In the **To** field, specify the end date and time for the report.
12. Click **Save**.

A success message **Created report** is displayed. The newly created report is displayed under the SAP reports list. The report Run Date becomes *Running Now* and Status becomes *In Progress*. After the run is complete, the date and time of the run is displayed and the Status becomes Completed.

Viewing SAP Reports

The SAP Reports page displays the list of existing reports. Filter the reports by their names.

To view existing SAP reports:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > SAP**. The list of available reports is displayed. The following details of the reports are displayed:

Column Name	Description
Report Name	Name of the report.
Run Date	When the report is run. If a report is running currently, Running Now is displayed.
Type	Type of the report. The type can be: • Key Activity • Key Rotation
Start Date	Start date from when the report is generated.
End Date	End date for the report.
Status	Status of the report. The status can be: • Completed • In Progress

Viewing Details of a SAP Report

To view the details of a SAP report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > SAP**. The list of available reports is displayed.
3. Click the desired **Report Name** link.

Alternatively, click the overflow icon () corresponding to the desired report and click **View**.

The following details of the reports are displayed:

Column Name	Description
Key Name	Name of the key.
SAP User	Name of the SAP user who created the report.
Key Activity	Activity performed on the key.
Event Date	Time and date when the activity took place.
Created At	Date and time when the report is created.
Updated At	Date and time when the report is updated.

Downloading SAP Reports

To download a SAP report as a CSV file:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > SAP**. The list of available reports is displayed.
3. Click the overflow icon () corresponding to the desired report and click **Download**.

Alternatively, click the desired report under **Report Name**, and click **Download CSV** in the detail view.

The report is downloaded as a CSV file.

Deleting SAP Reports

To delete a SAP report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > SAP**. The list of available reports is displayed.
3. Click the overflow icon () corresponding to the desired report and click **Delete**. The Delete Report

message is displayed.

4. Click **Delete**.

The report is removed from the list of reports.

Viewing Linked SAP Applications

When a SAP group is added to the CipherTrust Manager, linked SAP applications are automatically added. One application can have multiple SAP groups associated with it.

When the last group linked with a SAP application is deleted from the CipherTrust Manager, the application is also deleted.

The Applications tab of the SAP Data Custodian Groups page shows the list of applications linked with the groups added to the CipherTrust Manager. Search for the applications by Tenant.

To view the list of applications:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > SAP Groups**. The Groups tab of the SAP Data Custodian Groups page shows the list of groups added to the CipherTrust Manager.
3. Click the **Applications** tab. The tab displays the following details:

Column	Description
Application	Name of the SAP application.
Tenant	Name of the SAP tenant. For SAP Technical Users (TUs), the field remains blank.
Date Added	Date and time when the application is added to the CipherTrust Manager.

Performance Summary

This section provides metrics and results of SAP performance testing.

Test Setup

Test Runner	
Operating System	Ubuntu 20.4
RAM	32GB
Core	16

Appliance	
Build	2.9.0.7341
RAM	16GB

Process

Scripts are written on top of the Locust Framework for load and stress testing. The testing is done for create, read, update, and delete (CRUD) key operations with respect to different numbers of virtual users (VUS) over parallel connections, one for every virtual user.

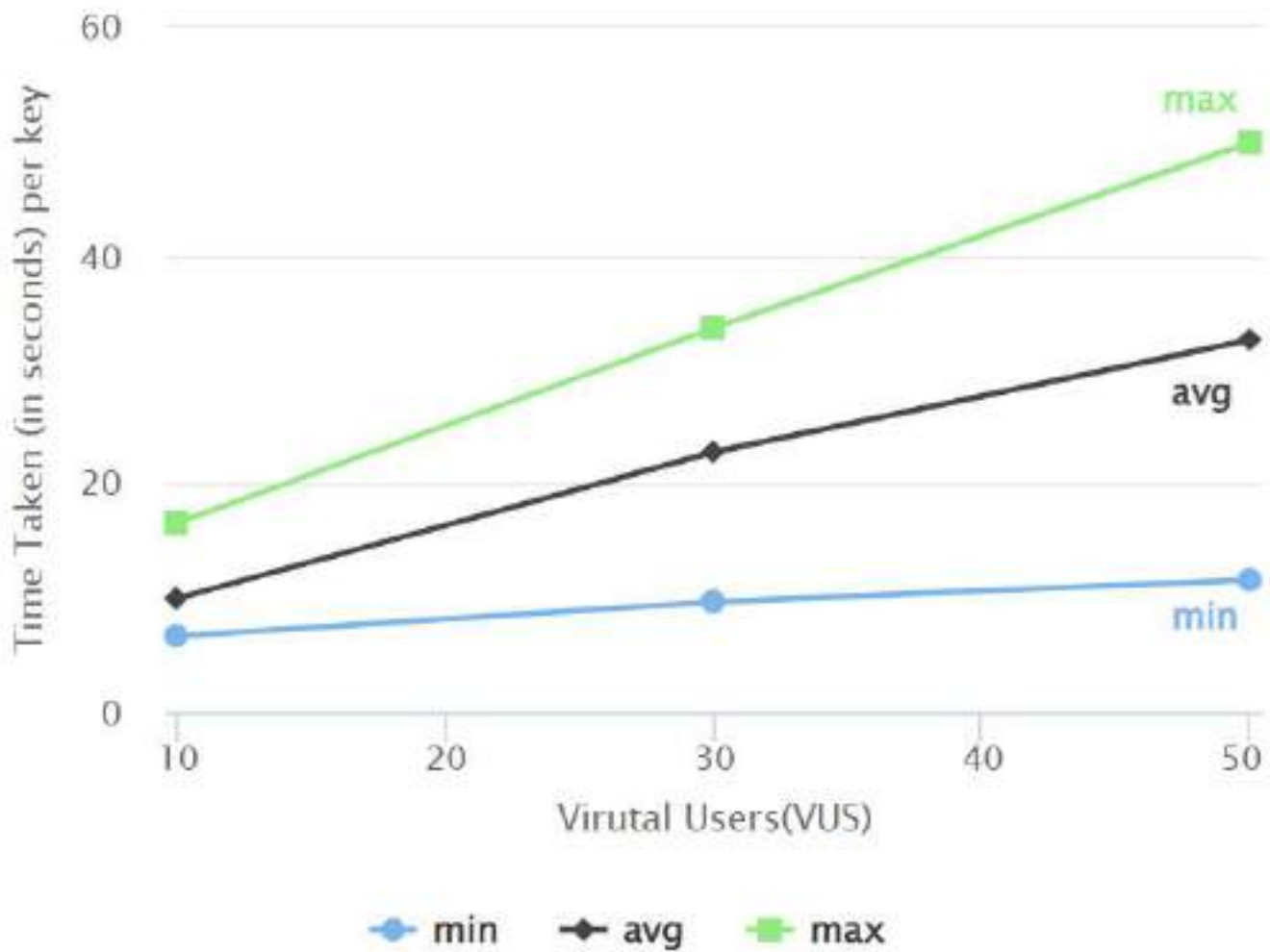
The testing also done to determine the response for uploading keys by a number (N) of virtual users in time (T).

CRUD Operations

Expand the headings below to view performance numbers.

Time taken per key upload with respect to virtual users

Virutal Users	Minimum	Average	Maximum
10	6.712	9.998	16.605
30	9.7	22.8	33.673
50	11.6	32.64	49.936



In the graph above:

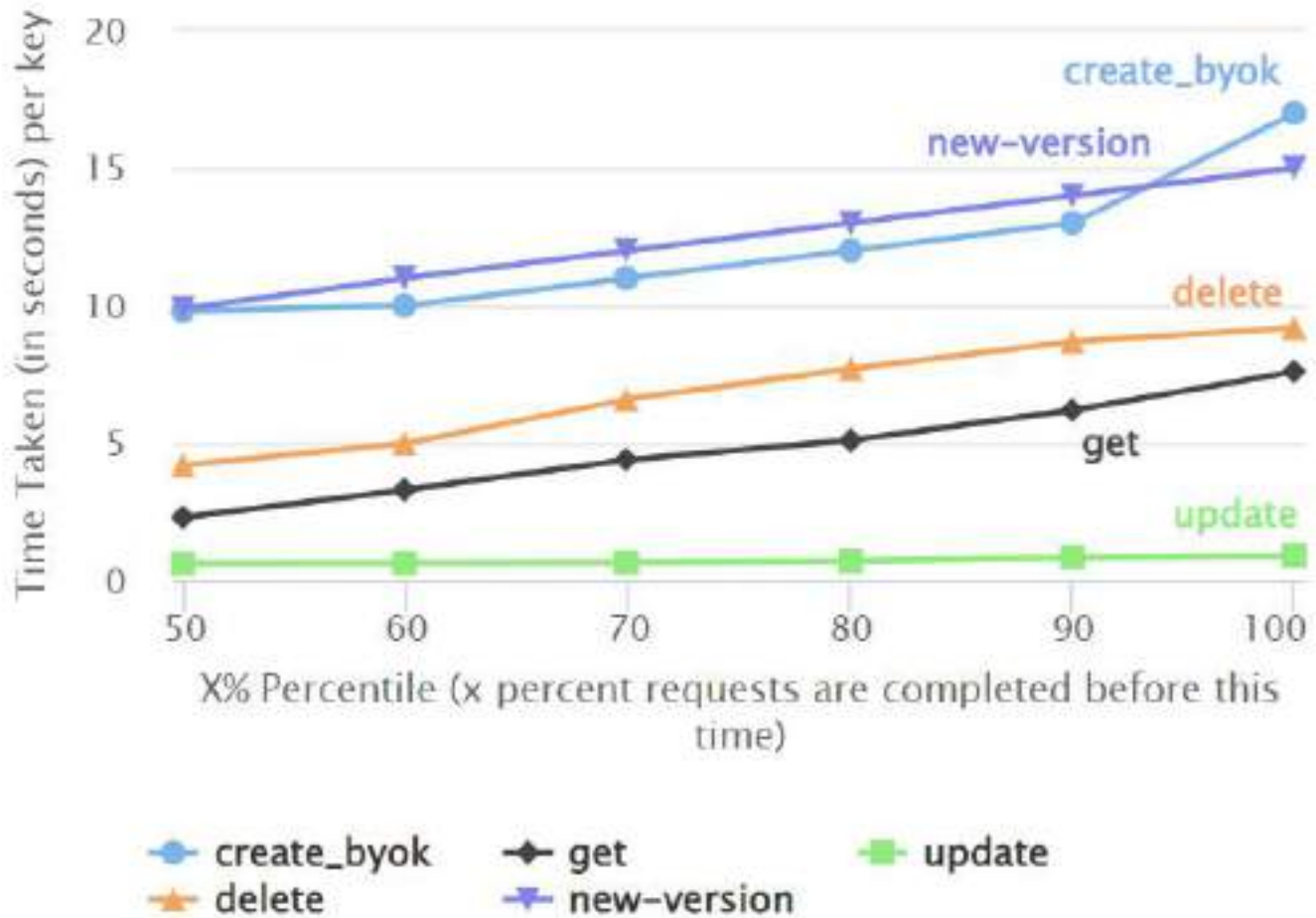
- X-axis represents the increment of virtual users.
- Y-axis represents the time taken by a virtual user to perform BYOK operations.
- 3 line graphs display information for minimum (min), average (avg), and maximum (max) time taken by one of the virtual users running at that time.

Response with request coverage in percentile (10 VUS, 180 seconds)

X% percentile	create_byok	get	update	delete	new-version
50	9.8	2.3	0.62	4.2	9.9
60	10	3.3	0.64	5	11
70	11	4.4	0.66	6.6	12
80	12	5.1	0.71	7.7	13
90	13	6.2	0.84	8.7	14

X% percentile	create_byok	get	update	delete	new-version
100	17	7.6	0.9	9.2	15

Here, X% percentile indicates that X percent requests are completed before this time.



In the graph above:

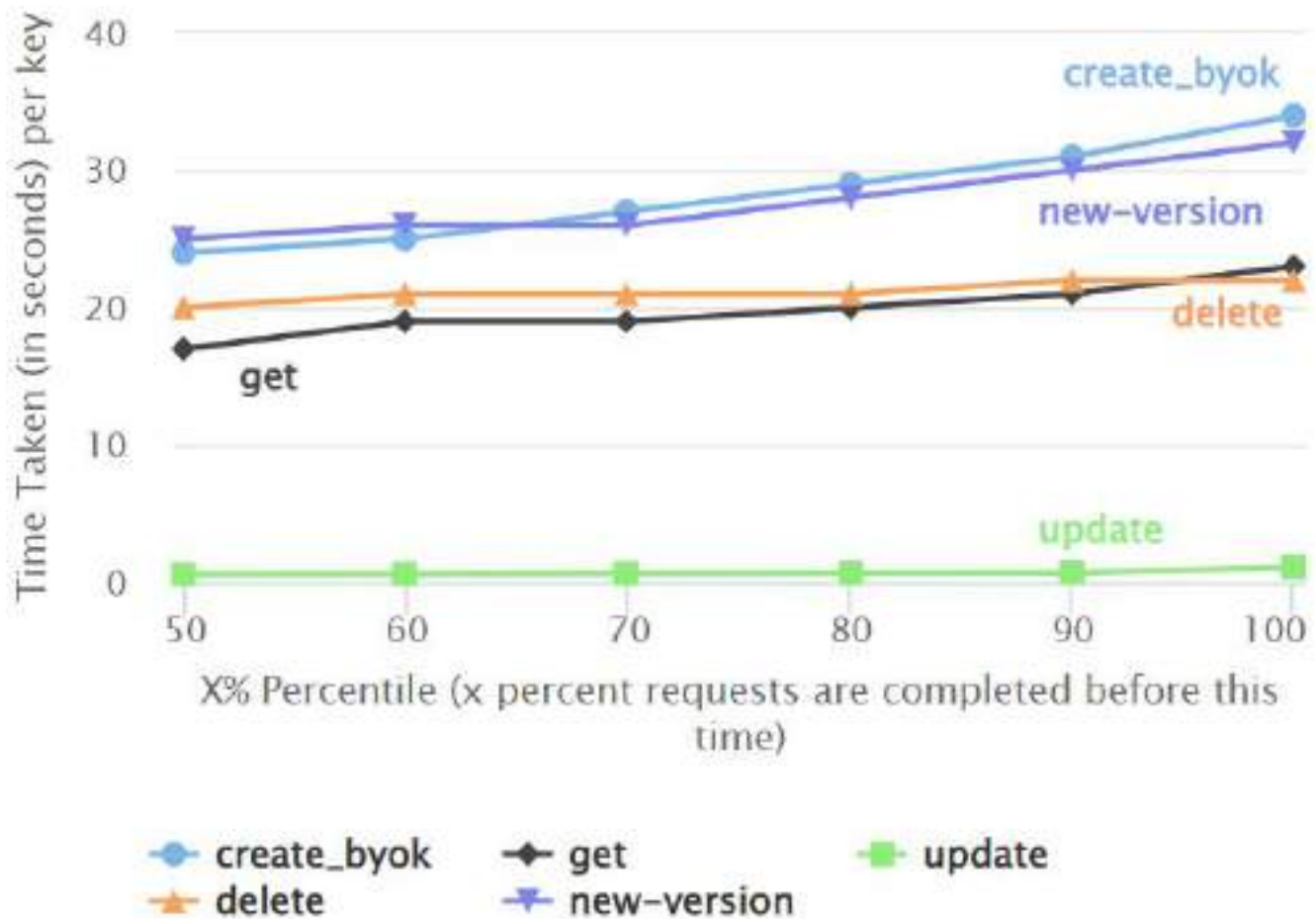
- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
- Y-axis represents the time taken by a virtual user to perform any CRUD operation.
- An individual virtual user has performed these CRUD operations sequentially, whereas all the virtual users are spawned in parallel.

Response with request coverage in percentile (30 VUS, 180 seconds)

X% percentile	create_byok	get	update	delete	new-version
50	24	17	0.59	20	25
60	25	19	0.62	21	26

X% percentile	create_byok	get	update	delete	new-version
70	27	19	0.64	21	26
80	29	20	0.67	21	28
90	31	21	0.71	22	30
100	34	23	1.1	22	32

Here, X% percentile indicates that X percent requests are completed before this time.



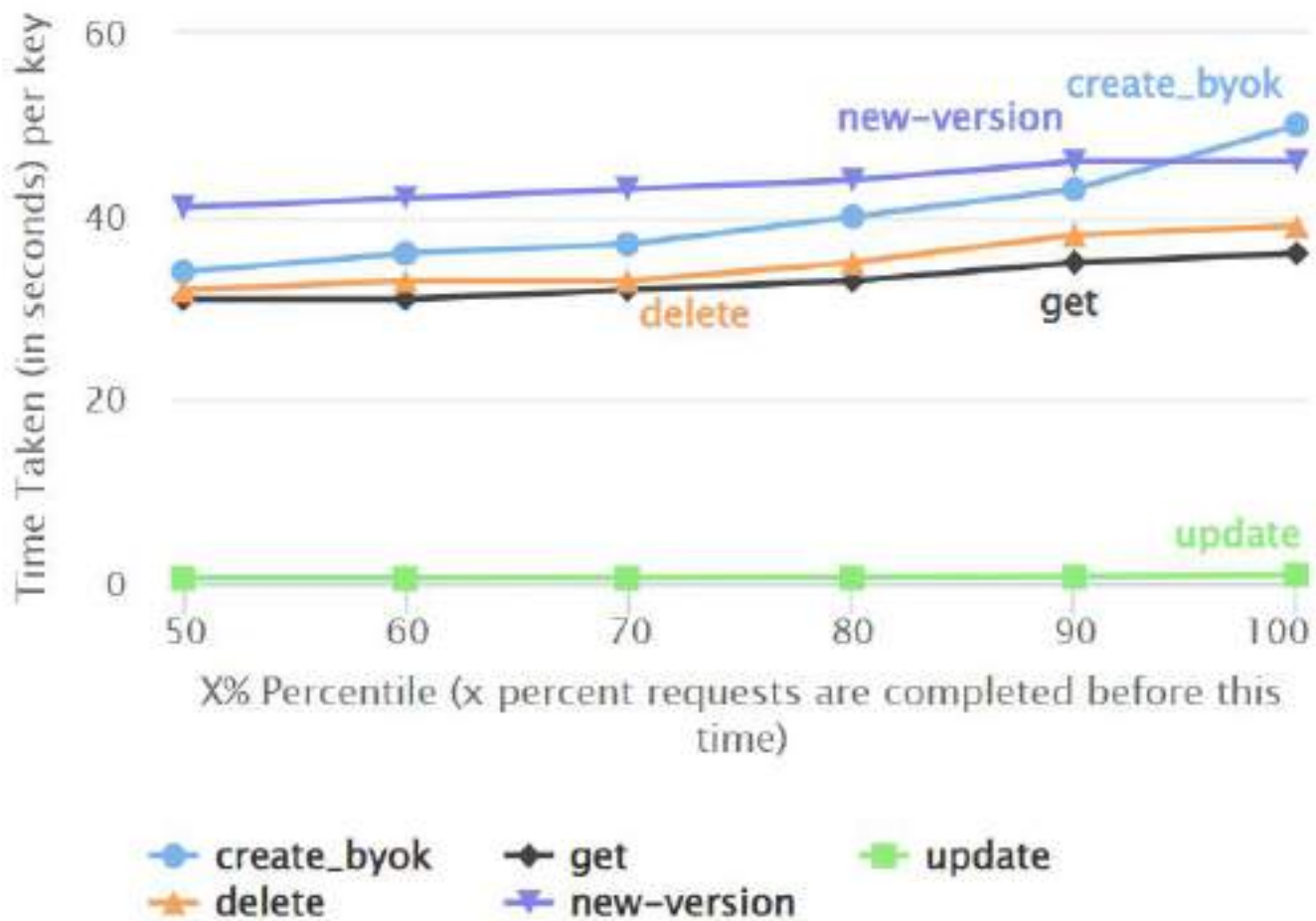
In the graph above:

- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
- Y-axis represents the time taken by a virtual user to perform any CRUD operation.
- An individual virtual user has performed these CRUD operations sequentially, whereas all the virtual users are spawned in parallel.
- t represents time taken for a key operation when N virtual users are running in parallel.

Response with request coverage in percentile (50 VUS, 180 seconds)

X% percentile	create_byok	get	update	delete	new-version
50	34	31	0.59	32	41
60	36	31	0.6	33	42
70	37	32	0.63	33	43
80	40	33	0.66	35	44
90	43	35	0.76	38	46
100	50	36	0.95	39	46

Here, X% percentile indicates that X percent requests are completed before this time.



In the graph above:

- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
- Y-axis represents the time taken by a virtual user to perform any CRUD operation.
- An individual virtual user has performed these CRUD operations sequentially, whereas all the virtual users are

spawned in parallel.

- t represents time taken for a key operation when N virtual users are running in parallel.
-

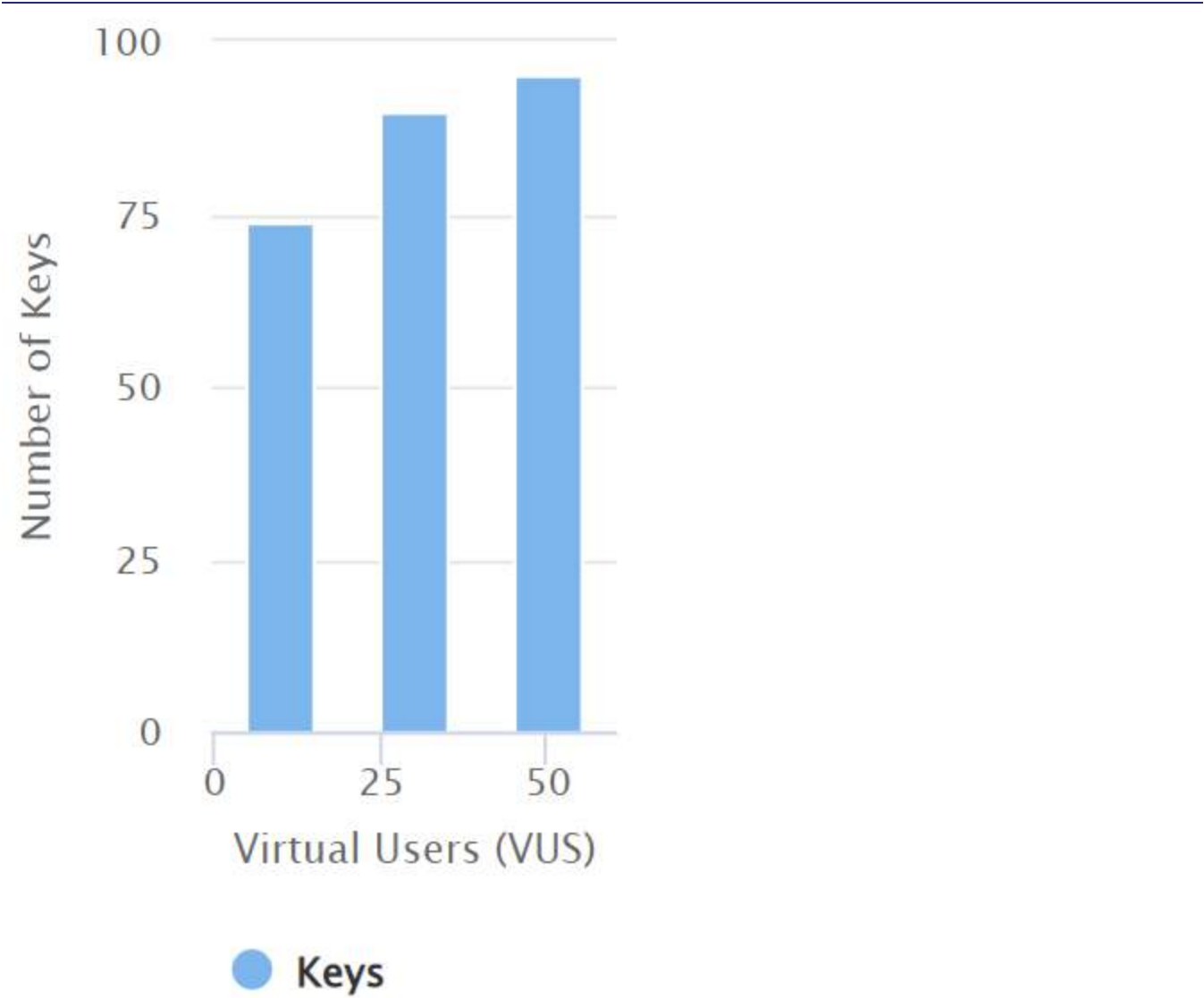
Key Upload

This section provides performance numbers for uploading keys by a number (N) of virtual users in time (T).

Expand the headings below to view performance numbers.

Number of keys uploaded by virtual users in 100 seconds

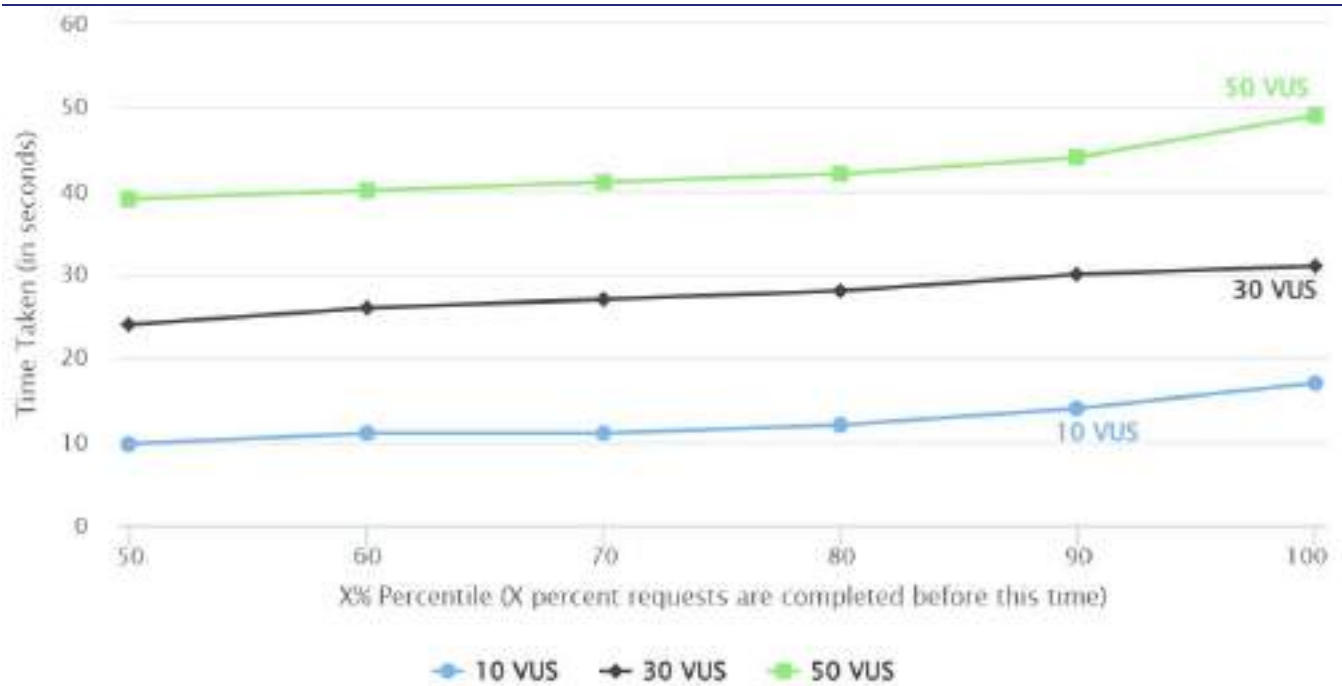
Virtual Users	Number of Keys
10	74
30	90
50	95



The graph above shows the number of keys uploaded in t time when N virtual users are spawned in parallel. Also, it indicates throttling condition breakpoints and helps you calculate throughput per second (TPS).

Response for upload with percentile request coverage

X% percentile	10 VUS	30 VUS	50 VUS
50	9.7	24	39
60	11	26	40
70	11	27	41
80	12	28	42
90	14	30	44
100	17	31	49



In the graph above:

- X-axis represents X% (percentile) of request time. It means, X% requests have taken time less than or equal to Y-axis value at X%.
- Y-axis represents the time taken by a virtual user to perform the BYOK (upload key) operation.
- While calculating TPS, Locust informs about the time taken to complete X% requests when all the virtual users are spawned in parallel.
- The number of virtual users does not significantly affect the number of keys being uploaded. This proves the rate limit implemented by SAP side causes throttling.

Downloading SAP Metadata

Downloading SAP Keys Metadata

CCKM users with permissions to view the SAP keys can export and download the keys metadata displayed on the CCKM GUI. Filter the keys metadata to display on the GUI based on your requirements, for example, all keys created in the past 12 months, all keys residing in a specific SAP group, and all keys based on a specific algorithm. You can then download the filtered metadata as a CSV file.

To download the SAP keys metadata visible:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > SAP**. The SAP Keys page is displayed.
3. Filter the keys metadata based on your requirements.

Use the Search box to filter the keys. To further filter the metadata, show/hide the desired columns and use the column-level filters.

4. Click **Download as CSV**.

The filtered SAP keys metadata is downloaded as a CSV file.

SAP HYOK Resources

SAP Data Custodian KMS supports a customer-managed keystore that allows you to create HYOK keys residing in your external key manager. You manage the full lifecycle of HYOK keys within your external key manager. SAP Data Custodian KMS does not have any control over these keys.

Your external key manager handles crypto operations within its secure enclave, using a network endpoint. SAP KMS forwards these requests to your external key manager, where the key material remains protected.

Prerequisites

Before integrating CipherTrust Manager as your external key manager:

1. Ensure that CipherTrust Manager is up and running. Activate and install the CCKM license. Refer to the [CipherTrust Manager Deployment Guide](#) and [Licensing](#) for details.
2. Ensure CipherTrust Manager has a valid publicly accessible hostname and a trusted third-party CA certificate installed on its web interface.
3. Add the SAP Cloud Root CA as both an external CA and an external trusted CA on CipherTrust Manager:
 - a. Download the SAP Cloud Root CA.crt file from the Download section of the [SAP Trust Center Services](#) page.
 - b. Rename the extension of the file from CRT to PEM.
 - c. Add the SAP Cloud Root CA.pem file as an external CA on CipherTrust Manager. Refer to [Add an external CA](#).
 - d. Add the SAP Cloud Root CA.pem file as an external trusted CA on CipherTrust Manager. Refer to [Add an external trusted CA](#).

After ensuring the prerequisites, perform the steps described in the subsequent sections.

Create and download the SAP certificate

1. Log in to your SAP Data Custodian system.
 2. Open the **Key Management** menu.
 3. Select **Keystore Configuration**.
 4. Select **Thales CipherTrust Manager** from the Customer Managed Keystore tab.
-

-
5. Click **Certificate Auth..**
 6. Click **Create**. A new certificate will be generated.
 7. Select the new certificate.
 8. Select **Actions**.
 9. Select **Download**.
 10. Copy the **SAP Certificate ID**.

Create a keystore

Create a keystore on the CipherTrust Manager. Refer to [Creating SAP HYOK Keystores](#) for details. After the keystore is created successfully, copy the keystore URL.

Create a group for HYOK in SAP

You need to create a group for HYOK in SAP for CipherTrust Manager. To create a group.

1. Log in to SAP Data Custodian tenant.
2. Open the **Dashboard**.
3. Select the **Key Management Service** tab.
4. Select an application context.
5. Click **Create Group**.
6. Complete the Group Details section.
7. Open the **Dashboard** again.
8. Complete the Keystore Selection section.
 - a. Select **Customer Managed Keystore (HYOK)**.
 - b. Select **Thales CipherTrust Manager** from the **Keystore** drop-down menu.
 - c. Review Authentication Method section, auto populates to **Certificate**.
9. Select the **Key Management Service** tab again.
10. Complete the Configure Authentication Method section.
 - a. Enter Thales CipherTrust Manager keystore URL in the **Keystore URL** field. You copied it above while creating a SAP keystore.

-
- b. Enter the SAP Certificate ID in the **Certificate ID** field. You copied it above while creating and downloading the certificate.

11. Click **Review**.
12. Review the group details.
13. Click **Create**.

Create an endpoint (key)

Create an endpoint on the CipherTrust Manager. Refer to [Creating SAP HYOK Endpoints](#) for details. After the endpoint is created successfully, copy the globally unique identifier of the endpoint.

Register the key for HYOK in SAP

You need to register the key for HYOK in SAP for CipherTrust Manager. To register the key.

1. Log in to SAP Data Custodian tenant.
2. Open the **Dashboard**.
3. Select the **Key Management Service** tab.
4. Select an application context.
5. Select the HYOK group where the endpoint will be registered.
6. Click the **Keys** tab.
7. Click **Register New Key**.
8. Complete the Key Details section.
 - a. Select the **Key ID**.
 - b. Enter the unique identifier of the endpoint that you created above.
9. Click **Review**.
10. Review key details.
11. Click **Register**.

What's next?

You can manage keystores and endpoints on the CipherTrust Manager.

- [Managing SAP HYOK Keystores](#)

- [Managing SAP HYOK Endpoints](#)
- [Scheduling Operations](#)

Licensing

Thales provides encryption of SAP HYOK keystores through CCKM licensing models. SAP HYOK consumes one CCKM license (cloud unit) per keystore.

The number of SAP HYOK keystores that can be added to the CipherTrust Manager is limited by the number of licenses purchased. For example, if a 1-year license is purchased for 10 cloud units, then only 10 SAP HYOK keystores can be added to the CipherTrust Manager.

An archived or deleted SAP HYOK keystore releases one cloud unit of the CCKM license. It means, you can add another AWS account, Azure subscription, Google Workspace, or SAP HYOK keystore to the CipherTrust Manager. A recovered, enabled, or disabled SAP HYOK keystore consumes one cloud unit of the CCKM license.

Refer to [CCKM Licensing](#) for details.

SAP HYOK endpoints

This section describes how to manage the SAP HYOK endpoints.

Caution

Modify endpoints only through the CipherTrust Cloud Key Management (CCKM) GUI, API, or CLI. Do not modify the KEK using the **Keys** menu, the `ksctl keys` command, or the `/v1/vault/keys2` endpoint. Such modifications might make the KEK unavailable to the SAP HYOK service.

Creating SAP HYOK endpoints

The creation of an endpoint requires SAP HYOK keystores. So, before creating an SAP HYOK endpoint, ensure that you've already [created an SAP HYOK keystore](#).

To create an SAP HYOK endpoint:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed.
3. Click **Create Endpoint**. The General Info tab of the Create Endpoint screen is displayed.

General Info

1. Enter **Name**.
2. (Optional) Enter **Description**.
3. Click **Next**. The Key Material tab is displayed.

Key Material

1. Select a **Method**, the options are **Create New Key** and **Select Existing Key**.

Click the desired tab to view the instructions.

Create New Key

- a. Click **Create New Key**.
- b. Enter the **Source Key Name**.
- c. Select **Key Type**, the options are **AES** or **RSA**.
- d. Select the **Key Size** based on the key type:
 - For the **AES** key type, the option is 256.
 - For the **RSA** key type, the options are 3072 and 4096.
- e. Select the **Key Usage** based on the key type. You need to select at least one key usage.
 - For the **AES** key type, the options are Encrypt and Decrypt.
 - For the **RSA** key type, the options are Encrypt, Decrypt, Sign, and Verify.

Copy Existing Key

- a. Click **Copy Existing Key**.
 - b. Select **Key Type**, the options are **AES** or **RSA**.
 - c. Select a **Domain**.
 - d. Select the **Key Size** based on the key type:
 - For the **AES** key type, the option is 256.
 - For the **RSA** key type, the options are 3072 and 4096.
 - e. Select a **Source Key** from the list.
2. Click **Next**. The Select Keystore tab is displayed.

Select Keystore

1. Select a **Keystore** from the list.
2. Click **Next**. The Add Schedule tab is displayed.

Add Schedule

1. (Optional) Select a **Key Rotation Schedule** from the list.
2. Click **Next**. The Review and Add tab is displayed.

Review and Add

This screen shows the endpoint details that you have provided. These details are divided into GENERAL INFO, KEY MATERIAL, SELECT KEYSTORE, and ADD SCHEDULE sections.

Before creating the endpoint, review all details. After the endpoint is added, certain features will no longer be editable.

1. Review the endpoint details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the GENERAL INFO, KEY MATERIAL, SELECT KEYSTORE, and ADD SCHEDULE sections, and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Endpoint**. A success message is displayed on the screen.
3. Click **Close**. The newly created endpoint is displayed in the list of SAP HYOK endpoints.

Note

If you are planning to use this endpoint to register a master/primary key on SAP Data Custodian, please select the key usage as per SAP app requirements. Most SAP applications require master/primary key to have only encrypt and decrypt key usage.

Viewing SAP HYOK endpoints

To view the list of the available SAP HYOK endpoints:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed. The tab displays the following details:

Column	Description
Key Name	Name of the SAP key.
Key ID	ID of the SAP key.
Keystore Name	Name of the SAP keystore.
Blocked	Indicates whether the endpoint is blocked or unblocked.
Algorithm	Algorithm of the SAP key. The algorithm can be AES

Column	Description
	and RSA.
Size	Size of the SAP Key.
Version	Version of the SAP key.
State	State of the key. The state can be Enabled or Disabled.
SAP Tenant ID	ID of the SAP tenant.
Description	Description of the endpoint.
Creation At	Time when the endpoint is created.
Updated At	Time when the endpoint is updated.

To view the custom columns, click the Customize View () icon, select the desired option, and click **OK**.

Viewing and editing details of an SAP HYOK endpoint

To view or edit the details of an SAP HYOK endpoint:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed. The list of added SAP HYOK endpoints is displayed.
3. Click the **Name** link of the desired SAP HYOK endpoint.

Alternatively, click the overflow icon () corresponding to the desired SAP HYOK endpoint, and click **View/Edit**. The SAP HYOK Endpoints page is displayed.

The SAP HYOK Endpoints page shows additional details of the selected endpoint under the VERSIONS, GENERAL INFORMATION, and KEY SCHEDULE sections. Expand each section to view and edit their details.

VERSIONS

- a. Expand the VERSIONS section. The list of versions is displayed.
- b. (Optional) You can add a version to the key, refer to [Rotating an SAP HYOK Endpoint](#).

GENERAL INFORMATION

- a. Expand the GENERAL INFORMATION section.
- b. (Optional) Update the name of the SAP HYOK endpoint in the **Name** field.
- c. (Optional) Update the description of the SAP HYOK endpoint in the **Description** field.

-
- d. Click **Update**.

KEY SCHEDULE

- a. Select a **Rotation** from the drop-down list.
- b. Click **Update**.

Rotating an SAP HYOK endpoint

After rotating the endpoint, the globally unique external key ID will remain the same and a new version of the endpoint will be created which points to a new source endpoint.

To rotate an SAP HYOK Endpoint:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed. The list of added SAP HYOK endpoints is displayed.
3. Click the overflow icon () corresponding to the desired SAP HYOK endpoint.
4. Click **Rotate**. The Select Material Origin tab of the Add Version screen is displayed.
5. Select a **Key Type** to rotate the SAP HYOK endpoint. The options are **Create/Upload New Key Material** and **Clone Existing Key Material**.
6. Click the desired tab to view the instructions.

Create/Upload New Key Material

- a. Select the **Create/Upload New Key Material** option.
- b. Click **Next**. The Create Source Key and Add Version tab is displayed.
- c. Enter **Source Key Name**.
- d. Select the **Key Usage** based on the key type. You need to select at least one key usage.
 - For the **AES** key type, the options are Encrypt and Decrypt.
 - For the **RSA** key type, the options are Encrypt, Decrypt, Sign, and Verify.
- e. Click **Add Version**.

Clone Existing Key Material

- a. Select the **Clone Existing Key Material** option.
- b. Click **Next**. The Select Source Key and Add Version tab is displayed

-
- c. Select a **Source Key** from the list.
 - d. Click **Add Version**.

The version is added successfully.

Disabling an SAP HYOK endpoint

You can disable an endpoint to prevent it from being used for cryptographic operations. To disable an SAP HYOK endpoint:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed. The list of added SAP HYOK endpoints is displayed.
3. Click the overflow icon () corresponding to the desired SAP HYOK endpoint.
4. Click **Disable**. The Disable Endpoint dialog box is displayed.
5. Click **Disable** to confirm the action.

The endpoint is disabled successfully. The state of the endpoint changes to disabled.

Enabling an SAP HYOK endpoint

You can enable an endpoint to allow it to be used for cryptographic operations. To enable an SAP HYOK endpoint:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed. The list of added SAP HYOK endpoints is displayed.
3. Click the overflow icon () corresponding to the desired SAP HYOK endpoint.
4. Click **Enable**. The Enable Endpoint dialog box is displayed.
5. Click **Enable** to confirm the action.

The endpoint is enabled successfully. The state of the endpoint changes to enabled.

Blocking an SAP HYOK endpoint

You can block the access to a specific endpoint. To block an SAP HYOK endpoint:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed. The list of added SAP HYOK endpoints is displayed.

-
3. Click the overflow icon () corresponding to the desired SAP HYOK endpoint.
 4. Click **Block**. The Block Endpoint dialog box is displayed.
 5. Click **Block** to confirm the action.

The endpoint is blocked successfully.

Unblocking an SAP HYOK endpoint

You can unblock the access to a specific endpoint. To block an SAP HYOK endpoint:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed. The list of added SAP HYOK endpoints is displayed.
3. Click the overflow icon () corresponding to the desired SAP HYOK endpoint.
4. Click **Unblock**. The Unblock Endpoint dialog box is displayed.
5. Click **Unblock** to confirm the action.

The endpoint is unblocked successfully.

Archiving an SAP HYOK endpoint

Archiving is an intermediate state for deleting an endpoint, you need to archive an endpoint to make it deletable. To archive an SAP HYOK Endpoint:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed. The list of added SAP HYOK endpoints is displayed.
3. Click the overflow icon () corresponding to the desired SAP HYOK endpoint.
4. Click **Archive**. The Archive Endpoint dialog box is displayed.
5. Select **I wish to archive this endpoint**.
6. Click **Archive**.

You will see a message that the endpoint is archived successfully. The state of the endpoint changes to archived.

Recovering an SAP HYOK endpoint

You can recover an archived endpoint. To recover an SAP HYOK Endpoint:

1. Open the **Cloud Key Manager** application.

2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed. The list of added SAP HYOK endpoints is displayed.
3. Click the overflow icon () corresponding to the desired SAP HYOK endpoint.
4. Click **Recover**. The Recover Endpoint dialog box is displayed.
5. Click **Recover**.

You will see a message that the endpoint is recovered successfully.

Deleting an SAP HYOK endpoint

You can delete the archived endpoint from the CCKM. To delete an SAP HYOK Endpoint:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The Endpoints tab of the SAP HYOK page is displayed. The list of added SAP HYOK endpoints is displayed.
3. Click the overflow icon () corresponding to the desired SAP HYOK endpoint.
4. Click **Delete**. The Delete Endpoint dialog box is displayed.
5. Select **I wish to delete this endpoint**.
6. Click **Delete**.

You will see a message that the endpoint is deleted successfully.

SAP HYOK keystores

This section describes how to manage the SAP HYOK keystores.

Creating SAP HYOK keystores

Creation of a keystore requires an SAP tenant ID, keystore URL hostname, and SAP certificate. To create an SAP HYOK keystore:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The SAP HYOK page is displayed.
3. Click the **SAP Keystores** tab.
4. Click **Create Keystore**. The General Info tab of the Create SAP HYOK Keystore screen is displayed.

General Info

1. Enter **Name**.

2. (Optional) Enter **Description**.
3. Click **Next**. The Configure Keystore tab is displayed.

Configure Keystore

1. Enter **SAP Tenant ID**.
2. Enter **Keystore URL Hostname**.

Note

Enable or disable audit recording of successful operations within an external keystore. The default value is false.

3. (Optional) Enable **SAP Enable success audit events**.
4. Add a **SAP Certificate**. You can use the following options to add the SAP Certificate.

File Upload: Select and upload the certificate (in PEM format).

Text: Select and paste the certificate content.

Note

You can [create and download the SAP certificate](#) from the SAP Data Custodian portal.

5. Click **Next**. The Review and Create tab is displayed.

Review and Create

This screen shows the key details that you have provided. These details are divided into GENERAL INFO and CONFIGURE KEYSTORE sections.

Before creating the keystore, review all details. After the keystore is added, certain features will no longer be editable.

1. Review the keystore details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the GENERAL INFO and CONFIGURE KEYSTORE sections, and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Create**. A success message is displayed on the screen.

3. Click **Close**. The newly created keystore is displayed in the list of SAP HYOK keystores.

Viewing SAP HYOK keystores

To view the list of the available SAP HYOK Keystores:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The SAP HYOK page is displayed.
3. Click the **SAP Keystores** tab. The list of added SAP HYOK keystores is displayed. The tab displays the following details:

Column	Description
Name	Name of the SAP HYOK keystore.
Keystore URL Hostname	Hostname for the SAP HYOK keystore URL.
Keystore URL	URL of the keystore.
Status	Status of the keystore. The status can be: • Active • Inactive
Block	Indicates whether the keystore is blocked or unblocked.
Tenant	ID of the SAP tenant.
Creation Date	Time when the keystore is created.
Last Updated	Time when the keystore is updated.
Description	Description of the keystore.

To view the custom columns, click the Customize View () icon, select the desired option, and click **OK**.

Viewing and editing details of an SAP HYOK keystore

To view or edit the details of an SAP HYOK keystore:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The SAP HYOK page is displayed.
3. Click the **SAP Keystores** tab. The list of added SAP HYOK keystores is displayed.
4. Click the **Name** link of the desired SAP HYOK keystore.

Alternatively, click the overflow icon () corresponding to the desired SAP HYOK keystore, and click **View/**

Edit. The SAP HYOK Keystore page is displayed.

The SAP HYOK Keystore page shows additional details of the selected keystore under the **ENDPOINT KEYS**, **GENERAL INFORMATION**, **KEYSTORE CONFIGURATION**, and **ACCESS CONTROL** sections. Expand each section to view and edit their details.

ENDPOINT KEYS

- a. Expand the **ENDPOINT KEYS** section.
- b. The list of endpoint associated with the keystore is displayed. To manage these endpoints, refer to [SAP HYOK Endpoints](#).

GENERAL INFORMATION

- a. Expand the **GENERAL INFORMATION** section.
- b. (Optional) Update the name of the SAP HYOK keystore in the **Name** field.
- c. (Optional) Update the description of the SAP HYOK keystore in the **Description** field.
- d. Click **Update**.

KEYSTORE CONFIGURATION

- a. Expand the **KEYSTORE CONFIGURATION** section.
- b. (Optional) Enter a new **Keystore URL Hostname** to use a different URL hostname.
- c. (Optional) Enable or disable **SAP Enable success audit events**.

Note

Enable or disable audit recording of successful operations within an external keystore. The default value is false.

- d. (Optional) Update the **SAP Certificate**. You can use the following options to update the SAP Certificate.

File Upload: Select and upload the certificate (in PEM format).

Text: Select and paste the certificate content.

Note

Create and download this certificate from SAP Data Custodian Portal Key Management Service > Configuration > Thales CipherTrust Manager.

e. Click **Update**.

ACCESS CONTROL

Refer to [Managing User Permissions on SAP HYOK Keystore](#) for details.

Managing User permissions on SAP HYOK keystore

To work with the external resources, users/groups must have the minimum set of permissions that allow them to use the SAP HYOK keystores.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform operations on SAP HYOK keystores.

Users with the following characteristics can perform operations on SAP HYOK keystores:

- Users in the **CCKM Admins** group
- Users in the **Admin** group
- Users who are administrators for a **domain**
- Users who are in the **CCKM Users** group and which have had a CCKM Admin assign permissions through the UI or the `/v1/cckm/sap/ekm/keystores/{id}/update-acls` endpoint in the REST API.

Adding permissions for a User/Group

To add permissions for a user/group:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The SAP HYOK page is displayed.
3. Click the **SAP Keystores** tab. The list of added SAP HYOK keystores is displayed.
4. Click the **Name** link of the desired SAP HYOK keystore.

Alternatively, click the overflow icon () corresponding to the desired SAP HYOK keystore, and click **View/Edit**. The SAP HYOK Keystore page is displayed.

5. Expand the **ACCESS CONTROL** section.
6. Click **Assign User/Group**. The **Assign User/Group** dialog box is displayed.
7. Select the desired user or group from the **User/Group** drop-down list.
8. Click **Save**.

The newly added user/group is displayed under **Name** in the **ACCESS CONTROL** section.

CCKM allows the following operations on the SAP HYOK keystores:

- View Keys, Add Key, Delete Key.
- Refresh Domain.

You can now grant additional permissions to the user/group, as appropriate. Refer to [Granting Permission to Perform an Operation](#) for details.

Granting permission to perform an operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. In the **ACCESS CONTROL** section, select the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

To revoke permissions from a user/group, refer to [Removing a Permission](#) for details.

Removing a permission

To remove a permission assigned to a user or group:

1. In the **ACCESS CONTROL** section, clear the check box under the desired operation corresponding to the desired users or groups.
2. Click **Update**.

A success message is displayed on the screen.

Removing permission from a User/Group

To remove current permissions assigned to the user/group:

1. In the **ACCESS CONTROL** section, under **Unassign**, click the **X** button corresponding to the desired user/group.

2. On the **Remove User / Remove Group** screen, click **Remove**.

Note

Removing this user/group will remove all permissions currently assigned to the user/group.

3. Click **Remove** to confirm the action. To cancel the action, click **Keep It**.

A success message is displayed on the screen.

Blocking an SAP HYOK keystore

You can block a keystore to deny all proxy API requests from both the keystore and its endpoints. To block an SAP HYOK keystore:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The SAP HYOK page is displayed.
3. Click the **SAP Keystores** tab. The list of added SAP HYOK keystores is displayed.
4. Click the overflow icon () corresponding to the desired SAP HYOK keystore.
5. Click **Block**. The Block Keystore dialog box is displayed.
6. Click **Block** to confirm the action.

The keystore is blocked successfully. The state of the keystore changes to Blocked.

Unblocking an SAP HYOK keystore

You can block a keystore to accept all proxy API requests from both the keystore and its endpoints. To block an SAP HYOK keystore:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The SAP HYOK page is displayed.
3. Click the **SAP Keystores** tab. The list of added SAP HYOK keystores is displayed.
4. Click the overflow icon () corresponding to the desired SAP HYOK keystore.
5. Click **Unblock**. The Unblock Keystore dialog box is displayed.
6. Click **Unblock** to confirm the action.

The keystore is unblocked successfully. The state of the keystore changes to Unblocked.

Deleting an SAP HYOK keystore

You can delete an archived keystore that has no endpoints from the CCKM. To delete an SAP HYOK keystore:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The SAP HYOK page is displayed.
3. Click the **SAP Keystores** tab. The list of added SAP HYOK keystores is displayed.
4. Click the overflow icon () corresponding to the desired SAP HYOK keystore.
5. Click **Delete**. The Delete Keystore dialog box is displayed.

Note

The keystore can only be deleted if it does not contain any SAP HYOK Endpoints.

6. Select **I wish to delete key material**.
7. Click **Delete**.

You will see a message that the keystore is deleted successfully.

Archiving an SAP HYOK keystore

Archiving transitions a keystore to an intermediate state, a prerequisite for its deletion. When a keystore is archived, its associated endpoints are also archived, and a CCKM license is released.

To archive an SAP HYOK Endpoint:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The SAP HYOK page is displayed.
3. Click the **SAP Keystores** tab. The list of added SAP HYOK keystores is displayed.
4. Click the overflow icon () corresponding to the desired SAP HYOK keystore.
5. Click **Archive**. The Archive Keystore dialog box is displayed.
6. Select **I wish to archive this keystore**.
7. Click **Archive**.

You will see a message that the keystore is archived successfully. The state of the keystore changes to archived.

Recovering an SAP HYOK keystore

You can recover an archived keystore. To recover an SAP HYOK Keystore:

Note

Recovering an archived SAP keystore consumes a CCKM license unit. This process does not automatically recover its previously associated endpoints; you must recover them individually. Refer to [Recovering an SAP HYOK endpoint](#) for details.

To recover an SAP HYOK Keystore:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Services > SAP HYOK**. The SAP HYOK page is displayed.
3. Click the **SAP Keystores** tab. The list of added SAP HYOK keystores is displayed.
4. Click the overflow icon () corresponding to the desired SAP HYOK keystore.
5. Click **Recover**. The Recover Keystore dialog box is displayed.
6. Click **Recover**.

You will see a message that the keystore is recovered successfully.

Scheduling Operations

CCKM allows you to periodically rotate keys in the background. To rotate keys, you need to create schedule configuration. A schedule configuration defines when to run a rotate job. You can use either basic format or cron format to specify the time when the rotate job will run.

Specify basic format in the following order:

[Repeats, at]

Where,

- Repeats: frequency of the scheduler. Possible values are daily, weekly, monthly, and yearly.
- at: specific time at which the refresh operation will be performed. Possible value lies between 12:00-11:00 AM/PM UTC.

Specify cron format in the following order:

"* * * * *"

These five values indicate when the job should be executed. These values are mandatory and must be specified in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

Field	Allowed values	Allowed special characters
Minute	0-59	* / , -
Hour	0-23	* / , -
Day of month	1-31	* / , - ?
Month	1-12 or JAN-DEC	* / , -
Day of week	0-6 or SUN-SAT	* / , - ?

Adding Key Rotation Schedule

A key rotation schedule replaces selected keys with new cryptographic keys at the specified time.

To add key rotation schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. **Select Schedule Type as Key Rotation.**
5. Click **Next**.
6. On the **General Info** screen, enter or select the following details:
 - Enter a unique **Scheduler Name**.
 - From the **Cloud Name** drop-down list, select **SAP HYOK**.
 - (Optional) Add a **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. Set the values on the **Schedule Config** screen.
 - a. Configure **DURATION** for the scheduler. Specify the following values:
 - **Schedule Starts**
 - **Schedule Ends**

If you select the **Never** check box, the scheduler configuration will never expire, and the Schedule Ends field becomes unavailable.

- b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either the **Basic** or the **Raw (Cron)** format to specify the value.

8. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Viewing/Editing Schedules

To view/edit a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The Schedules page displays the following details:

Field	Description
Name	Unique name of the scheduler configuration.
Schedule Type	Type of the schedule. Possible types are: • Key Rotation • Key Refresh • Add KMS Container (GCP and Azure only) For Luna HSM, only Key Refresh is supported.
Cloud Name	Name of the cloud.
Last Modified	Time when the schedule is modified.
Frequency	Frequency of the scheduler configuration.
Start Date	Creation time of the scheduler configuration.
End Date	Expiry time of the scheduler configuration.
Status	Status of the scheduler configuration. Possible values are: • Enabled • Disabled

3. Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.
4. Edit or configure the following fields and click **Update**:
 - Description.
 - Status of the scheduler configuration.
 - Scheduler configuration parameters such as duration, frequency, and expiry time of the key.

The Schedules page also contains a section named **JOB HISTORY** that contains all the related information about a schedule such as **Run Date**, **Job ID**, **Status**, and **Errors**.

Disabling Schedules

To disable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Disable**. The Disable Schedule message is displayed.
4. Click **Disable Schedule**.

A message **Successfully disabled the schedule** is displayed on the screen.

Enabling Schedules

To enable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Enable**. The Enable Schedule message is displayed.
4. Click **Enable Schedule**.

A message **Successfully enabled the schedule** is displayed on the screen.

Manually Running Schedules

To manually run a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Run Now**.

The Run Now schedule is started in the background. A message **Schedule is running now. It will take a few seconds to finish.** is displayed on the screen. After the schedule is run successfully, a message **Successfully finished running the schedule** is displayed.

Deleting Schedules

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.

3. Click the overflow icon () corresponding to the desired schedule and click **Delete Schedule**. The Delete Schedule message is displayed.
4. Click **Delete**.

A message **Successfully deleted schedule** is displayed on the screen.

Salesforce Resources

CCKM supports the Salesforce and Salesforce Sandbox clouds. The Salesforce Sandbox is a copy of your production Salesforce environment used primarily for testing, development, or training. In this document, the information about Salesforce is applicable to both Salesforce and Salesforce Sandbox unless stated otherwise.

Your login credentials for Salesforce and Salesforce Sandbox are unique to each cloud. If you are configuring the setup for Salesforce Sandbox, ensure to configure within the Salesforce Sandbox cloud.

Manage the Salesforce organizations and tenant secrets stored in your Salesforce account using CCKM.

Note

The instructions provided in this section are specific to the Salesforce Classic interface. If you are using Salesforce Lightning, you will notice a different interface. So, refer to the corresponding sections of the Salesforce Lightning interface documentation to complete the prerequisites.

Note

- Salesforce's Shield Platform Encryption Service must be enabled.
- To generate Salesforce reports, **Event Monitoring** (a paid feature) must be enabled in the Salesforce cloud.
- Salesforce provides two profile interfaces: *Original Profile Interface* and *Enhanced Profile User Interface*. Steps in this section are based on the *Enhanced Profile User Interface*. To enable the Enhanced Profile User Interface, refer to [Enable the Enhanced Profile User Interface](#) in the Salesforce documentation.

Prerequisites

Access to Salesforce resources can be configured using profiles and permission sets. Depending on your requirements, use either method. However, permission sets is the recommended method to configure access. After configuring the access, you can add a Salesforce connection on the CipherTrust Manager.

Before you can manage Salesforce resources on CCKM:

1. [Configure Access to Salesforce](#)

-
2. Configure Advanced Settings on Salesforce
 3. Add Salesforce Connection on CipherTrust Manager

Configure Access to Salesforce

Click a tab to view details.

Using Permission Set

1. Create a New Profile

Create a new profile or use an existing one. Creating a new profile is recommended. This section describes steps to create a new profile.

1. Log on to your Salesforce cloud account.
2. At the top right, click **Setup**.
3. In the left pane, under **Administer**, click **Manage Users > Profiles**. The Profiles page shows the list of existing profiles.
4. Click **New profile**. The Clone Profile page is displayed.
 - a. From the **Existing Profile** drop-down list, select **Minimum Access - Salesforce**.
 - b. Enter **Profile Name**, for example, **no_permission_profile**.
 - c. Click **Save**.

The profile is created.

5. Remove the assigned permissions.

The GUI slightly differs for Salesforce production environment and Salesforce Sandbox.

Click a tab to view details.

Salesforce Sandbox

- a. Scroll down to the **System** section.
- b. Click the **System Permissions** link. The list of permissions assigned to the profile is displayed. These permissions are based on the profile you selected from the **Existing Profile** drop-down list.
- c. Next to **System Permissions**, click **Edit**. The edit view of the profile is displayed.
- d. Clear all the set permissions.

Note

A few permissions are enabled and read-only. They cannot be cleared, just ignore them.

- e. Scroll up the page.
- f. Click **Save**. The Permission Changes Confirmation dialog box is displayed.

Salesforce (Production Environment)

- a. Next to **Profile Detail**, click **Edit**. The edit view of the profile is displayed.
- b. Scroll down to the **Administrative Permissions** section. The list of permissions assigned to the profile is displayed. These permissions are based on the profile you selected from the **Existing Profile** drop-down list.
- c. Clear all the set permissions.

Note

A few permissions are enabled and read-only. They cannot be cleared, just ignore them.

- d. Scroll down to the **General User Permissions** section.
 - e. Clear all the set permissions.
 - f. Scroll up the page.
6. Click **Save** to confirm the changes.

2. Create a New User

Create a new user or edit an existing one to change the profile. This section describes steps to create a new user.

1. In the left pane, under **Administer**, click **Manage Users > Users**. The All Users page is displayed.
2. In the middle of the page, click **New User**. The New User page is displayed.
3. Specify the user details such as first name, last name, alias, email address, username, and nickname.
4. Select the **User License** that has the **Manage Encryption Keys** and **View Event Log Files** system permissions.
5. From the **Profile** drop-down list, select **no_permission_profile** (the profile created in 1. [Create a New](#)

Profile).

6. Click **Save**.

The user is created.

3. Create a New Permission Set

1. In the left pane, under **Administer**, click **Manage Users > Permission Sets**. The Permission Sets page displays the list of existing permission sets.
2. Click **New**. The Create page is displayed.
3. Enter permission set information such as Label, API Name, and Description.
4. Under **Select the type of users who will use this permission set**, make sure that **None** is selected in the **License** drop-down list.
5. Click **Save**.

The permission set is created.

4. Add Permissions to the Permission Set

1. Scroll down to the **System** section.
2. Click the **System Permissions** link.
3. Next to **System Permissions**, click **Edit**. The edit view of the permissions set is displayed.
4. Make sure the following permissions are selected:
 - **API Enabled**: To access any Salesforce.com API.
 - **Manage Encryption Keys**: To manage tenant secrets.
 - **Modify Metadata Through Metadata API Functions**: To synchronize certificates.
 - **View Event Log Files**: To run reports.
 - **View Roles and Role Hierarchy**: To view roles and role hierarchy.
 - **View Setup and Configuration**: To get list of SFDC organizations and their details.
5. Scroll up the page.
6. Click **Save**. The Permission Changes Confirmation dialog box is displayed.
7. Click **Save** to confirm the changes.

5. Assign the User to the Permission Set

1. At the top of the page, click **Manage Assignments**. The Assigned Users page of the permission set is displayed.
2. Click **Add Assignments**. The All Users page displays the list of existing users.
3. Select the user created in 2. [Create a New User](#).
4. Click **Next**.
5. Click **Assign**. The permission set is assigned to the selected user.
6. Click **Done**.

The user assigned to the permission set is displayed in the list of assigned users.

6. Create a Connected App

Create a new Connected App or use an existing one. If you are using an existing App, then remove any profiles currently assigned to the Connected App and assign the permission set created in 3. [Create a New Permission Set](#). This section describes steps to create a new Connected App.

1. In the left pane, under **Build**, click **Create > Apps**. The Apps page shows the list of available apps. On this page, you can create a new Connected App.
2. Navigate to the **Connected Apps** section. Scroll down the page if needed.
3. Click **New**. The New Connected App page is displayed.
4. Specify the **Basic Information**, for example, the app's name, logo, and contact information. The mandatory fields are:
 - **Connected App Name**: A unique name for your Connected App. The name can only contain underscores and alphanumeric characters. The name must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
 - **API Name**: A name for the connected API. It is populated automatically with the Connected App Name you specified above. Change the API name if you want.
 - **Contact Email**: A valid email address.

You can also specify additional (optional) details such as contact phone number, URLs, and description.

5. Under **API (Enable OAuth Settings)**:
 - a. Select **Enable OAuth Settings**. More fields are displayed.
 - b. (Optional but recommended. Applicable to certificate-based authentication.) Select **Use digital signatures**. This specifies that a digital certificate is used to make API calls to Salesforce. The digital

certificate is the public key of an RSA key pair (a public-private key pair) created on the CipherTrust Manager.

Click **Choose File** to upload the desired digital certificate (public key). This certificate can be downloaded on the CipherTrust Manager when a connection to the Salesforce cloud is added. Refer to [Connection Management](#) for details.

- c. Under **Selected OAuth Scopes**, select the OAuth scopes to apply to the Connected App. OAuth scopes define permissions for the Connected App that are granted as tokens after the app is authorized. The OAuth token name is in parentheses.

To select an OAuth scope:

- a. From the **Available OAuth Scopes** list, select the desired OAuth scope.

CipherTrust Manager requires the following OAuth scopes:

- Access the identity URL service (id, profile, email, address, phone)
- Manage user data via APIs (api)
- Perform requests at any time (refresh_token, offline_access)

Use the **Ctrl** key to select multiple scopes.

- b. Click the **Add** button (between Available OAuth Scopes and Selected OAuth Scopes sections). The selected OAuth scope moves under the Selected OAuth Scopes section.

It is recommended to not select the **Full access (full)** scope.

- d. Make sure that **Require Secret for Web Server Flow** is selected. This ensures the app's client secret is required in exchange for an access token.
- e. Clear **Require Secret for Refresh Token Flow**.

6. Click **Save**. Changes can take up to 10 minutes to take effect.

7. Click **Continue**.

When the Connected App is created, the app details are displayed. Note down the Consumer Key, Consumer Secret, and Salesforce username with its password. This information is required when adding a Salesforce connection to the CipherTrust Manager.

To gather this information any time:

1. Log on to your Salesforce cloud account.
2. In the left pane, under **Build**, click **Create > Apps**. The Apps page shows available apps.
3. Navigate to the **Connected Apps** section. Scroll down the page if needed.

-
4. Click your Connected App to view its details. The Consumer Key and Consumer Secret are displayed under the API (Enable OAuth Settings) section.

On the Add Connection page of the CipherTrust Manager GUI, the Salesforce Consumer Key and Consumer Secret are called Client ID and Client Secret respectively.

Client ID and Client Secret are required for Client Secret as the Authentication type. For Certificate as the Authentication type, only Username and Client ID are required.

7. Add OAuth Policies on Salesforce

Next, you need to specify who all can access the Connected App. This is done by adding appropriate OAuth policy to the Connected App on the Salesforce cloud.

1. In the left pane, under **Administer**, click **Manage Apps > Connected Apps**. The Connected Apps page shows the available apps.
2. Under **Master Label**, click your Connected App. The Connected App Detail page is displayed.
3. Click **Edit Policies**. The edit view of the Connected App is displayed.
4. Under the **OAuth Policies** section, from the **Permitted Users** drop-down list, select **Admin approved users are pre-authorized**.

Changing the OAuth policies can cause access deny issues for users currently using the app.

5. Click **OK** to confirm the action.
6. From the **IP Relaxation** drop-down list, select an appropriate option. Consult your Salesforce administrator for the appropriate option.
7. Select **Refresh token is valid until revoked**.
8. Click **Save**. You are returned to the Connected App Detail page.

8. Assign the Permission Set to the Connected App

1. Scroll down to the **Permission Sets** section.
2. Click **Manage Permission Sets**. The Application Permission Set Assignment page is displayed.
3. Select the permission set created in 3. [Create a New Permission Set](#).
4. Click **Save**.

The selected permission set is assigned to the Connected App.

Using Profile

1. Create a New Profile

1. Log on to your Salesforce cloud account.
2. At the top right, click **Setup**.
3. In the left pane, under **Administer**, click **Manage Users > Profiles**. The Profiles page shows the list of existing profiles.
4. Click **New profile**. The Clone Profile page is displayed.
5. From the **Existing Profile** drop-down list, select **Minimum Access - Salesforce**.
6. Enter **Profile Name**.
7. Click **Save**.

The profile is created.

2. Assign Permissions to the Profile

1. Edit the profile.

The GUI slightly differs for Salesforce production environment and Salesforce Sandbox.

Click a tab to view details.

Salesforce Sandbox

- a. Scroll down to the **System** section.
- b. Click the **System Permissions** link. The list of permissions assigned to the profile is displayed. These permissions are based on the profile you selected from the **Existing Profile** drop-down list.
- c. Next to **System Permissions**, click **Edit**. The edit view of the profile is displayed.
- d. Make sure the following permissions are selected:
 - **API Enabled**: To access any Salesforce.com API.
 - **Manage Encryption Keys**: To manage tenant secrets.
 - **Modify Metadata Through Metadata API Functions**: To synchronize certificates.
 - **View Event Log Files**: To run reports.
 - **View Roles and Role Hierarchy**: To view roles and role hierarchy.

- **View Setup and Configuration:** To get list of SFDC organizations and their details.

e. Scroll up the page.

f. Click **Save**. The Permission Changes Confirmation dialog box is displayed.

Salesforce (Production Environment)

a. Next to **Profile Detail**, click **Edit**. The edit view of the profile is displayed.

b. Scroll down to the **Administrative Permissions** section. The list of permissions assigned to the profile is displayed. These permissions are based on the profile you selected from the **Existing Profile** drop-down list.

c. Make sure the following permissions are selected:

- **API Enabled:** To access any Salesforce.com API.
- **Manage Encryption Keys:** To manage tenant secrets.
- **Modify Metadata Through Metadata API Functions:** To synchronize certificates.
- **View Roles and Role Hierarchy:** To view roles and role hierarchy.
- **View Setup and Configuration:** To get list of SFDC organizations and their details.

d. Under **General User Permissions**, select **View Event Log Files**. This is required to run reports.

e. Scroll up the page.

2. Click **Save** to confirm the changes.

3. Create a New User

Create a new user or edit an existing one to change the profile. This section describes steps to create a new user.

1. In the left pane, under **Administer**, click **Manage Users > Users**. The All Users page is displayed.
2. In the middle of the page, click **New User**. The New User page is displayed.
3. Specify the user details such as first name, last name, alias, email address, username, and nickname.
4. Select the **User License** that has the **Manage Encryption Keys** and **View Event Log Files** system permissions.
5. From the **Profile** drop-down list, select the profile created in [1. Create a New Profile](#).
6. Click **Save**.

The user is created.

4. Create a Connected App

1. In the left pane, under **Build**, click **Create > Apps**. The Apps page shows available apps. On this page, you can create a new connected app.
2. Navigate to the **Connected Apps** section. Scroll down the page if needed.
3. Click **New**. The New Connected App page is displayed.
4. Specify the **Basic Information**, for example, the app's name, logo, and contact information. The mandatory fields are:
 - **Connected App Name:** A unique name for your connected app. The name can only contain underscores and alphanumeric characters. The name must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
 - **API Name:** A name for the connected API. It is populated automatically with the Connected App Name you specified above. Change the API name if you want.
 - **Contact Email:** A valid email address.

You can also specify additional (optional) details such as contact phone number, URLs, and description.

5. Under **API (Enable OAuth Settings)**:
 - a. Select **Enable OAuth Settings**. More fields are displayed.
 - b. (Optional but recommended. Applicable to certificate-based authentication.) Select **Use digital signatures**. This specifies that a digital certificate is used to make API calls to Salesforce. The digital certificate is the public key of an RSA key pair (a public-private key pair) created on the CipherTrust Manager.

Click **Choose File** to upload the desired digital certificate (public key). This certificate can be downloaded on the CipherTrust Manager when a connection to the Salesforce cloud is added. Refer to [Connection Management](#) for details.

- c. Under **Selected OAuth Scopes**, select the OAuth scopes to apply to the connected app. OAuth scopes define permissions for the connected app that are granted as tokens after the app is authorized. The OAuth token name is in parentheses.

To select an OAuth scope:

- a. From the **Available OAuth Scopes** list, select the desired OAuth scope.

CipherTrust Manager requires the following OAuth scopes:

- Access the identity URL service (id, profile, email, address, phone)
- Manage user data via APIs (api)

-
- Perform requests at any time (refresh_token, offline_access)

Use the **Ctrl** key to select multiple scopes.

- b. Click the **Add** button (between Available OAuth Scopes and Selected OAuth Scopes sections). The selected OAuth scope moves under the Selected OAuth Scopes section.

It is recommended to not select the **Full access (full)** scope.

- d. Make sure that **Require Secret for Web Server Flow** is selected. This ensures that the app's client secret is required in exchange for an access token.

- e. Clear **Require Secret for Refresh Token Flow**.

6. Click **Save**. Changes can take up to 10 minutes to take effect.

7. Click **Continue**.

When the connected app is created, the app details are displayed. Note down the Consumer Key, Consumer Secret, and Salesforce username with its password. This information is required when adding a Salesforce connection to the CipherTrust Manager.

To gather this information any time:

1. Log on to your Salesforce cloud account.
2. In the left pane, under **Build**, click **Create > Apps**. The Apps page shows available apps.
3. Navigate to the **Connected Apps** section. Scroll down the page if needed.
4. Click your connected app to view its details. The Consumer Key and Consumer Secret are displayed under the API (Enable OAuth Settings) section.

On the Add Connection page of the CipherTrust Manager GUI, the Salesforce Consumer Key and Consumer Secret are called Client ID and Client Secret respectively.

Client ID and Client Secret are required for Client Secret as the Authentication type. For Certificate as the Authentication type, only Username and Client ID are required.

5. Add OAuth Policies on Salesforce

Next, you need to specify who all can access the Connected App. This is done by adding appropriate OAuth policy to the Connected App on the Salesforce cloud.

1. In the left pane, under **Administer**, click **Manage Apps > Connected Apps**. The Connected Apps page shows available apps.
 2. Under **Master Label**, click your Connected App.
 3. Click **Edit Policies**. The edit view of the Connected App is displayed.
 4. Under the **OAuth Policies** section, from the **Permitted Users** drop-down list, select **Admin approved users**
-

are pre-authorized.

Changing the OAuth policies can cause access deny issues for users currently using the app.

5. Click **OK** to confirm the action.
6. From the **IP Relaxation** drop-down list, select an appropriate option. Consult your Salesforce administrator for the appropriate option.
7. Select **Refresh token is valid until revoked**.
8. Click **Save**. You are returned to the Connected App Detail page.

6. Assign the Profile to the Connected App

Next, you need to select the appropriate profiles to choose which users have access to this application.

1. Scroll down to the **Profiles** section.
2. Click **Manage Profiles**. The Application Profile Assignment page is displayed.
3. Select the profile created in [1. Create a New Profile](#).
4. Click **Save**.

The selected profile is assigned to the Connected App.

Configure Advanced Settings on Salesforce

Additionally, configure the following settings on Salesforce (**Security Controls > Platform Encryption > Advanced Settings**).

Note

Note that a few settings might not be available depending on your Salesforce account.

- **Allow BYOK to Opt Out of Key Derivation:** Enable or disable the Salesforce key derivation process.
 - Turn ON to disable the Salesforce key derivation process and use your uploaded key material as the final encryption key. If this option is disabled, the BYOK upload will fail.
 - Turn OFF to use key derivation.
- **Allow Cache-Only Keys with BYOK:** Allow or deny the use of cache-only keys with BYOK.
 - Turn ON to use cache-only keys. This lets Salesforce fetch remotely-stored encryption keys from an endpoint that you provide.

-
- Turn OFF for *non* cache-only keys.
 - **Enable Replay Detection for Cache-Only Keys:** Replay detection protects your cache-only keys if a callout is fraudulently intercepted.
 - Turn ON to insert an auto-generated, unique marker called a request identifier into every callout. Any callout with missing or mismatched request identifiers is aborted by Salesforce.
 - Turn OFF to disable replay detection.

Add Salesforce Connection on CipherTrust Manager

Before you can add a Salesforce organization to the CCKM, a connection to your Salesforce account must already exist on the CipherTrust Manager. A CipherTrust Manager administrator manages connections to external resources on the **Access Management > Connections Management** page of the CipherTrust Manager GUI. Refer to [Connection Manager](#) for details.

If you need a connection with certificate-based authentication, specify Username and Client ID, select Certificate as the Authentication type, and generate and download the certificate. The downloaded certificate needs to be uploaded to the connected app on the Salesforce cloud. Refer to [Upload Public Certificate to Salesforce](#) for details.

After completing the prerequisites, you can view linked Salesforce organizations and manage linked tenant secrets on the CipherTrust Manager.

Refer to the following sections:

- [Managing Salesforce Organizations](#)
- [Managing Salesforce Tenant Secrets](#)
- [Scheduling Operations](#)
- [Managing Salesforce Reports](#)
- [Managing Salesforce Cache-Only Key Endpoints](#)
- [Managing Certificates from Salesforce Organizations](#)

Managing Salesforce Organizations

This section describes how to manage Salesforce organizations on CCKM. Before proceeding, a connection to your Salesforce account must exist on the CipherTrust Manager. Refer to [Connection Manager](#) for details.

After the connection is configured, you can add organizations to the CipherTrust Manager. Salesforce organizations can be added, viewed, modified, or deleted on the Salesforce Organizations page.

Each Salesforce organization consumes one [CCKM license cloud unit](#) when added to CCKM.

A Salesforce organization is needed to [upload tenant secrets](#), [upload cache-only keys](#), and [view Salesforce reports](#).

Adding an Existing Salesforce Organization

You can add existing organizations linked to a Salesforce connection to the CipherTrust Manager. An existing organization can only be added just once.

To add an existing Salesforce organization:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Salesforce Organizations**. The Salesforce Organizations page is displayed.
3. Click **Add Existing Organization**. The Add Existing Organization screen is displayed.
4. Select the desired **Connection** from the drop-down list. The list of existing organizations linked with the selected connection is displayed in the Organization field.
5. Select the desired **Organization**.
6. Click **Add**.

The selected organization is displayed on the Salesforce Organizations page. Now, you can manage the organization from CCKM on the CipherTrust Manager.

The organization is available to [upload tenant secrets](#), [associate to cache-only key endpoints](#), and [view Salesforce reports](#).

If the selected organization is already added to the CipherTrust Manager, an error message is displayed.

Viewing Salesforce Organizations

The Salesforce Organizations page shows the list of organizations added to the CipherTrust Manager. Search for the organizations by Name or Connection.

To view the list of organizations added to the CipherTrust Manager:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Salesforce Organizations**. The Salesforce Organizations page shows the list of organizations added to the CipherTrust Manager.

The page displays the following details:

Column	Description
Organization Name	Name of the organization.
Organization ID	ID of the Salesforce organization.
Organization Type	Type of the Salesforce organization.
Connection	Name of the Salesforce connection added to the

Column	Description
	CipherTrust Manager.
Cloud Name	Name of the Salesforce cloud. The cloud can be SFDC or Sandbox.
Last Refreshed	Time when the group was refreshed the last.

To view the custom columns, click the Customize View (⚙️) icon, select the desired option, and click **OK** to display the column.

Refreshing Organizations

Refreshing is the process to download keys created in Salesforce organizations to the CCKM. You can refresh keys from individual or all Salesforce organizations.

- [Refreshing Specific Organizations](#)
- [Refreshing All Organizations](#)

Refreshing Specific Organizations

To refresh an organization:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Salesforce Organizations**. The Salesforce Organizations page is displayed.
3. Click the overflow icon (⋮) corresponding to the desired organization and click **Refresh Now**.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Salesforce > Salesforce Keys** page. Refer to [Viewing Salesforce Keys](#) for details.

Refreshing All Organizations

To refresh all organizations:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Salesforce Organizations**. The Salesforce Organizations page is displayed.
3. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all organizations is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Salesforce > Salesforce Keys** page. Refer to [Viewing Salesforce Keys](#) for details.

Viewing Details of an Organization

To view the details of an organization on CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Salesforce Organizations**.
3. Click the **Organization Name** link of the desired organization.

Alternatively, click the overflow icon () corresponding to the desired organization, and click **View/Edit Details**.

The edit view of the Salesforce Organizations page shows additional details of the selected organization under the ACCESS CONTROL, CERTIFICATES, ENDPOINTS, and CONNECTION sections. Expand each section to view more details.

Changing the Salesforce Connection

To change the Salesforce connection:

1. Expand **CONNECTION**.
2. From the **Connection ID** drop-down list, select the Salesforce connection.
3. Click **Update**.

The connection of the Salesforce organization is changed.

Managing User Permissions on Salesforce Organizations

To work with the Salesforce cloud, users/group must have the minimum set of permissions that allow them to use the Salesforce resources such as keys, organizations, and certificates. Initially, the CCKM user only has permission to view the keys. However, if required, the CCKM administrator can grant and revoke permissions.

Note

Only the users who are member of the CCKM Users group will be granted permissions to perform operations on Salesforce organizations.

Users with the following characteristics can perform operations for Salesforce keys, Salesforce organizations, and Salesforce certificates:

- Users in the **CCKM Admins** group
- Users in the **Admin** group
- Users who are administrators for a **domain**
- Users who are in the **CCKM Users** group and which have had a CCKM Admin assign permissions through the UI or the `/v1/cckm/sfdc/organizations/{id}/update-acls` endpoint in the REST API.

Note

A CCKM administrator can **only** assign permissions to do Salesforce certificate operations or Salesforce cache-only key operations through the REST API.

Adding Permissions for a User/Group

To add permissions for a user/group:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Salesforce Organizations**.
3. Click the **Organization Name** link of the desired organization.

Alternatively, click the overflow icon () corresponding to the desired organization, and click **View/Edit Details**.

4. Expand the **ACCESS CONTROL** section. This section contains the **BYOK Keys**, **Native Keys**, **Cache-Only Keys**, and **Certificates** tabs.
5. Click **Assign User/Group**. The **Assign User/Group** dialog box is displayed.
6. Select the desired user or group from the **User/Group** drop-down list.
7. Click **Save**.

The newly added user/group is displayed under **Name** in the **ACCESS CONTROL** section. You can now grant

additional permissions to the user/group, as appropriate. Refer to [Granting Permission to Perform an Operation](#) for details.

Granting Permission to Perform an Operation

To grant permissions to the user or group to perform any of the above mentioned operations:

1. In the ACCESS CONTROL section, click the desired tab.
2. Select the check box under the desired operation corresponding to the desired users or groups.
3. Click **Update**.

A success message is displayed on the screen.

To revoke permissions from a user/group, refer to [Removing a Permission](#) for details.

Removing a Permission

To remove a permission assigned to a user or group:

1. In the ACCESS CONTROL section, click the desired tab.
2. Clear the check box under the desired operation corresponding to the desired users or groups.
3. Click **Update**.

A success message is displayed on the screen.

Removing Permission from a User/Group

To remove current permissions assigned to the user/group:

1. In the ACCESS CONTROL section, click the desired tab.
2. Under **Unassign**, click the **X** button corresponding to the desired user/group.
3. On the **Remove User / Remove Group** screen, click **Remove**.

Note

Removing this user/group will remove all permissions currently assigned to the user/group.

4. Click **Remove** to confirm the action. To cancel the action, click **Keep It**.

A success message is displayed on the screen.

Removing a Salesforce Organization

Salesforce organizations can be removed on the Salesforce Organizations page. Search for existing organizations using Name or Connection.

Note

When a Salesforce organization is removed from CCKM, it is also deleted from the Salesforce cloud.

To remove an organization from CCKM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Salesforce Organizations**. The Salesforce Organizations page is displayed. The list of organizations added to CCKM is displayed.
3. Click the overflow icon () corresponding to the organization you want to remove.
4. Click **Delete**. The Delete Organization dialog box is displayed.
5. Click **Delete**.

The Salesforce organization is deleted successfully. The organization is removed from the list of Salesforce organizations.

Managing Salesforce Tenant Secrets

This section describes how to manage Salesforce tenant secrets on CCKM. The Salesforce tenant secrets are referred to as *keys* on the CCKM GUI.

When a new tenant secret is created, its status becomes **ACTIVE**, and the status of the previous tenant secret becomes **ARCHIVED**.

When an **ARCHIVED** tenant secret is destroyed, its status becomes **DESTROYED**. A destroyed tenant secret can either be imported or deleted. When a destroyed tenant secret is imported, its status changes to **ARCHIVED**.

Before proceeding, you must have a Salesforce organization added to the CCKM. Refer to [Salesforce Organizations](#) for details.

Salesforce Tenant Secret Types

CCKM lets you manage three types of tenant secrets, Salesforce Native, Bring Your Own Key (BYOK) or cache-only key.

- Salesforce Native tenant secrets are generated directly on Salesforce.
- BYOK tenant secrets are keys which CCKM creates from CipherTrust Manager or DSM, and uploads to Salesforce through a secure channel. Traffic for BYOK operations only takes place from CCKM to Salesforce

and not the reverse direction. BYOK keys can act either as key encryption keys or data encryption keys on Salesforce, depending on whether you select the key derivation option.

- Cache-only key tenant secrets are keys which CCKM creates from CipherTrust Manager or DSM, and uploads to Salesforce, which stores the key material in the encrypted key cache to encrypt or decrypt data on demand. After Salesforce flushes its cache and a new cryptographic request is made, Salesforce fetches key material from a cache-only key endpoint on CCKM. These key transmissions occur in a secure tunnel.

You can perform the following operations with Salesforce tenant secrets:

- [Key Creation Methods and Sources](#)
- [Creating/Uploading New Key Material](#)
- [Cloning Existing Key Material](#)
- [Viewing Salesforce Keys](#)
- [Refreshing Salesforce Keys](#)
- [Viewing Versions of a Key](#)
- [Destroying a Key Version](#)
- [Importing a Destroyed Key Version](#)
- [Deleting a Key Backup](#)
- [Activating a Cache-Only Key](#)

Key Creation Methods and Sources

Methods to create Salesforce keys using CCKM are:

- **Creating/Uploading New Key Material:** Add key material by creating and uploading new source key or creating new native key. The key source can be:
 - **CipherTrust (Local):** A new key is first created on the CipherTrust Manager. Then, this key material is uploaded to Salesforce to create a new Salesforce key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.
 - **Salesforce (Native):** A new key is directly created on Salesforce using the native Salesforce application. The key origin is **NATIVE**.
 - **CipherTrust (External):** A new key is first created on the external CipherTrust Manager. Then, this key material is uploaded to Salesforce to create a new Salesforce key. As the key material is uploaded from the external CipherTrust Manager, the key origin is **CipherTrust (External)**.
 - **Vormetric DSM:** A new DSM key is first created on the CipherTrust Manager. Then, this key material is uploaded to Salesforce to create a new Salesforce key. As the key material is uploaded from the
-

CipherTrust Manager, the key origin is **CCKM**.

- **Cloning Existing Key Material:** Clone key material from an existing key to create a new key. The key sources can be:
 - **CipherTrust (Local):** An existing local CipherTrust Manager key is uploaded to Salesforce to create a new Salesforce key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.
 - **Vormetric DSM:** An existing DSM key is first cloned on the CipherTrust Manager. Then, the key material is uploaded to the Salesforce cloud to create a new Salesforce key. As the key material is uploaded from the CipherTrust Manager, the key origin is **CCKM**.
 - **CipherTrust (External):** A new key is first cloned on the external CipherTrust Manager. Then, this key material is uploaded to Salesforce to create a new Salesforce key. As the key material is uploaded from the external CipherTrust Manager, the key origin is **CipherTrust (External)**.

Creating/Uploading New Key Material

To add a Salesforce key by creating/uploading new key material:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Salesforce**.
3. Click **Add Key**. The Select Material Origin screen of the Add Salesforce Key wizard is displayed.
4. Under **Select Method**, select **Create/Upload New Key Material**. The **Select Source** section appears. Depending on your requirements, select from the following sources:
 - **CipherTrust (Local):** Refer to [Uploading CipherTrust \(Local\) Key Material](#) for details.
 - **Salesforce (Native):** Refer to [Creating Salesforce \(Native\) Key Material](#) for details.
 - **CipherTrust (External):** Refer to [Uploading CipherTrust \(External\) Key Material](#) for details.
 - **Vormetric DSM:** Refer to [Uploading Vormetric DSM Key Material](#) for details.

Refer to [Key Creation Methods and Sources](#) for details on key sources.

Uploading CipherTrust (External) Key Material

Upload the key material of an external CipherTrust Manager key using the CipherTrust Manager to configure the source key.

Select Material Origin > Select CipherTrust Key

1. Select **CipherTrust (External)**.
2. Click **Next**. The Configure CipherTrust (External) Key screen is displayed.

Configure CipherTrust (External) Key

1. Select **Domain** from the drop-down list. The drop-down list shows the external CM domains linked with the configured external CM connection.
2. Enter a **Key Name**. A new key with this name will be created on the external CipherTrust Manager and its key material will be uploaded to Salesforce cloud.
3. Click **Next**.

Configure Salesforce Key

1. Select the desired **Organization** from the drop-down list. The new key will be uploaded to this organization.
2. Select the **Tenant Secret Type** from the drop-down list. The options are Data, EventBus, SearchIndex, DeterministicData, and Analytics.
3. (Optional) Select either of the following check boxes depending on your requirements:

- **Use Key Derivation**. This option is for BYOK tenant secrets only.

If you do not want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned ON (on Salesforce, **Security > Platform Encryption > Advanced Settings**), otherwise, the BYOK upload fails.

Enabling the **Allow BYOK to Opt Out of Key Derivation** option disables the Salesforce key derivation process and allows you to use your uploaded key material as the final encryption key.

However, if you want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned OFF on Salesforce.

- **Make this a cache-only key**. When you select this check box, the **Named Credential** field is displayed.

If you select **Make this a cache-only key**, ensure that the **Allow Cache-Only Keys with BYOK** option is turned ON on Salesforce. Enabling this option lets Salesforce fetch remotely-stored encryption keys from an endpoint that you provide.

For a non cache-only key (that is, when **Make this a cache-only key** is not selected), a BYOK-compatible certificate is mandatory. Refer to the next step.

4. Select the desired **Certificate** from the drop-down list.

The certificate that you select must be a BYOK-compatible 4096-RSA certificate. It can be a self-signed or CA-signed certificate. If the certificate does not meet these requirements, an error occurs and the key upload fails. Refer to [Generate a BYOK-Compatible Certificate](#) for details.

5. (Required if you selected **Make this a cache-only key**) Select the desired **Named Credential** from the drop-down list.
 6. Click **Next**. The Review and Add screen is displayed.
-

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY and DESTINATION KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Salesforce Key wizard is closed.

The newly created key is displayed in the list of Salesforce keys.

Uploading CipherTrust (Local) Key Material

Upload the local key material using the CipherTrust Manager to configure the source key.

Select Material Origin > Select CipherTrust Key

1. Select **CipherTrust (Local)**.
2. Click **Next**. The Configure CipherTrust (Local) Key screen is displayed.

Configure CipherTrust (Local) Key

1. Enter a **Key Name**. A new key with this name will be created on the CipherTrust Manager and its key material will be uploaded to Salesforce cloud.
2. Click **Next**.

Configure Salesforce Key

1. Select the desired **Organization** from the drop-down list. The new key will be uploaded to this organization.
2. Select the **Tenant Secret Type** from the drop-down list. The options are Data, EventBus, SearchIndex, DeterministicData, and Analytics.
3. (Optional) Select either of the following check boxes depending on your requirements:
 - **Use Key Derivation**. This option is for BYOK tenant secrets only.

If you do not want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key**

Derivation option is turned ON (on Salesforce, **Security > Platform Encryption > Advanced Settings**), otherwise, the BYOK upload fails.

Enabling the **Allow BYOK to Opt Out of Key Derivation** option disables the Salesforce key derivation process and allows you to use your uploaded key material as the final encryption key.

However, if you want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned OFF on Salesforce.

- **Make this a cache-only key.** When you select this check box, the **Named Credential** field is displayed.

If you select **Make this a cache-only key**, ensure that the **Allow Cache-Only Keys with BYOK** option is turned ON on Salesforce. Enabling this option lets Salesforce fetch remotely-stored encryption keys from an endpoint that you provide.

For a non cache-only key (that is, when **Make this a cache-only key** is not selected), a BYOK-compatible certificate is mandatory. Refer to the next step.

4. Select the desired **Certificate** from the drop-down list.

The certificate that you select must be a BYOK-compatible 4096-RSA certificate. It can be a self-signed or CA-signed certificate. If the certificate does not meet these requirements, an error occurs and the key upload fails. Refer to [Generate a BYOK-Compatible Certificate](#) for details.

5. (Required if you selected **Make this a cache-only key**) Select the desired **Named Credential** from the drop-down list.
6. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY and DESTINATION KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Salesforce Key wizard is closed.
-

The newly created key is displayed in the list of Salesforce keys.

Creating Salesforce (Native) Key Material

Create the key material directly using native Salesforce application.

Select Material Origin > Select Source

1. Select **Salesforce (Native)**.
2. Click **Next**. The Configure Salesforce Key screen is displayed.

Configure Salesforce Key

1. Select the desired **Organization** from the drop-down list.
2. Select the key **Type** from the drop-down list. The options are Data, EventBus, SearchIndex, DeterministicData, and Analytics.
3. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN and NATIVE KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **NATIVE KEY** section and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the NATIVE KEY section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Salesforce Key wizard is closed.

The newly created key is displayed in the list of Salesforce cloud keys. The origin of the key is **NATIVE**.

Uploading Vormetric DSM Key Material

Upload the key material using Vormetric DSM to configure the source key.

Select Material Origin > Select Source

1. Select **Vormetric DSM**.

-
- Click **Next**. The Configure DSM Key screen is displayed.

Configure DSM Key

- Enter a **DSM Key Name**. A new key with this name will be created on the DSM and its key material will be uploaded to the Salesforce cloud.
- (Optional) Provide a **Description** for the key.
- Select a **DSM Domain** for the key. The drop-down list shows the DSM domains linked with the configured DSM connection.
- Click **Next**.

Configure Salesforce Key

- Select the desired **Organization** from the drop-down list. The new key will be uploaded to this organization.
- Select the key **Type** from the drop-down list. The options are Data, EventBus, SearchIndex, DeterministicData, and Analytics.
- (Optional) Select either of the following check boxes depending on your requirements:

- **Use Key Derivation**. This option is for BYOK tenant secrets only.

If you do not want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned ON (on Salesforce, **Security > Platform Encryption > Advanced Settings**), otherwise, the BYOK upload fails.

Enabling the **Allow BYOK to Opt Out of Key Derivation** option disables the Salesforce key derivation process and allows you to use your uploaded key material as the final encryption key.

However, if you want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned OFF on Salesforce.

- **Make this a cache-only key**. When you select this check box, the **Named Credential** field is displayed.

If you select **Make this a cache-only key**, ensure that the **Allow Cache-Only Keys with BYOK** option is turned ON on Salesforce. Enabling this option lets Salesforce fetch remotely-stored encryption keys from an endpoint that you provide.

For a non cache-only key (that is, when **Make this a cache-only key** is not selected), a BYOK-compatible certificate is mandatory. Refer to the next step.

- Select the desired **Certificate** from the drop-down list.

The certificate that you select must be a BYOK-compatible 4096-RSA certificate. It can be a self-signed or CA-signed certificate. If the certificate does not meet these requirements, an error occurs and the key upload fails. Refer to [Generate a BYOK-Compatible Certificate](#) for details.

5. (Required if you selected **Make this a cache-only key**) Select the desired **Named Credential** from the drop-down list.
6. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY and DESTINATION KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Salesforce Key wizard is closed.

The newly created key is displayed in the list of Salesforce cloud keys.

Cloning Existing Key Material

To add a new Salesforce key by cloning key material existing on the CipherTrust Manager or the DSM:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Salesforce**.
3. Click **Add Key**. The Select Material Origin screen of the Add Salesforce Key wizard is displayed.
4. Under **Select Method**, select **Clone Existing Key Material**. The **Select Source** section appears. Depending on your requirements, select from the following sources:
 - **CipherTrust (Local)**: Refer to [Cloning CipherTrust \(Local\) Key Material](#) for details.
 - **CipherTrust (External)**: Refer to [Cloning CipherTrust \(External\) Key Material](#) for details.
 - **Vormetric DSM**: Refer to [Cloning Vormetric DSM Key Material](#) for details.

Refer to [Key Creation Methods and Sources](#) for details on these key sources.

Cloning CipherTrust (External) Key Material

Clone and upload the key material of an external CipherTrust Manager key using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

1. Select **CipherTrust (External)**.
2. Click **Next**. The Select CipherTrust (External) Key screen is displayed.

Select CipherTrust (External) Key

1. **Select Domain** from the drop-down list. The drop-down list shows the external CM domains linked with the configured external CM connection.
2. Select the desired key from the **Select Key** drop-down list. This field shows the available external CipherTrust Manager keys.
3. Click **Next**. The Configure Salesforce Key screen is displayed.

Configure Salesforce Key

1. Select the desired **Organization** from the drop-down list. The new key will be uploaded to this organization.
2. Select the key **Type** from the drop-down list. The options are Data, EventBus, SearchIndex, DeterministicData, and Analytics.
3. (Optional) Select either of the following check boxes depending on your requirements:
 - **Use Key Derivation**. This option is for BYOK tenant secrets only.

If you do not want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned ON (on Salesforce, **Security > Platform Encryption > Advanced Settings**), otherwise, the BYOK upload fails.

Enabling the **Allow BYOK to Opt Out of Key Derivation** option disables the Salesforce key derivation process and allows you to use your uploaded key material as the final encryption key.

However, if you want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned OFF on Salesforce.

- **Make this a cache-only key**. When you select this check box, the **Named Credential** field is displayed.

If you select **Make this a cache-only key**, ensure that the **Allow Cache-Only Keys with BYOK** option is turned ON on Salesforce. Enabling this option lets Salesforce fetch remotely-stored encryption keys from an endpoint that you provide.

For a non cache-only key (that is, when **Make this a cache-only key** is not selected), a BYOK-compatible certificate is mandatory. Refer to the next step.

-
4. Select the desired **Certificate** from the drop-down list.

The certificate that you select must be a BYOK-compatible 4096-RSA certificate. It can be a self-signed or CA-signed certificate. If the certificate does not meet these requirements, an error occurs and the key upload fails. Refer to [Generate a BYOK-Compatible Certificate](#) for details.

5. (Required if you selected **Make this a cache-only key**) Select the desired **Named Credential** from the drop-down list.
6. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Salesforce Key wizard is closed.

The newly created key is displayed in the list of Salesforce keys.

Cloning CipherTrust (Local) Key Material

Clone and upload the local key material using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

1. Select **CipherTrust (Local)**.
2. Click **Next**. The Select CipherTrust (Local) Key screen is displayed.

Select CipherTrust (Local) Key

1. Select the desired key from the **CipherTrust Key Name** drop-down list. This field shows the available local CipherTrust Manager keys.
 2. Click **Next**. The Configure Salesforce Key screen is displayed.
-

Configure Salesforce Key

1. Select the desired **Organization** from the drop-down list. The new key will be uploaded to this organization.
2. Select the key **Type** from the drop-down list. The options are Data, EventBus, SearchIndex, DeterministicData, and Analytics.
3. (Optional) Select either of the following check boxes depending on your requirements:

- **Use Key Derivation.** This option is for BYOK tenant secrets only.

If you do not want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned ON (on Salesforce, **Security > Platform Encryption > Advanced Settings**), otherwise, the BYOK upload fails.

Enabling the **Allow BYOK to Opt Out of Key Derivation** option disables the Salesforce key derivation process and allows you to use your uploaded key material as the final encryption key.

However, if you want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned OFF on Salesforce.

- **Make this a cache-only key.** When you select this check box, the **Named Credential** field is displayed.

If you select **Make this a cache-only key**, ensure that the **Allow Cache-Only Keys with BYOK** option is turned ON on Salesforce. Enabling this option lets Salesforce fetch remotely-stored encryption keys from an endpoint that you provide.

For a non cache-only key (that is, when **Make this a cache-only key** is not selected), a BYOK-compatible certificate is mandatory. Refer to the next step.

4. Select the desired **Certificate** from the drop-down list.

The certificate that you select must be a BYOK-compatible 4096-RSA certificate. It can be a self-signed or CA-signed certificate. If the certificate does not meet these requirements, an error occurs and the key upload fails. Refer to [Generate a BYOK-Compatible Certificate](#) for details.

5. (Required if you selected **Make this a cache-only key**) Select the desired **Named Credential** from the drop-down list.
6. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** and

DESTINATION KEY sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the DESTINATION KEY section becomes Complete and the Key ID link is displayed, the key is created successfully.

3. Click **Close**. The Add Salesforce Key wizard is closed.

The newly created key is displayed in the list of Salesforce keys.

Cloning Vormetric DSM Key Material

Clone and upload the DSM key material using the CipherTrust Manager to configure the source key.

Select Material Origin > Select Source

1. Select **Vormetric DSM**.
2. Click **Next**. The Select DSM Key screen is displayed.

Select DSM Key

1. Select the desired key from the **DSM Key Name** drop-down list. This field displays the available DSM keys.
2. Click **Next**. The Configure Salesforce Key screen is displayed.

Configure Salesforce Key

1. Select the desired **Organization** from the drop-down list. The new key will be uploaded to this organization.
2. Select the key **Type** from the drop-down list. The options are Data, EventBus, SearchIndex, DeterministicData, and Analytics.
3. (Optional) Select either of the following check boxes depending on your requirements:
 - **Use Key Derivation**. This option is for BYOK tenant secrets only.

If you do not want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned ON (on Salesforce, **Security > Platform Encryption > Advanced Settings**), otherwise, the BYOK upload fails.

Enabling the **Allow BYOK to Opt Out of Key Derivation** option disables the Salesforce key derivation process and allows you to use your uploaded key material as the final encryption key.

However, if you want to select **Use Key Derivation**, ensure that the **Allow BYOK to Opt Out of Key Derivation** option is turned OFF on Salesforce.

- **Make this a cache-only key.** When you select this check box, the **Named Credential** field is displayed.

If you select **Make this a cache-only key**, ensure that the **Allow Cache-Only Keys with BYOK** option is turned ON on Salesforce. Enabling this option lets Salesforce fetch remotely-stored encryption keys from an endpoint that you provide.

For a non cache-only key (that is, when **Make this a cache-only key** is not selected), a BYOK-compatible certificate is mandatory. Refer to the next step.

4. Select the desired **Certificate** from the drop-down list.

The certificate that you select must be a BYOK-compatible 4096-RSA certificate. It can be a self-signed or CA-signed certificate. If the certificate does not meet these requirements, an error occurs and the key upload fails. Refer to [Generate a BYOK-Compatible Certificate](#) for details.

5. (Required if you selected **Make this a cache-only key**) Select the desired **Named Credential** from the drop-down list.
6. Click **Next**. The Review and Add screen is displayed.

Review and Add

This screen shows the key details that you have provided. These details are divided into MATERIAL ORIGIN, SOURCE KEY, and DESTINATION KEY sections.

Before adding the key, review all details. After the key is added, certain features will no longer be editable.

1. Review the key details displayed on the screen.

If details are incorrect or you want to make any changes, click **Edit** next to the **SOURCE KEY** and **DESTINATION KEY** sections and update details. Alternatively, click **Back** and make changes, as appropriate.

2. Click **Add Key**.

The key creation starts. A Create Key In Progress message is displayed on the screen. Leave the window open until the process is completed.

When the status next to the SOURCE KEY and DESTINATION KEY sections becomes Complete and the Key ID links are displayed, the key is created successfully.

3. Click **Close**. The Add Salesforce Key wizard is closed.

The newly created key is displayed in the list of Salesforce cloud keys.

Viewing Salesforce Keys

The Salesforce Keys page shows the list of Salesforce keys available on the CipherTrust Manager. Search for keys by Organization ID or Organization Name. The Salesforce keys within an organization are categorized by data types.

To view the Salesforce keys:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Salesforce**. The list of available Salesforce keys is displayed. The Salesforce Keys page displays the following details:

Field	Description
Organization Name	Name of the Salesforce organization where the key is created.
Organization ID	ID of the Salesforce organization where the key is created.
Data Type	Type of the key data. The data type can be: • Data in Salesforce • Deterministic • Search Index • Analytics • Event Bus
Version	Latest version of the Salesforce key.
Status	State of the Salesforce key. The status can be: • Active • Archived • Destroyed
Origin	Source of the key material. The origin of the key can be: • CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud. • External (Unknown): Source of the key material is unknown. It is different than CCKM and the native cloud.
Cache Only	Whether the key is a cache-only key. A cache-only key is indicated by a check mark (✓).
Backup	Whether the key material is downloaded from the Salesforce cloud to the CipherTrust Manager. A check mark (✓) shows that the key material is downloaded. The import operation enabled for such keys. A blank column indicates that the key material of the particular key is not downloaded from Salesforce. The import operation is not available for such keys.
Creation Date	Date and time when the Google Cloud key is

Field	Description
	created.

To view keys of a particular type under a Salesforce organization, click the expand icon (▶) to the left of the Organization Name link.

Sometimes, you might notice certain Salesforce keys are displayed as grayed out. This happens when the keys are no longer accessible. For example, when:

- Any cloud permissions on the keys are changed. The keys are no longer accessible from the Salesforce cloud connection.
- Connection is changed in KMS. The new connection does not have permissions to access the keys.

Refreshing Salesforce Keys

Refreshing is the process of downloading keys created in Salesforce organizations to CCKM. You can refresh keys from all Salesforce organizations at once.

To refresh keys:

1. Open the Cloud Key Manager application.
2. In the left pane, click **Cloud Keys > Salesforce**. The Salesforce Keys page is displayed. This page displays the list of Salesforce keys.
3. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all keys is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

4. Click **Refresh All** to continue.

A message Refresh started... is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

The refreshed keys are listed on the **Cloud Keys > Salesforce > Salesforce Keys** page.

Viewing Versions of a Key

To view the versions of a key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Salesforce**.

3. Under Organization Name, click the expand icon () to the left of the desired key type. The key version details are displayed:

Field	Description
Data Type	Data type of the key.
Version	Version number of the key. All versions of the key are listed, each on a separate row.
Status	State of the Salesforce key. The status can be: • Active • Archived • Destroyed
Origin	Source of the key material. The origin of the key can be: • CCKM: Key material is created on CCKM. • Native: Key material is created on the cloud. • External (Unknown): Source of the key material is unknown. It is different than CCKM and the native cloud.
Cache Only	Whether the key is a cache-only key. A cache-only key is indicated by a check mark ().
Backup	Whether the key material is downloaded from the Salesforce cloud to the CipherTrust Manager. A check mark () shows that the key material is downloaded. The import operation enabled for such keys. A blank column indicates that the key material of the particular key is not downloaded from Salesforce. The import operation is not available for such keys.
Creation Date	Date and time when the Google Cloud key is created.

Alternatively, click the overflow icon () corresponding to the desired key, click **View/Edit**, and expand **KEY VERSIONS** to view key version details. You can also click the Organization Name, Organization ID, or Data Type link corresponding to desired key to view the key version details.

Destroying a Key Version

If required, you can destroy archived versions of a key. A destroyed version cannot operate on data. Destroy tenant secrets and key material if you no longer need access to related data. After a key material is destroyed, related data cannot be accessed unless you import previously exported key material.

A destroyed key version can be imported or deleted later.

To destroy an archived key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Salesforce**.
3. Click the expand icon () to the left of the desired key.
4. Click the overflow icon () corresponding to the desired archived version.
5. Click **Destroy**. The Destroy Key dialog box is displayed.
6. Click **Destroy** to confirm the action.

The state of the key version changes to Destroyed.

Importing a Destroyed Key Version

If required, you can import a destroyed version of a key. After a version is imported, its status becomes archived.

To import a key version:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Salesforce**.
3. Click the expand icon () to the left of the desired key.
4. Click the overflow icon () corresponding to the desired destroyed version.
5. Click **Import**. The Import Key dialog box is displayed.
6. Click **Import** to confirm the action.

The state of the key version changes to Archived.

Deleting a Key Backup

You can delete the backup of a key from CCKM. After the key backup is deleted, it cannot be imported into the Salesforce cloud. Only the backup of a destroyed key can be deleted.

To delete a key backup:

1. Open the **Cloud Key Manager** application.
 2. In the left pane, click **Cloud Keys > Salesforce**.
 3. Click the expand icon () to the left of the desired key.
 4. Click the overflow icon () corresponding to the desired destroyed version.
 5. Click **Delete**. The Delete Backup dialog box is displayed.
-

6. Select **I wish to delete the backup of this key**.
7. Click **Delete Key Backup** to confirm the action.

The deleted key backup cannot be imported to the Salesforce cloud.

Activating a Cache-Only Key

Cache-only keys allow for an additional key status transition on Salesforce, from Destroyed to Active. This causes Salesforce to fetch the cache-only key material from CCKM.

The Salesforce API only allows CCKM to activate the *latest* version of the cache-only key. You cannot activate older versions.

To activate a cache-only key:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Cloud Keys > Salesforce**.
3. Click the expand icon (▶) to the left of the desired key.
4. Click the overflow icon (⋮) corresponding to the latest key version.
5. Click **Activate**.

Scheduling Operations

CCKM allows you to periodically refresh or rotate keys in the background. To refresh or rotate keys, you need to create schedule configuration. A schedule configuration defines when to run a refresh or rotation job. You can use either basic format or cron format to specify the time when the rotation or refresh job will run.

Specify basic format in the following order:

[Repeats, at]

Where,

- Repeats: frequency of the scheduler. Possible values are daily, weekly, monthly, and yearly.
- at: specific time at which the rotation or refresh operation will be performed. Possible value lies between 12:00-11:00 AM/PM UTC.

Specify cron format in the following order:

" * * * * * "

These five values indicate when the job should be executed. These values are mandatory and must be specified in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

Field	Allowed values	Allowed special characters
Minute	0-59	* / , -
Hour	0-23	* / , -
Day of month	1-31	* / , - ?
Month	1-12 or JAN-DEC	* / , -
Day of week	0-6 or SUN-SAT	* / , - ?

Adding Key Rotation Schedule

A key rotation schedule replaces selected keys with new cryptographic keys at the specified time.

To add key rotation schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. Select **Schedule Type** as **Key Rotation**.
5. Click **Next**.
6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **Salesforce** from the **Cloud Name** drop-down list.
 - Add **Description** for the scheduler in a maximum of 250 characters.
 - Select **Enable Schedule** to enable the schedule.
 - Click **Next**.
7. On the **Schedule Config** screen, enter or select the following details.
 - a. Configure **DURATION** for the scheduler. Specify the schedule start and end time:
 - **Schedule Starts**: Specify time when the schedule starts.
 - **Schedule Ends**: Unavailable by default, that is, the scheduler never expires.
 - **Never**: Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.

- b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate

keys. Select either **Basic** or **Raw (Cron)** format to specify the value.

- c. Configure **KEY SOURCE**. Select from the **Key Source** drop-down list. The options are Native, DSM, CipherTrust (Local), and CipherTrust (External). This will be the source of the new key material.
 - If the key source is DSM, you need to specify the DSM domain where the key will be created, the optional certificate whose public key will be used to encrypt the key, and whether to use key derivation.
 - If the key source is CipherTrust (Local) or CipherTrust (External), you can specify the optional certificate whose public key will be used to encrypt the key and whether to use key derivation.
- d. (Only for DSM and CipherTrust (External) key source) Select **Domain** where the key will be created to upload to Salesforce.
- e. Select **ORGANIZATION** to which the key belongs. Select from the **Organization** drop-down list.
- f. (Optional, only for DSM, CipherTrust (Local) and CipherTrust (External) key sources) Select the **Certificate** whose public key is to be used to encrypt the key.
- g. Configure **KEY TYPE**. Select the types of the key to be rotated. The options are Data, Event Bus, Search Index, Deterministic Data, and Analytics.
- h. (Optional) Select the **Use Key Derivation** check box. This field is not applicable to the Native key source.
- i. Click **Save**.

A message **Schedule successfully created** is displayed on the screen.

Adding Key Refresh Schedule

A key refresh schedule synchronizes keys from linked Salesforce cloud organizations to CCKM.

To add a key refresh schedule:

1. Open the **Cloud Key Manager** application.
 2. In the left pane, click **Schedules**. The list of available schedules is displayed.
 3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
 4. **Select Schedule Type** as **Key Refresh**.
 5. Click **Next**.
 6. On the **General Info** screen, enter or select following details:
 - Enter a unique **Scheduler Name**.
 - Select **Salesforce** from the **Cloud Name** drop-down list.
 - Add **Description** for the scheduler in a maximum of 250 characters.
-

- Select **Enable Schedule** to enable the schedule.
- Click **Next**.

7. On the **Schedule Config** screen, enter or select the following details.

- a. Configure **DURATION** for the scheduler. Specify the schedule start and end time:
 - **Schedule Starts**: Specify time when the schedule starts.
 - **Schedule Ends**: Unavailable by default, that is, the scheduler never expires.
 - **Never**: Selected by default, that is, the scheduler configuration never expires.

To set an end time for the scheduler, clear the **Never** check box, and specify the **Scheduler Ends** time.

- b. Configure **FREQUENCY** of the scheduler. The frequency determines how often the scheduler will rotate keys. Select either **Basic** or **Raw (Cron)** format to specify the value.
- c. Click **Next**.

8. On the **Add Organization** screen:

- a. Select the desired organization from the **Refresh Keys from Selected Organization(s)** drop-down list.
- b. Click **+** button. Similarly, you can add more organizations. To remove an added organization, click the close (X) icon in the name of the added organization.

Alternatively, if you want to refresh all the existing organizations, select the **Refresh All Organizations** check box. When selected, the **Refresh Keys from Selected Organization(s)** field becomes unavailable.

- c. Click **Save**.

A message **Schedule successfully created** is displayed on the screen. The newly created scheduled is displayed in the schedules list.

Viewing/Editing Schedules

To view/edit a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The Schedules page displays following details:

Field	Description
Name	Unique name of the scheduler configuration.
Schedule Type	Type of the schedule. Possible types are: • Key Rotation

Field	Description
	• Key Refresh • Add KMS Container (GCP and Azure only)
Cloud Name	Name of the cloud.
Last Modified	Time when the schedule is modified.
Frequency	Frequency of the scheduler configuration.
Start Date	Creation time of the scheduler configuration.
End Date	Expiry time of the scheduler configuration. Never is displayed if the schedule is set to never expire.
Status	Status of the scheduler configuration. Possible values are: • Enabled • Disabled

- Click the overflow icon () corresponding to the desired schedule and click **View/Edit**.

Alternatively, click the **Name** link corresponding to the desired schedule.

- Edit or configure the following fields:

- Optional, basic description of the schedule.
- Status of the scheduler configuration. Select **Enable Schedule** to enable the schedule, clear to disable.
- Scheduler configuration parameters such as duration, frequency, and expiry time of the key. Refer to the key rotation [Schedule Config screen](#) or key refresh [Schedule Config screen](#), as appropriate for details.
- (For key refresh schedules only) Organizations whose keys you want to refresh. If you want to refresh all the existing organizations, select the **Refresh All Organizations** check box. When selected, the **Refresh Keys from Selected Organization(s)** field becomes unavailable.

- Click **Update** to save the changes.

The Schedules page also contains a section named **JOB HISTORY**. This section displays information related to a schedule such as **Run Date**, **Job ID**, **Status**, and **Errors**.

Disabling Schedules

To disable a schedule configuration:

- Open the **Cloud Key Manager** application.
- In the left pane, click **Schedules**. The list of available schedules is displayed.

-
3. Click the overflow icon () corresponding to the desired schedule and click **Disable**. The Disable Schedule message is displayed.
 4. Click **Disable Schedule**.

A message **Successfully disabled the schedule** is displayed on the screen.

Enabling Schedules

To enable a schedule configuration:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Enable**. The Enable Schedule message is displayed.
4. Click **Enable Schedule**.

A message **Successfully enabled the schedule** is displayed on the screen.

Manually Running Schedules

To manually run a schedule:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Run Now**.

The Run Now schedule is started in the background. A message **Schedule is running now. It will take a few seconds to finish.** is displayed on the screen. After the schedule is run successfully, a message **Successfully finished running the schedule** is displayed.

Deleting Schedules

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Schedules**. The list of available schedules is displayed.
3. Click the overflow icon () corresponding to the desired schedule and click **Delete Schedule**. The Delete Schedule message is displayed.
4. Click **Delete**.

A message **Successfully deleted schedule** is displayed on the screen.

Managing Salesforce Reports

CCKM provides options to generate key visibility reports based on key-related activities between CCKM and Salesforce organization. Salesforce supports the Key Activity Report.

Use the Key Activity Report to inspect individual Salesforce tenant secret (called key on CCKM GUI) histories by operations, for example, when they were refreshed, rotated, edited, or deleted. Also, use this report to compare key activities between CCKM and Salesforce organization.

Creating Salesforce Reports

To create a Salesforce report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Salesforce**. The Salesforce Reports page is displayed.
3. Click **Add Report**. The Choose Report Type and Name screen of the Add Salesforce Report wizard is displayed.
4. Select **Key Activity Report**.
5. Specify a **Report Name**. This is a mandatory field.
6. Click **Next**. The Choose Organization screen of the wizard is displayed.
7. Select a Salesforce organization from the **Organizations** drop-down list. The drop-down list contains the list of Salesforce organizations added to the CipherTrust Manager.
8. Click **Next**. The Set Start and End Dates screen of the wizard is displayed.
9. In the **Include Entries From** field, specify the start date and time for the report.
 - Click the field and select the date and time on the on-screen calendar.
 - Alternatively, enter the time in **MM/DD/YYYYHH : MM** format.
10. In the **To** field, specify the end date and time for the report.
11. Click **Save**.

A success message **Created report** is displayed. The newly created report is displayed under the Salesforce reports list. The report Run Date becomes *Running Now*. After the run is complete, the date and time of the run is displayed.

Viewing Salesforce Reports

The Salesforce Reports page displays the list of existing reports. Filter the reports by their names.

To view existing Salesforce reports:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Salesforce**. The list of available reports is displayed. The following details of the reports are displayed:

Column Name	Description
Report Name	Name of the report.
Run Date	When the report is run. If a report is running currently, Running Now is displayed.
Type	Type of the report is Key Activity Report.
Start Date	Start date from when the report is generated.
End Date	End date for the report.

Viewing Details of a Salesforce Report

To view the details of a Salesforce report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Salesforce**. The list of available reports is displayed.
3. Click the desired **Report Name** link.

Alternatively, click the overflow icon () corresponding to the desired report and click **View/Edit**.

The following details of the reports are displayed:

Column Name	Description
Key Name	Name of the key.
Key Version	Version of the key.
Organization ID	ID of the Salesforce organization.
Modified Date	When the report is modified.
CCKM Operation	CCKM operation performed. The operation can be TS Generated and TS Read.
Salesforce User	Name of the user on Salesforce.

Downloading Salesforce Reports

To download a Salesforce report as a CSV file:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Salesforce**. The list of available reports is displayed.
3. Click the overflow icon () corresponding to the desired report and click **Download**.

Alternatively, click the desired report under **Report Name**, and click **Download CSV** in the detail view.

The report is downloaded as a CSV file.

Deleting Salesforce Reports

To delete a Salesforce report:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **Reports > Salesforce**. The list of available reports is displayed.
3. Click the overflow icon () corresponding to the desired report and click **Delete**. The Confirm Delete Report message is displayed.
4. Click **Delete Report**.

The report is removed from the list of reports.

Managing Salesforce Cache-Only Key Endpoints

CipherTrust Cloud Key Manager (CCKM) can act as a key service for [Salesforce Shield Platform Encryption's Cache-Only Key Service](#). Salesforce fetches keys from CCKM endpoints on demand to immediately encrypt and decrypt data. The key material is securely transmitted through an authenticated channel. Salesforce then retains the key in the encrypted key cache until the cache is flushed.

You can provide a cache only key sourced from [local CipherTrust Manager](#) or [Data Security Manager \(DSM\)](#). Salesforce requires that the key algorithm is AES 256.

CCKM / CipherTrust Manager Prerequisites

- The CipherTrust Manager network location must be publicly accessible, with a Fully Qualified Domain Name (FQDN).
- The web interface must have a [TLS certificate signed by an external Certificate Authority \(CA\) trusted by Salesforce](#). Salesforce trusts certificates issued by well-known public CAs such as Verisign.
- The web interface port within CipherTrust Manager can be changed from the default port of 443 to another port. If you plan to change the default port for the CipherTrust Manager web interface, ensure to change it **BEFORE** configuring the Salesforce Cache-Only Key service on CCKM. Also reflect this port change when creating an endpoint for Salesforce Cache-Only Key on CCKM as part of the prerequisites for deploying Salesforce Cache-Only Key. **Changing the default port AFTER configuring Salesforce Cache-Only Key is not supported.**

- You must have a configured [Salesforce Connection](#) and associated [Salesforce organization](#).

Salesforce Prerequisite

Enable "Allow Cache-only Keys" in Platform Encryption Advanced Settings. Replay Detection is optional.

Prepare the Cache-Only Key Integration

1. Configure a [Salesforce Connection](#) and associated [Salesforce organization](#) if you have not already.
2. Create the endpoint. This endpoint is tied to a Salesforce organization, as identified by the organization ID generated and applied by CCKM. You must choose a username and password, which serve as the Named Credential for Salesforce communication to the endpoint, and a friendly name for the endpoint. You must also provide the base hostname for CipherTrust Manager.

```
ksctl cckm salesforce endpoints create --sfdc_organization_resource_id <CCKM-organization-id> --url_hostname <base-hostname-for-CipherTrust-Manager> --name <friendly-name> --epusername <username-for-Salesforce-communication> --eppassword <password-for-Salesforce-communication>
```

This functionality is also available in the CipherTrust Manager web console under **Cloud Key Manager > Salesforce Organizations**. Click a particular Organization, and expand **Endpoints**. Click **Add Cache-Only Key Endpoint**.

When the endpoint is created, a unique ID is generated, and the endpoint is available for Salesforce at a URL of `https://<ciphertrust-hostname>/api/v1/cckm/sfdc/endpoints/<unique-id>/keys/`

Note

If you changed the web interface port within CipherTrust Manager from the default port of 443 to another port, ensure to add that same port number when configuring the Cache-Only Key Endpoint by specifying the new port number along with the hostname. For example, `example.com:8443`.

3. Associate at least one certificate with the Salesforce organization in CCKM.

- To create a new certificate.

```
ksctl cckm salesforce certificates add --name <certificate-name> --sfdc_organization_resource_id <CCKM-organization-ID>
```

This functionality is also available in the CipherTrust Manager web console under **Cloud Key Manager > Salesforce Organizations**. Click a particular Organization, and expand **Certificates**. Click **Add**

Certificate and provide a Name.

- To use a certificate already present in Salesforce, synchronize the certificates from the Salesforce organization.

```
ksctl cckm salesforce certificates synchronization-jobs start --sfdc_organization_resource_ids <CCKM-organization-id> --synchronize_all false
```

This functionality is also available in the CipherTrust Manager web console under **Cloud Key Manager > Salesforce Organizations**. Click a particular Organization, and expand **Certificates**. Click **Refresh All**.

4. Add the Named Credential in [Salesforce console](#):
 - a. Specify the endpoint URL.
 - b. For Authentication, specify **Identity Type** as **Named Principal** and **Authentication Protocol** as Password Authentication.
 - c. Add the username and password you associated with the endpoint in CCKM.
5. Return to CCKM to upload a cache-only key associated with the Named Credential to Salesforce. This cache-only key becomes a tenant secret on Salesforce, with meta data retained.
6. To use the CLI to upload an existing key from [local CipherTrust Manager](#) or [Data Security Manager \(DSM\)](#).
 - a. Fetch the named credential from Salesforce. You require the organization ID generated and applied by CCKM.

```
ksctl cckm salesforce endpoints get-named-credentials --sfdc_organization_resource_id <organization-id>
```

The 18-digit value returned in the **id** field is the relevant value. For example, **0XA3h000000L0D5GAK** is the named credential ID in the following response.

```
{
  "totalSize": 1,
  "done": true,
  "records": [
    {
      "attributes": {
        "type": "NamedCredential",
        "url": "/services/data/v52.0/subjects/NamedCredential/0XA6h000000L0D7GAK"
      },
      "Endpoint": "https://test.com/kmaas/byok/00D7X000000EM3TEA8/"
    }
  ]
}
```



```

    "id": "0XA3h000000L0D5GAK",
    "IsDeleted": false,
    "MasterLabel": "test-namedcredential-1",
    "DeveloperName": "test-namedcredential-1"
  }
],
"organizationId": "a916ba75-242b-4f29-b886-52bf51d70e16"
}

```

b. Upload the key. You must provide the following values:

- **Organization ID.** This is the organization ID generated and applied by CCKM.
- **Type:** Type for tenant secret. One of Data, EventBus, SearchIndex, DeterministicData and Analytics. This depends on the Salesforce encryption purpose for the cache-only key.
- **Source for the key:** One of **local** (from CipherTrust Manager) or **dsm** (from Data Security Manager).
- **The key ID**
- **The certificate ID**
- **The named credential ID** from step A.

```

ksctl cckm salesforce keys upload-cache-only-key --sfdc_organization_resource_id <organization-id> --name <key-name> --type <tenant-secret-type> --source-key-tier <source-of-cache-only-key> --source-key-identifier <key-id> --certificate-id <certificate-associated-with-organization> --sfdc-namedcredential-id <named-credential-id>

```

7. Alternatively, you can use the CipherTrust Manager web console to create a new key or upload an existing key from CipherTrust Manager or DSM. Follow the steps to create a tenant secret for [CipherTrust Manager](#) or [DSM](#), making sure to select the option to **Make this a cache-only key**, and provide **Certificate** and **Named Credential**.

Note

The option presented in the CipherTrust Manager GUI to create a **Salesforce (Native)** key cannot create cache-only keys.

At this point, you can manage the cache-only key using CCKM's [Salesforce Tenant Secret](#) menus and commands.

View a Salesforce Cache-Only Key Endpoint

You can view a Salesforce cache-only key endpoint through the CipherTrust Manager GUI, CLI, or REST API. An endpoint's URL is needed for Salesforce to make requests to the endpoint.

- In the GUI, Salesforce cache-only key endpoints are grouped under their organizations. Navigate to **Cloud Key Manager > Salesforce Organizations**. Click a particular Organization, and expand **Endpoints**.
- In the CLI, Salesforce cache-only key endpoint details can be retrieved using a variety of identifiers.

To view a single endpoint:

```
ksctl cckm salesforce endpoints get --id <endpoint-id>
```

To view all endpoints:

```
ksctl cckm salesforce endpoints list
```

To return multiple endpoints filtered on name, organization name, organization ID, or organization resource ID.

```
ksctl cckm salesforce endpoints list [--name <endpoint-id> --organization-name  
<org-name> --sfdc_organization_id <organization-ID> --sfdc_organization_resourc  
e_id <organization-resource-id> ]
```

- In the REST API, use `GET /v1/cckm/sfdc/endpoints` to list details for all endpoints, and `GET /v1/cckm/sfdc/endpoints/{id}` to list details for a specific endpoint.

Edit a Salesforce Cache-Only Key Endpoint

You can edit a Salesforce cache-only key endpoint through the CipherTrust Manager GUI, CLI, or REST API. You can change the base hostname, and the username and password that correspond to the **Named Credential** in Salesforce. In the GUI, you can also update the friendly name corresponding to the password.

- In the GUI, Salesforce cache-only key endpoints are grouped under their organizations.
 - a. Navigate to **Cloud Key Manager > Salesforce Organizations**.
 - b. Click a particular Organization, and expand **Endpoints**.
 - c. Click the overflow icon () corresponding to the desired endpoint, and click **Edit**.
- In the CLI, Salesforce cache-only key endpoints can be edited using the endpoint ID.

```
ksctl cckm salesforce endpoints update --id <endpoint-id> [ --url_hostname <bas  
e-hostname> --epusername <named-credential-username> --eppassword <named-creden  
tial-password>]
```

- In the REST API, use `PATCH /v1/cckm/sfdc/endpoints/{id}` to edit the endpoint.

Delete a Salesforce Cache-Only Key Endpoint

You can delete a Salesforce cache-only key endpoint through the CipherTrust Manager GUI, CLI, or REST API. This makes the endpoint inaccessible to Salesforce, and the cache-only key becomes unusable on Salesforce. Make sure that Salesforce applications are no longer using a cache-only key before deleting a Salesforce cache-only key endpoint.

- In the GUI, Salesforce cache-only key endpoints are grouped under their organizations.
 - a. Navigate to **Cloud Key Manager > Salesforce Organizations**.
 - b. Click a particular Organization, and expand **Endpoints**.
 - c. Click the overflow icon () corresponding to the desired endpoint, and click **Delete**.
- In the CLI, Salesforce cache-only key endpoints can be deleted using the endpoint ID.

```
ksctl cckm salesforce endpoints delete --id <endpoint-id>
```

- In the REST API, use `DELETE /v1/cckm/sfdc/endpoints/{id}` to delete the endpoint.

Managing Certificates from Salesforce Organizations

This section describes how to manage the certificates of a Salesforce organization within CCKM. Certificates within a Salesforce organization can be added, viewed, and deleted from the **Certificates** section of the **Salesforce Organizations** page.

Adding a New Certificate to an Existing Salesforce Organization

In CCKM, you have the ability to add a new certificate to an existing Salesforce organization. Before proceeding, ensure the Salesforce organization is added to CCKM. Refer to [Managing Salesforce Organizations](#) for more information.

To add a new certificate to a Salesforce organization:

1. Open the **Cloud Key Manager** application.
2. In the left pane, click **KMS Containers > Salesforce Organizations**. The Salesforce Organizations page shows the list of organizations added to the CipherTrust Manager.
3. Click the **Organization Name** link of the desired organization.

Alternatively, click the overflow icon () corresponding to the desired organization, and click **View/Edit Details**.

The edit view of the Salesforce Organizations page shows additional details of the selected organization under the **ACCESS CONTROL**, **CERTIFICATES**, **ENDPOINTS**, and **CONNECTION** sections.

4. Expand the **CERTIFICATE** section. The **CERTIFICATE** section expands to show additional details for certificates.
5. Click **Add Certificate**. The **Add Certificate** is displayed along with the name of the organization name.
6. Enter the name of the certificate in the **Name** field.
7. Click **Add**.

The new certificates are displayed on the **CERTIFICATE** section. Now, you can manage this new certificate from CCKM on the CipherTrust Manager.

Viewing an Organization's Certificates

To view an organization's certificates:

1. Perform steps 1 through 4 in [Adding a New Certificate to an Existing Salesforce Organization](#).
2. The **CERTIFICATE** section displays the following certificate details:

Column	Description
Name	Name of the certificate.
Subject	The owner or user of the certificate.
Valid From	The start date of the certificate's validity period.
Valid To	The end date of the certificate's validity period.
Issuer	The CA that issued the certificate.

Viewing Details of an Organization's Certificate

The **Details** page provides a table with a list of all of the properties of a certificate along with the values for each including the organization ID, organization name, Salesforce organization ID, and key size.

To view the details of an organization's certificate:

1. Perform steps 1 through 4 in [Adding a New Certificate to an Existing Salesforce Organization](#).
2. Click the overflow icon () corresponding to the certificate for which to view the details.
3. Click **Details**. The **Details** page is displayed.

Refreshing All Certificates for an Organization

When you wish to retrieve new certificates created in Salesforce and download these to CCKM, you can use **Refresh All** from the **CERTIFICATE** section. Thereafter, the refreshed certificates display within the section.

To refresh all certificates for a Salesforce organization:

1. Perform steps 1 through 4 in [Adding a New Certificate to an Existing Salesforce Organization](#).
2. Click **Refresh All**. The **This may take a while...** message is displayed.

Note

Refresh all organization certificates is a time intensive operation that could take several hours or days to complete. It will continue running in the background.

3. Click **Refresh All** to continue.

A message **Refresh started...** is displayed on the screen. To cancel the refresh, click **Cancel Refresh**.

After the Refresh is completed, a message **Refresh Completed** is displayed on the screen, and the refreshed certificates are listed within the CERTIFICATE section.

Removing an Organization's Certificate

Using CCKM, you can remove or delete certificates from an organization.

To remove a certificate from an organization:

1. Perform steps 1 through 4 in [Adding a New Certificate to an Existing Salesforce Organization](#).
2. Click the overflow icon () corresponding to the certificate to remove.
3. Click **Delete**. The **Delete Certificate** dialog box displays.

A message **Are you sure you want to delete?** is displayed on the screen. To cancel the deletion, click **Cancel**.

4. Click **Delete** to confirm the deletion.

The certificate is deleted successfully, and it is removed from the list of organization certificates.

Migrate from CCKM Appliance

This section describes steps to export migration data from a CCKM Appliance to CCKM on a CipherTrust Manager (referred to as CCKM Embedded).

Before proceeding with migration:

- Note that the origin of migrated keys might not match between the CCKM Embedded and CCKM Appliance due to different terminologies used by these solutions.
- Pause all the schedules in the CCKM Appliance until the migration is complete, otherwise, the CCKM Appliance and CipherTrust Manager might have data mismatch after migration. After the migration is

successful, you should enable all the schedules on the CipherTrust Manager.

- Ensure the CCKM Appliance is in sync with the configured clouds.
- If the CCKM Appliance is in a cluster, you need to perform migration **only** on any one node.

Minimum Supported Versions

- CCKM Appliance 1.10.4
- Data Security Manager 6.4.3

Note

Data Security Manager (DSM) is EOL as of June 2024 and its support as a key source is being deprecated.

- CipherTrust Manager 2.13.0

Supported Clouds

The migration only supports **AWS Cloud**, **Azure Cloud**, **Google Cloud**, **Salesforce**, and **Salesforce Sandbox** data.

Note

Migration of IBM Cloud and Salesforce Government Cloud data is not supported.

Prerequisites

- CCKM Appliance is configured with a supported version of key source (DSM or CipherTrust Manager)
- IP address, admin credentials, and SSL certificate of the CCKM Appliance
- CCKM Appliance is accessible from the CipherTrust Manager. By default, the CipherTrust Manager connects to the CCKM Appliance on port 443. If the CCKM Appliance is not running on the default port (443), you need to specify the custom port along with the IP address in the steps described below.
- DSM servers are accessible from the CipherTrust Manager
- The CipherTrust Manager is backed up (for rollback, if required)

Scenarios

1. Migrate cloud keys with DSM as key source, move source keys to a local CM

The steps below apply to root domain only. To perform migration on a child domain, refer to [Migration from CCKM Appliance to Child Domain](#).

Generate RSA Key Pair

Exporting migration data from CCKM Appliance requires an RSA key pair (public and private) on the CipherTrust Manager. The public key is used to encrypt the data while the private key is used to decrypt the exported data.

To generate an RSA key pair, run the `ksctl keys create` command:

```
./ksctl-linux-amd64 keys create --name <rsa-key-name> --alg <key-algorithm> --size <key-size>
```

Here,

- `--name`: Name for RSA key pair.
- `--alg`: Algorithm for the RSA key pair.
- `--size`: Size for the key pair.

Example:

```
./ksctl-linux-amd64 keys create --name rsa-key --alg RSA --size 4096
```

Output:

```
{
  "id": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1",
  "uri": "kylo:kylo:vault:keys:rsa-key-v0",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-05-07T03:55:47.519466Z",
  "name": "rsa-key",
  "updatedAt": "2021-05-07T03:55:47.519466Z",
  "usage": "sign",
  "usageMask": 3,
  "meta": {
    "ownerId": "local|5e3b45c6-6f26-4413-9752-e6fd15418a61"
  },
  "version": 0,
  "algorithm": "RSA",
  "size": 4096,
  "unexportable": false,
```

```

    "undeletable": false,
    "neverExported": true,
    "neverExportable": false,
    "emptyMaterial": false,
    "publickey": "-----BEGIN PUBLIC KEY-----\nMIICijANBgkqhkiG9w0BAQEFAAOCAg8AMIICGgKCAgEAWfd4fDEgdJydUP0kPpAL\nnriQW+IpwM9oSte//pv45UXpw0wkag1FbSmEUQQMf02KdRW5s\no+4jrkX44gQhmDTA\nnamrpijweJa0HzkaqnTkMtCutFP9nbx33JiWRSYSKqXsEJho+P9SqXz45uxf7iis\n5\n4NF0SpZSIYi3COH4xtJ7hK+6BXbLXBZHVPfQ6LN8p/+WDwcIIkSywGQxj4V0xTwV\n\nnfNBNoQrXvanr\nEX+nide28vuX1bJ1UzghUwcFT12VZL8KIrkvICKMwkBNiuiiXgh\n\nnbtYGBY84ZbPjREgaodbaU45vj3\n8/dpusL75Q2hkUdv5mYvTqN+OPVbJrTTQFzGfw\n\nnM3Pt86iBFfu3XH/ZMH4dbV3HHXJP7+mHI3c0hUlvo\njwx9hnKygn3fY4Darx/N0yr\n\nniCp6Sz7FI3sExAAIEF+AJ7zqyXK6a/NGve5gAqt1w3fn0YIFeD8f6oX0\nYBFFniu3\n\n3uX//4WcNdgyTXKXhDsZAtaLqmHv9jIwGZ0pTlj8xefZPbkoDN0N3uC92b0tzI7F\n\nn7+Iq0\niEf5bg4huU/EJh8emYgU8mPZGpwPtPVUFiKm0Y7EbvHS1C6RIqRE1hnCZAa\n\nnZSMup6LLzZGvk6SM0339\nC5gDJuS+kGkYK/f0wuWJ7q05m+T/27J1IoNna6JuZ9eL\n\nnZDMxs7Rqj4cdezaa3CTV4l8CAwEAA\nQ==\n\n-----END PUBLIC KEY-----\n",
    "defaultIV": "78f83dddc0ee01a2ab3ff579c908a33a",
    "sha1Fingerprint": "878bcd84e81c4170",
    "sha256Fingerprint": "c9c2d321b21d34a3e82460df8839e55f3ebca977766658d830d5100\nfb29bed75",
    "objectType": "Private Key",
    "activationDate": "2021-05-07T03:55:43.229267Z",
    "state": "Active",
    "aliases": [
      {
        "alias": "rsa-key",
        "type": "string",
        "index": 0
      }
    ],
    "links": [
      {
        "id": "6dc578f7-1864-43a7-899a-5035d54f1772",
        "uri": "kylo:kylo:vault:links:6dc578f7-1864-43a7-899a-5035d54f1772",
        "account": "kylo:kylo:admin:accounts:kylo",
        "application": "ncryptify:gemalto:admin:apps:kylo",
        "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
        "createdAt": "2021-05-07T03:55:47.531762Z",
        "updatedAt": "2021-05-07T03:55:47.531762Z",
        "type": "publicKey",
        "source": "kylo:kylo:vault:keys:rsa-key-v0",
        "sourceID": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202\nbd4c33c1",
        "target": "kylo:kylo:vault:keys:rsa-key-pub-v0",
        "targetID": "bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c213856\n9d99e395",
        "index": 0
      }
    ],
    "uuid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3",
    "muid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3e129ed33-317d-4584-9cbf-d0e882f5\n8fca"
  }

```

In the sample output above, "sourceID":

"b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1" is the private key ID. The "targetID":

"bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c2138569d99e395" under "links" is the public key ID, which is used to download the export file.

Download Export File

As a CipherTrust Manager administrator, download the migration data (the export file) from the CCKM Appliance. This file contains migration data to be uploaded to CCKM Embedded.

Run the command:

```
./ksctl-linux-amd64 migrations download --cckm-host "<cckm-ip-or-hostname>" --cckm-user "<cckm-user>" --cckm-certificate "<certificate-file-content>" --public-key-id "<public-key-id>"
```

Here,

- **--cckm-host**: IP address or hostname of the CCKM Appliance.
- **--cckm-user**: Admin user of the CCKM Appliance.
- **--cckm-certificate**: SSL certificate downloaded from the CCKM Appliance.

By default, the content of the SSL certificate spans over multiple new lines. However, to input the certificate as an inline argument, the certificate content needs to be converted into the single-line format where every new line is represented by `\n`.

To convert the certificate into the single-line format:

- a. Download the certificate in a file (say `certificate.cert`).
- b. Store the certificate file at a location on your machine, for example, at `/root/cert/`.
- c. Run the following command:

```
awk 'NF {sub(/\r/, ""); printf "%s\\n",$0;}' <full-path-to-certificate-file>
```

For example, to convert your certificate file (`/root/cert/certificate.cert`), run the command:

```
awk 'NF {sub(/\r/, ""); printf "%s\\n",$0;}' /root/cert/certificate.cert
```

The above command returns the certificate in the single-line format, as shown below:

```
-----BEGIN CERTIFICATE-----\nMIIDhjCCAm6gAwIBAgIGCBPpf2sUMA0GCSqGSIb3DQEBAU  
AMC8xIDAeBgNVBAMT\nF0NHIENBIFMgb24gcGt2LWNsdXN0ZXIxMQswCQYDVQQGEWJVUzAeFw0yM  
TAzMMDMw\nNzI0MDVaFw0zMTAzMDQwNzI0MDVaMCQxFTATBgNVBAMTDHBrdi1jbHVzdGVyMTEl\nnMAkGA1UEBhMCVVMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQD0h14c\n\nhxamUh7eO/T
```

```

zdS016djkaL8x1a/uljy8RediBA2W5Yrmm4sTBx0/PFZ0l+0wyzYe\n7VqsCG3P5+lsTiXw2VLXK
T7qyiil7P8ParCoKt6v5fp6BjtDZzq7q3uG8daZqL4M\nKrAiI8rcGPDgsAex/kSPAB4di1AFjDD
Nv7SCiet5kWJq8YeTZEuwYa8rxLi2QEqJ\n4h/mHDqD35jNiSf4FRksWfUwtsY652XIbVpDnwkp
sA/+kEN04Aiuve9CSvyIBmR\nKaqBwnmYdQZaBJGfKEvjnArtSMMwCULxikQCiuK1BaW9H7f/WQ
UuNh7xManRb7w\ndNkhqAhSPyqgjz/TAgMBAAGjgbIwga8wFgYDVR0lAQH/BAwwCgYIKwYBBQUHA
wEw\ndAYDVR0TAQH/BAIwADA0BgNVHQ8BAf8EBAMCA6gwHQYDVR00BBYEFFnFr/+FuXPw\nn02+F2
es0v8/XeGvzkuy55dT07SwhLGTkiuv5L2ADawGQmLnQ7thNUN2P\n-----END CERTIFICAT
E-----\n

```

- d. Copy the returned single-line certificate content. You need to paste this content as the value for `--cckm-certificate`.

- `--public-key-id`: ID of the local (public) key of the RSA key pair generated in [Generate RSA Key Pair](#).

The command prompts for the password. Enter the password of the CCKM Appliance user (`cckm-user`).

Example:

```

./ksctl-linux-amd64 migrations download --cckm-host "127.0.0.1" --cckm-user "admin" --cckm-certificate "-----BEGIN CERTIFICATE-----\nMIIDhjCCAm6gAwIBAgIGCBPpf2sU
MA0GCSqGSIb3DQEBAUAMC8xIDAeBgNVBAMT\nF0NHIENBIFMgb24gcGt2LWnsdXN0ZXIxMQswCQYDVQQ
GEwJUVUzAeFw0yMTAzMDMw\nNzI0MDVaFw0zMtAzMDQwNzI0MDVaMCQxFTATBgNVBAMTDHBrdi1jbHVzdG
VyMTTEL\nMAkGA1UEBhMCVVMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQD0h14c\nnhxamUh7
e0/TzdS016djkaL8x1a/uljy8RediBA2W5Yrmm4sTBx0/PFZ0l+0wyzYe\n7VqsCG3P5+lsTiXw2VLXKT
7qyiil7P8ParCoKt6v5fp6BjtDZzq7q3uG8daZqL4M\nKrAiI8rcGPDgsAex/kSPAB4di1AFjDDNv7SCi
et5kWJq8YeTZEuwYa8rxLi2QEqJ\n4h/mHDqD35jNiSf4FRksWfUwtsY652XIbVpDnwkpSsA/+kEN04Ai
uve9CSvyIBmR\nKaqBwnmYdQZaBJGfKEvjnArtSMMwCULxikQCiuK1BaW9H7f/WQUuNh7xManRb7w\nd
NkhqAhSPyqgjz/TAgMBAAGjgbIwga8wFgYDVR0lAQH/BAwwCgYIKwYBBQUHAwEw\ndAYDVR0TAQH/BAIw
ADA0BgNVHQ8BAf8EBAMCA6gwHQYDVR00BBYEFFnFr/+FuXPw\nn02+F2es0v8/XeGvzkuy55dT07SwhLGT
Kiuv5L2ADawGQmLnQ7thNUN2P\n-----END CERTIFICATE-----\n" --public-key-id "bd3e3bf
a246f470ea6327646b3db359fcb882a6e2a6d4f839c2138569d99e395"

```

Output:

```

CCKM enterprise user password:
File is downloading.
File downloaded successfully: migration_20210621143427.json

```

An export file named `"migration_<TIMESTAMP>.json"` (for example, `migration_20210621143427.json`) is created and downloaded to your machine.

Note

After downloading the migration data (the export file), open the file and check whether it contains a `"containers"` tag at the beginning of the file.

Upload Export File

As a CipherTrust Manager administrator, upload the downloaded export file to CCKM Embedded.

Run the command:

```
./ksctl-linux-amd64 migrations upload --file <export-file-name>
```

Here,

- **--file**: Name of the export file.

Example:

```
./ksctl-linux-amd64 migrations upload --file migration_20210621143427.json
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 21712358,
  "created_at": "2021-05-07T04:55:33.251439962Z",
  "checksum_sha256": "3e853d19e1d4a3687dfcf7bde5fc6bb192c19345462c5f4ae05c481acf26968b",
  "product": "CCKM"
}
```

Note down the "id" value, "26304f13-5626-419b-bd81-459fda7fdb68". It is required when applying the migration data. Now, you need to apply the migration data to CCKM Embedded (refer to [Migrate Complete Data](#)).

Apply Migration Data

The migration data can be applied to CCKM Embedded in two ways:

- [Migrate Data of Set of Containers](#)
- [Migrate Complete Data](#)

Note

- In a clustered CipherTrust Manager environment, apply the migration data on one node only. Migrated data is automatically replicated to other nodes of the cluster.
- After migration, all the source keys from DSM will be migrated to the current domain on the CipherTrust Manager (where the migration is being performed).

- Even if you migrate specific KMS containers, all of the DSM source keys will be migrated.
- After all the source keys are moved to the CipherTrust Manager, you can remove DSM from CCKM Embedded, if needed. Refer to [Deleting DSM Domains](#) and [Deleting a DSM Connection](#).

Migrate Data of Set of Containers

This method requires a list of containers. After the exported file is uploaded to the CipherTrust Manager, you can get the list of containers and select the containers to be migrated.

To migrate the data of a set of containers:

1. List the migrated containers.

Run the command:

```
./ksctl-linux-amd64 migrations containers list -id <export-file-id>
```

Here,

- **--id**: ID of the uploaded export file. Refer to [Upload Export File](#) for the ID of the export file.

Example:

```
./ksctl-linux-amd64 migrations containers list --id 26304f13-5626-419b-bd81-459fda7fdb68
```

Output:

```
{
  "Containers": [
    {
      "id": "b10ba3f2-b835-485d-b09a-07d4c002ff97",
      "type": "aws",
      "name": "805182230944",
      "tenant": "805182230944"
    },
    {
      "id": "0de664f3-7c70-4fae-9213-4e16bb99a4fc",
      "type": "aws",
```

```

        "name": "556782317223",
        "tenant": "556782317223"
    },
    {
        "id": "3f2b09ba-485d-b835-b09a-07d4c002ff97",
        "type": "salesforce",
        "name": "0056A0000011XDvQAM",
        "tenant": "0056A0000011XDvQAM"
    },
    {
        "id": "4f4b11ba-c385-456d-b09a-07d4c002ff97",
        "type": "salesforcesandbox",
        "name": "0056A0000011ABvZCM",
        "tenant": "0056A0000011ABvZCM"
    },
    {
        "id": "0941754d-9cb9-4b32-a241-59f7a4703756",
        "type": "azure",
        "name": "pkv-premium",
        "tenant": "d27d849e-e487-4b0e-a54c-a71e67687d10"
    }
}

```

This command returns the list of containers with their IDs. From this list, you can select the containers to be migrated.

2. Select the desired containers to be migrated. In this example, we will migrate the containers with the IDs `"b10ba3f2-b835-485d-b09a-07d4c002ff97"` and `"0941754d-9cb9-4b32-a241-59f7a4703756"`.
3. Apply the migration to migrate the desired container.

Run the command:

```

./ksctl-linux-amd64 migrations apply --id <export-file-id> --private-key-id <private-key-id> --migrate-cckm-source-keys --containers "<container1-id>,<container2-id>,...,<containerN-id>"

```

Here,

- `--id`: ID of the uploaded export file. Refer to [Upload Export File](#) for the ID of the export file.
- `--private-key-id`: ID of the private key of the RSA key pair. Refer to [Generate RSA Key Pair](#) for the private key ID.

- `--migrate-cckm-source-keys`: Tag to migrate the source keys.
- `--containers`: Comma-separated container IDs.

Example:

```
./ksctl-linux-amd64 migrations apply --id 26304f13-5626-419b-bd81-459fda7fdb68
--private-key-id b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c3
3c1 --migrate-cckm-source-keys --containers "b10ba3f2-b835-485d-b09a-07d4c002f
f97,0941754d-9cb9-4b32-a241-59f7a4703756",
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 22940401,
  "created_at": "2021-05-06T14:19:54.76812635Z",
  "status": "In progress",
  "checksum_sha256": "3386289c00e7e8bf7ad70a5ca12d0852913c0fb3a54f584c28460a
a9cd65383e",
  "product": "CCKM"
}
```

The container migration data is applied. It might take some time to complete. Verify the migration status. Refer to [Check Migration Status](#) for details.

Migrate Complete Data

To migrate the complete data, run the command:

```
./ksctl-linux-amd64 migrations apply --id <export-file-id> --private-key-id <priv
ate-key-id> --migrate-cckm-source-keys
```

Here,

- `--id`: ID of the uploaded export file. Refer to [Upload Export File](#) for the ID of the export file.
- `--private-key-id`: ID of the private key of the RSA key pair. Refer to [Generate RSA Key Pair](#) for the private key ID.
- `--migrate-cckm-source-keys`: Tag to migrate the source keys.

Example:

```
./ksctl-linux-amd64 migrations apply --id 26304f13-5626-419b-bd81-459fda7fdb68 --
private-key-id b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1
--migrate-cckm-source-keys
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 22940401,
  "created_at": "2021-05-06T14:19:54.76812635Z",
  "status": "In progress",
  "checksum_sha256": "3386289c00e7e8bf7ad70a5ca12d0852913c0fb3a54f584c28460aa9c
d65383e",
  "product": "CCKM"
}
```

Check Migration Status

After you have applied the migration data, verify the migration status by running the `ksctl migrations status` command.

Example:

```
./ksctl-linux-amd64 migrations status
```

Output:

```
{
  "id": "9c9149ad-901b-405f-aefd-b279e6257f97",
  "overall_status": "Completed",
  "source": "CCKM",
  "cckm_azure_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_sfmc_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_aws_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_dsm_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 }
}
```

}

In the sample output above, `"overall_status": "Completed"` indicates that migration of CCKM Appliance to CCKM Embedded is successful. Also, verify that migration status of different clouds and key sources such as `"cckm_azure_keys"`, `"cckm_aws_keys"`, and `"cckm_dsm_keys"` is `"Completed"`.

In the output, `"num_ignored"` indicates the number of Azure keys for which blobs are present on the DSM but the keys have been deleted.

After the migration status is `"Completed"`, you can add connection to supported clouds to perform operations on the added data.

After migration, you can see the migrated keys on the Keys page of the CipherTrust Manager GUI. The migrated keys are added with the prefix `Migration-DSM-`.

Add Connection

Add connection to the desired clouds on the CipherTrust Manager. Refer to [Connection Manager](#) for details on connection management.

Update Containers

After the connection is created, update connection in AWS KMS, Azure key vaults, Salesforce organizations, and Google key rings on the Containers tab. After the update, keys in supported clouds can be managed from CCKM Embedded.

For **Salesforce**, you also need to synchronize certificates and update endpoints linked with the organizations.

- **Synchronize certificates:** After migration, the certificates linked with the cache-only keys and schedulers are not synchronized. You need to manually synchronize them on CCKM. Refer to [Prepare the Cache-Only Key Integration](#) for details.
- **Update endpoints:** The endpoint URLs used in Named Credentials for cache-only keys before migration will not work after migration. This is because the Callout URL check performed by Salesforce while fetching the cache-only keys will fail. So, you need to update the endpoint URLs in Named Credential on Salesforce and create the same endpoints on CCKM.

Alternatively, you can create new endpoint URLs on Salesforce and CCKM and use them.

Refer to [Prepare the Cache-Only Key Integration](#) and [Edit a Salesforce Cache-Only Key Endpoint](#) for details.

2. Migrate cloud keys with DSM as key source, move source keys to an External CipherTrust Manager

Migrate Source Keys from DSM to external CipherTrust Manager

To migrate the source keys from DSM to the external CipherTrust Manager:

1. Download the CLI tool (`ksctl`).
2. Log on to the external CipherTrust Manager.


```
ksctl login --user <username> --password <password> --url <External Cipher
Trust Manager URL> --days <number of days> --hours <number of hours> --yes
--domain <domain-name-or-id>
```

3. Generate an RSA key pair and note down the private and public key IDs.

```
ksctl keys create --name <key name> --alg RSA --size 2048
```

Output:

```
{
  "id": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1",
  "uri": "kylo:kylo:vault:keys:rsa-key-v0",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-05-07T03:55:47.519466Z",
  "name": "rsa-key",
  "updatedAt": "2021-05-07T03:55:47.519466Z",
  "usage": "sign",
  "usageMask": 3,
  "meta": {
    "ownerId": "local|5e3b45c6-6f26-4413-9752-e6fd15418a61"
  },
  "version": 0,
  "algorithm": "RSA",
  "size": 4096,
  "unexportable": false,
  "undeletable": false,
  "neverExported": true,
  "neverExportable": false,
  "emptyMaterial": false,
  "publickey": "-----BEGIN PUBLIC KEY-----\nMIICIjANBgkqhkiG9w0BAQEFAAOCAg8A
MIICGkKCAgEAWfd4fDEgdJydUP0kPpAL\nriQW+IpwM9oSte//pv45UXpw0wkag1FbSmEUQQMf02Kd
RW5so+4jrkX44gQhmDTA\namrpjweJa0HzkaqnTkMtCUTFP9nbx33JiWRSYSKqXsEJho+P9SqXz45
uxf7iis5\n4NF0SpZSIYi3C0H4xtJ7hK+6BXbLXBZHVpfQ6LN8p/+WDwcIIkSyWGQxj4V0xTwV\nnfN
BNoQrXvanrEX+nide28vuX1bJ1UzglhUwcFT12VZL8KIrkvicKMwkBNIuuiXgh\nnbtYGBY84ZbPjRE
gaodbaU45vj38/dpusL75Q2hkUdv5mYvTqN+OPVbJrTTQFzGfw\nnM3Pt86iBFfu3XH/ZMH4dbV3HHX
JP7+mHI3c0hUlvojw9hnKygn3fY4Darx/N0yr\nniCp6Sz7FI3sExAAIEF+AJ7zqyXK6a/NGve5gAq
t1w3fn0YIFeD8f6oX0YBFFniu3\n3uX//4WcNdgyTXKXhDsZAtaLqmHv9jIwGZ0pTlj8xefZPbkoDN
ON3uC92b0tzI7F\nn7+Iq0iEf5bg4huU/EJh8emYgU8mPZGpwPtPVUFIKm0Y7EbvHS1C6RIqRE1hnCZ
Aa\nZSMup6LLzZGvk6SM0339c5gDJuS+kGkYK/f0wuWJ7q05m+T/27J1IoNna6JuZ9eL\nnZDMxs7Rq
j4cdezaa3CTV4l8CAWEAAQ==\n-----END PUBLIC KEY-----\n",
```

```

    "defaultIV": "78f83dddc0ee01a2ab3ff579c908a33a",
    "sha1Fingerprint": "878bcd84e81c4170",
    "sha256Fingerprint": "c9c2d321b21d34a3e82460df8839e55f3ebca977766658d830d5
100fb29bed75",
    "objectType": "Private Key",
    "activationDate": "2021-05-07T03:55:43.229267Z",
    "state": "Active",
    "aliases": [
      {
        "alias": "rsa-key",
        "type": "string",
        "index": 0
      }
    ],
    "links": [
      {
        "id": "6dc578f7-1864-43a7-899a-5035d54f1772",
        "uri": "kylo:kylo:vault:links:6dc578f7-1864-43a7-899a-5035d54f177
2",
        "account": "kylo:kylo:admin:accounts:kylo",
        "application": "ncryptify:gemalto:admin:apps:kylo",
        "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
        "createdAt": "2021-05-07T03:55:47.531762Z",
        "updatedAt": "2021-05-07T03:55:47.531762Z",
        "type": "publicKey",
        "source": "kylo:kylo:vault:keys:rsa-key-v0",
        "sourceID": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd
202bd4c33c1",
        "target": "kylo:kylo:vault:keys:rsa-key-pub-v0",
        "targetID": "bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c213
8569d99e395",
        "index": 0
      }
    ],
    "uuid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3",
    "muid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3e129ed33-317d-4584-9cbf-d0e88
2f58fca"
  }

```

4. Download the migration data from the CCKM Appliance.

```

ksctl migrations download --cckm-host <cckm-host> --cckm-user <cckm-user>
--cckm-certificate <cckm-certificate> --public-key-id <public-key-id>

```

Output:

```
CCKM enterprise user password:
File is downloading.
File downloaded successfully: migration_20210621143427.json
```

5. Upload the migration data to the CipherTrust Manager.

```
ksctl migrations upload --file <downloaded migrations filename>
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 21712358,
  "created_at": "2021-05-07T04:55:33.251439962Z",
  "checksum_sha256": "3e853d19e1d4a3687dfcf7bde5fc6bb192c19345462c5f4ae05c481acf26968b",
  "product": "CCKM"
}
```

6. List all containers from the uploaded migration file.

```
ksctl migrations containers list --id <migrations file resource id>
```

Output:

```
{
  "Containers": [
    {
      "id": "b10ba3f2-b835-485d-b09a-07d4c002ff97",
      "type": "aws",
      "name": "805182230944",
      "tenant": "805182230944"
    },
    {
      "id": "0de664f3-7c70-4fae-9213-4e16bb99a4fc",
      "type": "aws",
      "name": "556782317223",
      "tenant": "556782317223"
    }
  ]
}
```

```

        "id": "3f2b09ba-485d-b835-b09a-07d4c002ff97",
        "type": "salesforce",
        "name": "0056A0000011XDvQAM",
        "tenant": "0056A0000011XDvQAM"
    },
    {
        "id": "4f4b11ba-c385-456d-b09a-07d4c002ff97",
        "type": "salesforcesandbox",
        "name": "0056A0000011ABvZCM",
        "tenant": "0056A0000011ABvZCM"
    },
    {
        "id": "0941754d-9cb9-4b32-a241-59f7a4703756",
        "type": "azure",
        "name": "pkv-premium",
        "tenant": "d27d849e-e487-4b0e-a54c-a71e67687d10"
    }
}

```

7. Apply migration.

Note

- In a clustered CipherTrust Manager environment, apply the migration data on one node only. Migrated data is automatically replicated to other nodes of the cluster.
- After migration, all the source keys from DSM will be migrated to the current domain on the CipherTrust Manager (where the migration is being performed).
- After all the source keys are moved to the CipherTrust Manager, you can remove DSM from CCKM Embedded, if needed. Refer to [Deleting DSM Domains](#) and [Deleting a DSM Connection](#).

To apply the migration data, run the command:

```
ksctl migrations apply --id <migrations file resource id> --private-key-id
<RSA private key id> --migrate-only-cckm-source-keys
```

Output:

```

{
    "id": "26304f13-5626-419b-bd81-459fda7fdb68",

```

```

    "file_size": 22940401,
    "created_at": "2021-05-06T14:19:54.76812635Z",
    "status": "In progress",
    "checksum_sha256": "3386289c00e7e8bf7ad70a5ca12d08529    13c0fb3a54f584c284
60aa9cd65383e",
    "product": "CCKM"
  }

```

8. Check the migration status.

```
ksctl migrations status
```

Output:

```

{
  "id": "9c9149ad-901b-405f-aefd-b279e6257f97",
  "overall_status": "Completed",
  "source": "CCKM",
  "cckm_azure_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_sfdc_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_aws_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_dsm_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 }
}

```

Note

- Run the `migrations apply` command with the `migrate-only-cckm-source-keys` flag. Only the source keys from DSM will be migrated to the CipherTrust Manager without migrating cloud keys.
- The `migrate-only-cckm-source-keys` flag is optional and is applicable to the CCKM Appliance migrations only.

Migrate Cloud Keys to the CipherTrust Manager

Prerequisites for Migration

- Source keys from DSM are migrated to the external CipherTrust Manager.
- After migrating the DSM source keys to the external CipherTrust Manager, assign the **Key Admins** permission to the client CipherTrust Manager. Refer to [Add the Client](#) for details.

Migration Steps

To migrate the cloud keys to the CipherTrust Manager:

On the CipherTrust Manager

1. Configure the *external* CipherTrust Manager as a key source.
 - a. Add the external CipherTrust Manager connection. Refer to [CM Connection/External CM](#).
 - b. Add the *external* CipherTrust Manager domains where the DSM source keys were migrated.
 - c. Refresh the CipherTrust domains added to the CipherTrust Manager.
2. Download the CLI tool (`ksctl`).
3. Log on to the CipherTrust Manager.

```
ksctl login --user <username> --password <password> --url <CipherTrust Manager URL> --days <number of days> --hours <number of hours> --yes --domain <domain-name-or-id>
```

4. Generate an RSA key pair and note down the private and public key IDs.

```
ksctl keys create --name <key name> --alg RSA --size 2048
```

Output:

```
{
  "id": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1",
  "uri": "kylo:kylo:vault:keys:rsa-key-v0",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-05-07T03:55:47.519466Z",
  "name": "rsa-key",
  "updatedAt": "2021-05-07T03:55:47.519466Z",
  "usage": "sign",
  "usageMask": 3,
  "meta": {
    "ownerId": "local|5e3b45c6-6f26-4413-9752-e6fd15418a61"
  },
  "version": 0,
  "algorithm": "RSA",
  "size": 4096,
  "unexportable": false,
  "undeletable": false,
  "neverExported": true,
  "neverExportable": false,
  "emptyMaterial": false,
  "publickey": "-----BEGIN PUBLIC KEY-----\nMIICIjANBgkqhkiG9w0BAQEFAAOCAg8A\nMIICGkKCAgEAWfd4fDEgdJydUP0kPpAL\nnriQW+IpwM9oSte//pv45UXpw0wkag1FbSmEUQQMf02Kd\nRW5so+4jrKX44gQhmdTA\nnamrpijweJa0HzkaqnTkMtCuTFP9nbx33JiWRSYSKqXsEJho+P9SqXz45\nuxf7iis5\nn4NF0SpZSIYi3C0H4xtJ7hK+6BXbLXBZHVpfQ6LN8p/+WDwcIIkSyWGQxj4V0xTwW\n\nnfNBNoQrXvanrEX+nide28vuX1bJ1UzglhUwcFT12VZL8KirKviCKMwkBNiuiXgh\n\nnbtYGBY84ZbPjREgaodbaU45vj38/dpusL75Q2hkUdv5mYvtQn+OPVbJrTTQFzGfw\n\nnM3Pt86iBFfu3XH/ZMH4dbV3HHXJP7+mHI3c0hUlvjwx9hnKygn3fY4Darx/N0yr\n\n\nniCp6Sz7FI3sExAAIEF+AJ7zqyXK6a/NGve5gAqt1w3fn0YIFeD8f6oX0YBFFniu3\n\n\nn3uX//4WcNdgyTXKXhDsZAtaLqmHv9jIwGZ0pTlj8xefZPbkoDNON3uC92b0tzI7F\n\n\nn7+Iq0iEf5bg4huU/EJh8emYgU8mPZGpwPtPVUFiKmOY7EbVHS1C6RIqRE1hnCZAa\n\n\nnZSMup6LLzZGvk6SM0339c5gDJuS+kGkYK/f0WuWJ7q05m+T/27J1IoNna6JuZ9el\n\n\nnZDMxs7Rqj4cdezaa3CTV4l8CAwEAAQ==\n\n\n-----END PUBLIC KEY-----\n",
  "defaultIV": "78f83dddc0ee01a2ab3ff579c908a33a",
  "sha1Fingerprint": "878bcd84e81c4170",
  "sha256Fingerprint": "c9c2d321b21d34a3e82460df8839e55f3ebca977766658d830d5100fb29bed75",
  "objectType": "Private Key",
  "activationDate": "2021-05-07T03:55:43.229267Z",
  "state": "Active",
  "aliases": [
    {

```

```

        "alias": "rsa-key",
        "type": "string",
        "index": 0
    },
    ],
    "links": [
        {
            "id": "6dc578f7-1864-43a7-899a-5035d54f1772",
            "uri": "kylo:kylo:vault:links:6dc578f7-1864-43a7-899a-5035d54f177
2",
            "account": "kylo:kylo:admin:accounts:kylo",
            "application": "ncryptify:gemalto:admin:apps:kylo",
            "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
            "createdAt": "2021-05-07T03:55:47.531762Z",
            "updatedAt": "2021-05-07T03:55:47.531762Z",
            "type": "publicKey",
            "source": "kylo:kylo:vault:keys:rsa-key-v0",
            "sourceID": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd
202bd4c33c1",
            "target": "kylo:kylo:vault:keys:rsa-key-pub-v0",
            "targetID": "bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c213
8569d99e395",
            "index": 0
        }
    ],
    "uuid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3",
    "muid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3e129ed33-317d-4584-9cbf-d0e88
2f58fca"
}

```

5. Download the migration data from the CCKM Appliance.

```

ksctl migrations download --cckm-host <cckm-host> --cckm-user <cckm-user>
--cckm-certificate <cckm-certificate> --public-key-id <public-key-id>

```

Output:

```

CCKM enterprise user password:
File is downloading.
File downloaded successfully: migration_20210621143427.json

```

6. Upload the migration data to the CipherTrust Manager.


```
ksctl migrations upload --file <downloaded migrations filename>
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 21712358,
  "created_at": "2021-05-07T04:55:33.251439962Z",
  "checksum_sha256": "3e853d19e1d4a3687dfcf7bde5fc6bb192c19345462c5f4ae05c481acf26968b",
  "product": "CCKM"
}
```

7. List all containers from the uploaded migration file.

```
ksctl migrations containers list --id <migrations file resource id>
```

Output:

```
{
  "Containers": [
    {
      "id": "b10ba3f2-b835-485d-b09a-07d4c002ff97",
      "type": "aws",
      "name": "805182230944",
      "tenant": "805182230944"
    },
    {
      "id": "0de664f3-7c70-4fae-9213-4e16bb99a4fc",
      "type": "aws",
      "name": "556782317223",
      "tenant": "556782317223"
    },
    {
      "id": "3f2b09ba-485d-b835-b09a-07d4c002ff97",
      "type": "salesforce",
      "name": "0056A0000011XDvQAM",
      "tenant": "0056A0000011XDvQAM"
    },
    {
      "id": "4f4b11ba-c385-456d-b09a-07d4c002ff97",
      "type": "salesforcesandbox",
      "name": "0056A0000011ABvZCM",

```

```

        "tenant": "0056A0000011ABvZCM"
      },
      {
        "id": "0941754d-9cb9-4b32-a241-59f7a4703756",
        "type": "azure",
        "name": "pkv-premium",
        "tenant": "d27d849e-e487-4b0e-a54c-a71e67687d10"
      }
    ]
  }
}

```

8. Apply migration.

Note

In a clustered CipherTrust Manager environment, apply the migration data on one node only. Migrated data is automatically replicated to other nodes of the cluster.

To apply the migration data, run the command:

```
ksctl migrations apply --id <migrations file resource id> --private-key-id
<RSA private key id> --migrate-cloudkeys-with-externalcm
```

Output:

```

{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 22940401,
  "created_at": "2021-05-06T14:19:54.76812635Z",
  "status": "In progress",
  "checksum_sha256": "3386289c00e7e8bf7ad70a5ca12d0852913c0fb3a54f584c28460a
a9cd65383e",
  "product": "CCKM"
}

```

9. Check the migration status.

```
ksctl migrations status
```

Output:

```
{
  "id": "9c9149ad-901b-405f-aefd-b279e6257f97",
  "overall_status": "Completed",
  "source": "CCKM",
  "cckm_azure_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_sfmc_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_aws_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_dsm_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 }
}
```

Note

- Run the `migrations apply` command with the `migrate-cloudkeys-with-externalcm` flag. Only the cloud keys will be migrated to the CipherTrust Manager without migrating the DSM source keys.
- The `migrate-cloudkeys-with-externalcm` flag is optional and is applicable to the CCKM Appliance migrations only.
- If the migration data contains Salesforce schedules, then after migration, update the external CipherTrust Manager domain in the Salesforce schedules.

3. Migrate cloud keys but keep using DSM as key source

This section provides instructions to migrate *only* the cloud keys from a CCKM Appliance to a CipherTrust Manager. Before proceeding, read the [prerequisites](#).

Note

The user who performs the migration becomes the owner of the migrated keys.

The steps above apply to the root domain only. To perform migration on a child domain, refer to [Migration from CCKM Appliance to Child Domain](#).

Generate RSA Key Pair

Exporting migration data from CCKM Appliance requires an RSA key pair (public and private) on the CipherTrust Manager. The public key is used to encrypt the data while the private key is used to decrypt the exported data.

To generate an RSA key pair, run the `ksctl keys create` command:

```
./ksctl-linux-amd64 keys create --name <rsa-key-name> --alg <key-algorithm> --size <key-size>
```

Here,

- `--name`: Name for RSA key pair.
- `--alg`: Algorithm for the RSA key pair.
- `--size`: Size for the key pair.

Example:

```
./ksctl-linux-amd64 keys create --name rsa-key --alg RSA --size 4096
```

Output:

```
{
  "id": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1",
  "uri": "kylo:kylo:vault:keys:rsa-key-v0",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-05-07T03:55:47.519466Z",
  "name": "rsa-key",
  "updatedAt": "2021-05-07T03:55:47.519466Z",
  "usage": "sign",
  "usageMask": 3,
  "meta": {
    "ownerId": "local|5e3b45c6-6f26-4413-9752-e6fd15418a61"
  },
  "version": 0,
  "algorithm": "RSA",
  "size": 4096,
```

```

    "unexportable": false,
    "undeletable": false,
    "neverExported": true,
    "neverExportable": false,
    "emptyMaterial": false,
    "publickey": "-----BEGIN PUBLIC KEY-----\nMIICiJANBgkqhkiG9w0BAQEFAAOCAg8AMII
CCgKCAGeAwfd4fDEgdJydUP0kPpAL\nriQW+IpWm9oSte//pv45UXpw0wkag1FbSmEUQQMf02KdRW5s
o+4jrkX44gQhmDTA\nnamrpijweJa0HzkaqnTkMtCutFP9nbx33JiWRSYSKqXsEJho+P9SqXz45uxf7iis
5\n4NF0SpZSIYi3COH4xtJ7hK+6BXbLXBZHVpfQ6LN8p/+WDwcIIkSyWGQxj4V0xTwV\nnfNBNoQrXvanr
EX+nide28vuX1bJ1UzghUwcFT12VZL8KIrkviCKMwkBNiIuuiXgh\nbntYGBY84ZbPjREgaodbaU45vj3
8/dpusL75Q2hkUdv5mYvTqN+OPVbJrTTQFzGfw\nmM3Pt86iBFfu3XH/ZMH4dbV3HHXJP7+mHI3c0hUlvo
jwx9hnKygn3fY4Darx/N0yr\nniCp6Sz7FI3sExAAIEF+AJ7zqyXK6a/NGve5gAqt1w3fn0YIFeD8f6oX0
YBFFniu3\n3uX//4WcNdgyTXKXhDsZAtaLqmHv9jIwGZ0pTlj8xefZPbkoDN0N3uC92b0tzI7F\nn7+Iq0
iEf5bg4huU/EJh8emYgU8mPZGpwPtPVUFiKm0Y7EbvHS1C6RIqRE1hnCZAa\nnZSMup6LLzZGvk6SM0339
c5gDJuS+kGkYK/f0wuWJ7q05m+T/27J1IoNna6JuZ9el\nnZDMxs7Rqj4cdezaa3CTV4l8CAwEAA
Q==\n-----END PUBLIC KEY-----\n",
    "defaultIV": "78f83dddc0ee01a2ab3ff579c908a33a",
    "sha1Fingerprint": "878bcd84e81c4170",
    "sha256Fingerprint": "c9c2d321b21d34a3e82460df8839e55f3ebca977766658d830d5100
fb29bed75",
    "objectType": "Private Key",
    "activationDate": "2021-05-07T03:55:43.229267Z",
    "state": "Active",
    "aliases": [
      {
        "alias": "rsa-key",
        "type": "string",
        "index": 0
      }
    ],
    "links": [
      {
        "id": "6dc578f7-1864-43a7-899a-5035d54f1772",
        "uri": "kylo:kylo:vault:links:6dc578f7-1864-43a7-899a-5035d54f1772",
        "account": "kylo:kylo:admin:accounts:kylo",
        "application": "ncryptify:gernalto:admin:apps:kylo",
        "devAccount": "ncryptify:gernalto:admin:accounts:gernalto",
        "createdAt": "2021-05-07T03:55:47.531762Z",
        "updatedAt": "2021-05-07T03:55:47.531762Z",
        "type": "publicKey",
        "source": "kylo:kylo:vault:keys:rsa-key-v0",
        "sourceID": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202
bd4c33c1",
        "target": "kylo:kylo:vault:keys:rsa-key-pub-v0",
        "targetID": "bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c213856
9d99e395",
        "index": 0
      }
    ],
    "uuid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3",
    "muid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3e129ed33-317d-4584-9cbf-d0e882f5
8fca"
  }

```

In the sample output above, "sourceID":

"b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1" is the private key ID. The "targetID":

"bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c2138569d99e395" under "links" is

the public key ID, which is used to download the export file.

Download Export File

As a CipherTrust Manager administrator, download the migration data (the export file) from the CCKM Appliance. This file contains migration data to be uploaded to CCKM Embedded.

Run the command:

```
./ksctl-linux-amd64 migrations download --cckm-host "<cckm-ip-or-hostname>" --cckm-user "<cckm-user>" --cckm-certificate "<certificate-file-content>" --public-key-id "<public-key-id>"
```

Here,

- `--cckm-host`: IP address or hostname of the CCKM Appliance.
- `--cckm-user`: Admin user of the CCKM Appliance.
- `--cckm-certificate`: SSL certificate downloaded from the CCKM Appliance.

By default, the content of the SSL certificate spans over multiple new lines. However, to input the certificate as an inline argument, the certificate content needs to be converted into the single-line format where every new line is represented by `\n`.

To convert the certificate into the single-line format:

- a. Download the certificate in a file (say `certificate.cert`).
- b. Store the certificate file at a location on your machine, for example, at `/root/cert/`.
- c. Run the following command:

```
awk 'NF {sub(/\r/, ""); printf "%s\\n",$0;}' <full-path-to-certificate-file>
```

For example, to convert your certificate file (`/root/cert/certificate.cert`), run the command:

```
awk 'NF {sub(/\r/, ""); printf "%s\\n",$0;}' /root/cert/certificate.cert
```

The above command returns the certificate in the single-line format, as shown below:

```
-----BEGIN CERTIFICATE-----\nMIIDhjCCAm6gAwIBAgIGCBPpf2sUMA0GCSqGSIB3DQEBDAU\nAMC8xIDAeBgNVBAMT\nF0NHIENBIFMgb24gcGt2LWNsdXN0ZXIwMQswCQYDVQQGEwJVUzAeFw0yM
```

```
TAzMMDw\NzI0MDVaFw0zMTAzMDQwNzI0MDVaMCQxFTATBgNVBAMTDHBrdi1jbHVzdGVyMTEl\nM
AkGA1UEBhMCVVMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQD0h14c\nhxamUh7e0/T
zdS016djkaL8x1a/uljy8RediBA2W5Yrmm4sTBx0/PFZ0l+OwyzYe\n7VqsCG3P5+lsTiXw2VLXK
T7qyiil7P8ParCoKt6v5fp6BjtDZzq7q3uG8daZqL4M\nKrAiI8rcGPDgsAex/kSPAB4di1AFjDD
Nv7SCiet5kWJq8YeTZEuwYa8rxLi2QEqJ\n4h/mHDqD35jNiSf4FRksWfUwtsY652XIbVpDnwkp
sA/+kEN04Aiuve9CSvyIBmR\nKaqBwnmYdQZaBJGFKEvjnArtSMMwCULxikQCiuK1BaW9H7f/WQ
UuNh7xManRb7w\ndNkhqAhSPyqgjz/TAgMBAAGjgbIwga8wFgYDVR0LAQH/BAwwCgYIKwYBBQUHA
wEw\ndAYDVR0TAQH/BAIwADA0BgNVHQ8BAf8EBAMCA6gwHQYDVR00BBYEFFnFr/+FuXPw\n02+F2
es0v8/XeGvzkuy55dT07SwhLGTKiuv5L2ADawGQmLNQ7thNUN2P\n-----END CERTIFICAT
E-----\n
```

d. Copy the returned single-line certificate content. You need to paste this content as the value for `--cckm-certificate`.

- `--public-key-id`: ID of the local (public) key of the RSA key pair generated in [Generate RSA Key Pair](#).

The command prompts for the password. Enter the password of the CCKM Appliance user (`cckm-user`).

Example:

```
./ksctl-linux-amd64 migrations download --cckm-host "127.0.0.1" --cckm-user "admin" --cckm-certificate "-----BEGIN CERTIFICATE-----\nMIIDhjCCAm6gAwIBAgIGCBPpf2sU
MA0GCSqGSIb3DQEBAUAMC8xIDAeBgNVBAMT\nF0NHIENBIFMgb24gcGt2LWnsdXN0ZXIxMQswCQYDVQQ
GEwJVUzAeFw0yMTAzMDMw\NzI0MDVaFw0zMTAzMDQwNzI0MDVaMCQxFTATBgNVBAMTDHBrdi1jbHVzdG
VyMTEl\nMAkGA1UEBhMCVVMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQD0h14c\nhxamUh7
e0/TzdS016djkaL8x1a/uljy8RediBA2W5Yrmm4sTBx0/PFZ0l+OwyzYe\n7VqsCG3P5+lsTiXw2VLXKT
7qyiil7P8ParCoKt6v5fp6BjtDZzq7q3uG8daZqL4M\nKrAiI8rcGPDgsAex/kSPAB4di1AFjDDNv7SCi
et5kWJq8YeTZEuwYa8rxLi2QEqJ\n4h/mHDqD35jNiSf4FRksWfUwtsY652XIbVpDnwkpSA/+kEN04Ai
uve9CSvyIBmR\nKaqBwnmYdQZaBJGFKEvjnArtSMMwCULxikQCiuK1BaW9H7f/WQUuNh7xManRb7w\nd
NkhqAhSPyqgjz/TAgMBAAGjgbIwga8wFgYDVR0LAQH/BAwwCgYIKwYBBQUHAwEw\ndAYDVR0TAQH/BAIw
ADA0BgNVHQ8BAf8EBAMCA6gwHQYDVR00BBYEFFnFr/+FuXPw\n02+F2es0v8/XeGvzkuy55dT07SwhLGT
Kiuv5L2ADawGQmLNQ7thNUN2P\n-----END CERTIFICATE-----\n" --public-key-id "bd3e3bf
a246f470ea6327646b3db359fcb882a6e2a6d4f839c2138569d99e395"
```

Output:

```
CCKM enterprise user password:
File is downloading.
File downloaded successfully: migration_20210621143427.json
```

An export file named `"migration_<TIMESTAMP>.json"` (for example, `migration_20210621143427.json`) is created and downloaded to your machine.

Note

After downloading the migration data (the export file), open the file and check whether it contains a

"containers" tag at the beginning of the file.

Upload Export File

As a CipherTrust Manager administrator, upload the downloaded export file to CCKM Embedded.

Run the command:

```
./ksctl-linux-amd64 migrations upload --file <export-file-name>
```

Here,

- `--file`: Name of the export file.

Example:

```
./ksctl-linux-amd64 migrations upload --file migration_20210621143427.json
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 21712358,
  "created_at": "2021-05-07T04:55:33.251439962Z",
  "checksum_sha256": "3e853d19e1d4a3687dfcf7bde5fc6bb192c19345462c5f4ae05c481acf26968b",
  "product": "CCKM"
}
```

Note down the "id" value, "26304f13-5626-419b-bd81-459fda7fdb68". It is required when applying the migration data. Now, you need to apply the migration data to CCKM Embedded (refer to [Migrate Complete Data](#)).

Apply Migration Data

The migration data can be applied to CCKM Embedded in two ways:

- [Migrate Data of Set of Containers](#)
- [Migrate Complete Data](#)

Note

In a clustered CipherTrust Manager environment, apply the migration data on one node only. Migrated data is automatically replicated to other nodes of the cluster.

Migrate Data of Set of Containers

This method requires a list of containers. After the exported file is uploaded to the CipherTrust Manager, you can get the list of containers and select the containers to be migrated.

To migrate the data of a set of containers:

1. List the migrated containers.

Run the command:

```
./ksctl-linux-amd64 migrations containers list -id <export-file-id>
```

Here,

- **-id**: ID of the uploaded export file. Refer to [Upload Export File](#) for the ID of the export file.

Example:

```
./ksctl-linux-amd64 migrations containers list --id 26304f13-5626-419b-bd81-459fda7fdb68
```

Output:

```
{
  "Containers": [
    {
      "id": "b10ba3f2-b835-485d-b09a-07d4c002ff97",
      "type": "aws",
      "name": "805182230944",
      "tenant": "805182230944"
    },
    {
      "id": "0de664f3-7c70-4fae-9213-4e16bb99a4fc",
      "type": "aws",
      "name": "556782317223",

```

```

        "tenant": "556782317223"
    },
    {
        "id": "3f2b09ba-485d-b835-b09a-07d4c002ff97",
        "type": "salesforce",
        "name": "0056A0000011XDvQAM",
        "tenant": "0056A0000011XDvQAM"
    },
    {
        "id": "4f4b11ba-c385-456d-b09a-07d4c002ff97",
        "type": "salesforcesandbox",
        "name": "0056A0000011ABvZCM",
        "tenant": "0056A0000011ABvZCM"
    },
    {
        "id": "0941754d-9cb9-4b32-a241-59f7a4703756",
        "type": "azure",
        "name": "pkv-premium",
        "tenant": "d27d849e-e487-4b0e-a54c-a71e67687d10"
    }
}

```

This command returns the list of containers with their IDs. From this list, you can select the containers to be migrated.

2. Select the desired containers to be migrated. In this example, we will migrate the containers with the IDs `"b10ba3f2-b835-485d-b09a-07d4c002ff97"` and `"0941754d-9cb9-4b32-a241-59f7a4703756"`.
3. Apply the migration to migrate the desired container.

Run the command:

```

./ksctl-linux-amd64 migrations apply --id <export-file-id> --private-key-id <private-key-id> --containers "<container1-id>,<container2-id>,...,<containerN-id>"

```

Here,

- `--id`: ID of the uploaded export file. Refer to [Upload Export File](#) for the ID of the export file.
- `--private-key-id`: ID of the private key of the RSA key pair. Refer to [Generate RSA Key Pair](#) for the private key ID.

- `--containers`: Comma-separated container IDs.

Example:

```
./ksctl-linux-amd64 migrations apply --id 26304f13-5626-419b-bd81-459fda7fdb68
--private-key-id b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c3
3c1 --containers "b10ba3f2-b835-485d-b09a-07d4c002ff97,0941754d-9cb9-4b32-a24
1-59f7a4703756",
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 22940401,
  "created_at": "2021-05-06T14:19:54.76812635Z",
  "status": "In progress",
  "checksum_sha256": "3386289c00e7e8bf7ad70a5ca12d0852913c0fb3a54f584c28460a
a9cd65383e",
  "product": "CCKM"
}
```

The container migration data is applied. It might take some time to complete. Verify the migration status. Refer to [Check Migration Status](#) for details.

Migrate Complete Data

To migrate the complete data, run the command:

```
./ksctl-linux-amd64 migrations apply --id <export-file-id> --private-key-id <priv
ate-key-id>
```

Here,

- `--id`: ID of the uploaded export file. Refer to [Upload Export File](#) for the ID of the export file.
- `--private-key-id`: ID of the private key of the RSA key pair. Refer to [Generate RSA Key Pair](#) for the private key ID.

Example:

```
./ksctl-linux-amd64 migrations apply --id 26304f13-5626-419b-bd81-459fda7fdb68 --
private-key-id b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 22940401,
  "created_at": "2021-05-06T14:19:54.76812635Z",
  "status": "In progress",
  "checksum_sha256": "3386289c00e7e8bf7ad70a5ca12d0852913c0fb3a54f584c28460aa9c
d65383e",
  "product": "CCKM"
}
```

Check Migration Status

After you have applied the migration data, verify the migration status by running the `ksctl migrations status` command.

Example:

```
./ksctl-linux-amd64 migrations status
```

Output:

```
{
  "id": "9c9149ad-901b-405f-aefd-b279e6257f97",
  "overall_status": "Completed",
  "source": "CCKM",
  "cckm_azure_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_sfmc_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_aws_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_dsm_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 }
}
```

In the sample output above, `"overall_status": "Completed"` indicates that migration of CCKM Appliance to CCKM Embedded is successful. Also, verify that migration status of different clouds and key sources such as `"cckm_azure_keys"`, `"cckm_aws_keys"`, and `"cckm_dsm_keys"` is `"Completed"`.

In the output, **"num_ignored"** indicates the number of Azure keys for which blobs are present on the DSM but the keys have been deleted.

After the migration status is **"Completed"**, you can add connection to supported clouds and key sources to perform operations on the added data.

Add DSM Cluster Nodes

If your DSM was in a cluster before migration, then after migration, the DSM connection information will contain only the IP address or hostname with which your CCKM Appliance was configured.

To add other DSM cluster nodes, update the migrated DSM connection on the CipherTrust Manager. Refer to [Adding Node in an Existing DSM Connection](#) for details.

Add Connection

Add connection to the desired clouds on the CipherTrust Manager. Refer to [Connection Manager](#) for details on connection management.

Update Containers

After the connection is created, update connection in AWS KMS, Azure key vaults, Salesforce organizations, and Google key rings on the Containers tab. After the update, keys in supported clouds and key sources can be managed from CCKM Embedded.

For **Salesforce**, you also need to synchronize certificates and update endpoints linked with the organizations.

- **Synchronize certificates:** After migration, the certificates linked with the cache-only keys and schedulers are not synchronized. You need to manually synchronize them on CCKM. Refer to [Prepare the Cache-Only Key Integration](#) for details.
- **Update endpoints:** The endpoint URLs used in Named Credentials for cache-only keys before migration will not work after migration. This is because the Callout URL check performed by Salesforce while fetching the cache-only keys will fail. So, you need to update the endpoint URLs in Named Credential on Salesforce and create the same endpoints on CCKM.

Alternatively, you can create new endpoint URLs on Salesforce and CCKM and use them.

Refer to [Prepare the Cache-Only Key Integration](#) and [Edit a Salesforce Cache-Only Key Endpoint](#) for details.

Note

DSM is EOL as of June 2024 and its support as a key source is being deprecated. Therefore, after migrating the cloud keys, it's recommended to migrate the DSM source keys to CipherTrust Manager. Follow the instructions provided in [Migrate DSM Source Keys](#).

4. Migrate cloud keys when CipherTrust Manager is used as key source with CCKM Appliance

This section describes steps to migrate both cloud keys and the CipherTrust Manager source keys from CCKM Appliance to the CipherTrust Manager. Before proceeding, read the [prerequisites](#).

Note

- The user who performs the migration becomes the owner of the migrated keys.
- Migration of the CipherTrust Manager source keys from CCKM Appliance should be done on the same CipherTrust Manager with which the CCKM Appliance is configured.

The steps below apply to root domain only. To perform migration on a child domain, refer to [Migration from CCKM Appliance to Child Domain](#).

Generate RSA Key Pair

Exporting migration data from CCKM Appliance requires an RSA key pair (public and private) on the CipherTrust Manager. The public key is used to encrypt the data while the private key is used to decrypt the exported data.

To generate an RSA key pair, run the `ksctl keys create` command:

```
./ksctl-linux-amd64 keys create --name <rsa-key-name> --alg <key-algorithm> --size <key-size>
```

Here,

- `--name`: Name for RSA key pair.
- `--alg`: Algorithm for the RSA key pair.
- `--size`: Size for the key pair.

Example:

```
./ksctl-linux-amd64 keys create --name rsa-key --alg RSA --size 4096
```

Output:

```
{
  "id": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1",
  "uri": "kylo:kylo:vault:keys:rsa-key-v0",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-05-07T03:55:47.519466Z",
```

```

"name": "rsa-key",
"updatedAt": "2021-05-07T03:55:47.519466Z",
"usage": "sign",
"usageMask": 3,
"meta": {
  "ownerId": "local|5e3b45c6-6f26-4413-9752-e6fd15418a61"
},
"version": 0,
"algorithm": "RSA",
"size": 4096,
"unexportable": false,
"undeletable": false,
"neverExported": true,
"neverExportable": false,
"emptyMaterial": false,
"publickey": "-----BEGIN PUBLIC KEY-----\nMIICIJANBgkqhkiG9w0BAQEFAAOCAg8AMII
CCgKCAgEAWfd4fDEgdJydUP0kPpAL\nriQW+IpwM9oSte//pv45UXpw0wkag1FbSmEUQQMf02KdRW5s
o+4jrKX44gQhmDTA\nnamrpijweJa0HzkaqnTkMtCutFP9nbx33JiWRSYSKqXsEJho+P9SqXz45uxf7iis
5\n4NF0SpZSIYi3COH4xtJ7hK+6BXbLXBZHVpfQ6LN8p/+wDwcIIkSyWGQxj4V0xTwV\nnfNBNoQrXvanr
EX+nide28vuX1bJ1UzghUwcFT12VZL8KIrkviCKMwkBNiuiXgh\nnbtYGBY84ZbPjREgaodbaU45vj3
8/dpusL75Q2hkUdv5mYvTqN+OPVbJrTTQFzGfw\nmM3Pt86iBFfu3XH/ZMH4dbV3HHXJP7+mHI3c0hUlvo
jwx9hnKygn3fY4Darx/N0yr\nniCp6Ssz7FI3sExAAIEF+AJ7zqyXK6a/NGve5gAqt1w3fn0YIFeD8f6oX0
YBFFniu3\n3uX//4WcNdgyTXKXhDsZAtaLqmHv9jIwGZ0pTlj8xefZPbkoDN0N3uC92b0tzI7F\nn7+Iq0
iEf5bg4huU/EJh8emYgU8mPZGpwPtPVUFiKm0Y7EbvHS1C6RIqRE1hnCZAa\nnZSMup6LLzZGvk6SM0339
c5gDJuS+kGkYK/f0WuWJ7q05m+T/27J1IoNna6JuZ9e\nnZDMxs7Rqj4cdezaa3CTV4l8CAwEAA
Q==\n-----END PUBLIC KEY-----\n",
"defaultIV": "78f83dddc0ee01a2ab3ff579c908a33a",
"sha1Fingerprint": "878bcd84e81c4170",
"sha256Fingerprint": "c9c2d321b21d34a3e82460df8839e55f3ebca977766658d830d5100
fb29bed75",
"objectType": "Private Key",
"activationDate": "2021-05-07T03:55:43.229267Z",
"state": "Active",
"aliases": [
  {
    "alias": "rsa-key",
    "type": "string",
    "index": 0
  }
],
"links": [
  {
    "id": "6dc578f7-1864-43a7-899a-5035d54f1772",
    "uri": "kylo:kylo:vault:links:6dc578f7-1864-43a7-899a-5035d54f1772",
    "account": "kylo:kylo:admin:accounts:kylo",
    "application": "ncryptify:gemalto:admin:apps:kylo",
    "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
    "createdAt": "2021-05-07T03:55:47.531762Z",
    "updatedAt": "2021-05-07T03:55:47.531762Z",
    "type": "publicKey",
    "source": "kylo:kylo:vault:keys:rsa-key-v0",
    "sourceID": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202
bd4c33c1",
    "target": "kylo:kylo:vault:keys:rsa-key-pub-v0",
    "targetID": "bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c213856
9d99e395",
    "index": 0
  }
],

```

```

    "uuid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3",
    "muid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3e129ed33-317d-4584-9cbf-d0e882f58fca"
  }

```

In the sample output above, "sourceID":

"b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1" is the private key ID. The "targetID":

"bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c2138569d99e395" under "links" is the public key ID, which is used to download the export file.

Download Export File

As a CipherTrust Manager administrator, download the migration data (the export file) from the CCKM Appliance. This file contains migration data to be uploaded to CCKM Embedded.

Run the command:

```

./ksctl-linux-amd64 migrations download --cckm-host "<cckm-ip-or-hostname>" --cckm-user "<cckm-user>" --cckm-certificate "<certificate-file-content>" --public-key-id "<public-key-id>"

```

Here,

- **--cckm-host**: IP address or hostname of the CCKM Appliance.
- **--cckm-user**: Admin user of the CCKM Appliance.
- **--cckm-certificate**: SSL certificate downloaded from the CCKM Appliance.

By default, the content of the SSL certificate spans over multiple new lines. However, to input the certificate as an inline argument, the certificate content needs to be converted into the single-line format where every new line is represented by `\n`.

To convert the certificate into the single-line format:

- a. Download the certificate in a file (say `certificate.cert`).
- b. Store the certificate file at a location on your machine, for example, at `/root/cert/`.
- c. Run the following command:

```

awk 'NF {sub(/\r/, ""); printf "%s\\n",$0;}' <full-path-to-certificate-file>

```

For example, to convert your certificate file (`/root/cert/certificate.cert`), run the command:


```
awk 'NF {sub(/\r/, ""); printf "%s\n",$0;}' /root/cert/certificate.cert
```

The above command returns the certificate in the single-line format, as shown below:

```
-----BEGIN CERTIFICATE-----\nMIIDhjCCAm6gAwIBAgIGCBPpf2sUMA0GCSqGSIB3DQEBDAU
AMC8xIDAeBgNVBAMT\F0NHIENBIFMgb24gcGt2LWnsdXN0ZXIwMQswCQYDVQQGEwJVUzAeFw0yM
TAzMMDMw\nNzI0MDVaFw0zMTAzMDQwNzI0MDVaMCQxFTATBgNVBAMTDHBrdi1jbHVzdGVyMTTEL\nM
AkGA1UEBhMCMVVMwggEiMA0GCSqGSIB3DQEBAQUAA4IBDwAwggEKAoIBAQD0h14c\nhxamUh7e0/T
zdS016djkaL8x1a/uljy8RediBA2W5Yrmm4sTBx0/PFZ0l+0wyzYe\n7VqsCG3P5+lsTiXw2VLXK
T7qyiil7P8ParCoKt6v5fp6BjtDZzq7q3uG8daZqL4M\nKrAiI8rcGPDgsAex/kSPAB4di1AFjDD
Nv7SCiet5kWJq8YeTZEuwYa8rxLi2QEqJ\n4h/mHDqD35jNiSf4FRksWfUwtsY652XIIfBvPdnwkp
sA/+kEN04Aiuve9CSvyIBmR\nKaqBwnmYdQZaBJGFKEvjnArtSMMwCULlxikQCiUk1BaW9H7f/WQ
UuNh7xManRb7w\ndNkhqAhSPyqgjz/TAgMBAAGjgbIwga8wFgYDVR0LAQH/BAwwCgYIKwYBBQUHA
wEw\ndAYDVR0TAQH/BAIwADA0BgNVHQ8BAf8EBAMCA6gwHQYDVR00BBYEFFnFr/+FuXPw\n02+F2
es0v8/XeGvzkuy55dT07SwhLGTkiuv5L2ADawGQmLnQ7thNUN2P\n-----END CERTIFICAT
E-----\n
```

- d. Copy the returned single-line certificate content. You need to paste this content as the value for `--cckm-certificate`.

- `--public-key-id`: ID of the local (public) key of the RSA key pair generated in [Generate RSA Key Pair](#).

The command prompts for the password. Enter the password of the CCKM Appliance user (`cckm-user`).

Example:

```
./ksctl-linux-amd64 migrations download --cckm-host "127.0.0.1" --cckm-user "admin" --cckm-certificate "-----BEGIN CERTIFICATE-----\nMIIDhjCCAm6gAwIBAgIGCBPpf2sUMA0GCSqGSIB3DQEBDAUAMC8xIDAeBgNVBAMT\F0NHIENBIFMgb24gcGt2LWnsdXN0ZXIwMQswCQYDVQQGEwJVUzAeFw0yMTAzMMDMw\nNzI0MDVaFw0zMTAzMDQwNzI0MDVaMCQxFTATBgNVBAMTDHBrdi1jbHVzdGVyMTTEL\nMAkGA1UEBhMCMVVMwggEiMA0GCSqGSIB3DQEBAQUAA4IBDwAwggEKAoIBAQD0h14c\nhxamUh7e0/TzdS016djkaL8x1a/uljy8RediBA2W5Yrmm4sTBx0/PFZ0l+0wyzYe\n7VqsCG3P5+lsTiXw2VLXKT7qyiil7P8ParCoKt6v5fp6BjtDZzq7q3uG8daZqL4M\nKrAiI8rcGPDgsAex/kSPAB4di1AFjDDNv7SCiet5kWJq8YeTZEuwYa8rxLi2QEqJ\n4h/mHDqD35jNiSf4FRksWfUwtsY652XIIfBvPdnwkp sA/+kEN04Aiuve9CSvyIBmR\nKaqBwnmYdQZaBJGFKEvjnArtSMMwCULlxikQCiUk1BaW9H7f/WQUuNh7xManRb7w\ndNkhqAhSPyqgjz/TAgMBAAGjgbIwga8wFgYDVR0LAQH/BAwwCgYIKwYBBQUHAwEw\ndAYDVR0TAQH/BAIwADA0BgNVHQ8BAf8EBAMCA6gwHQYDVR00BBYEFFnFr/+FuXPw\n02+F2es0v8/XeGvzkuy55dT07SwhLGTkiuv5L2ADawGQmLnQ7thNUN2P\n-----END CERTIFICATE-----\n" --public-key-id "bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c2138569d99e395"
```

Output:

```
CCKM enterprise user password:
File is downloading.
File downloaded successfully: migration_20210621143427.json
```

An export file named "migration_<TIMESTAMP>.json" (for example, migration_20210621143427.json) is created and downloaded to your machine.

Note

After downloading the migration data (the export file), open the file and check whether it contains a "containers" tag at the beginning of the file.

Upload Export File

As a CipherTrust Manager administrator, upload the downloaded export file to CCKM Embedded.

Run the command:

```
./ksctl-linux-amd64 migrations upload --file <export-file-name>
```

Here,

- `--file`: Name of the export file.

Example:

```
./ksctl-linux-amd64 migrations upload --file migration_20210621143427.json
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 21712358,
  "created_at": "2021-05-07T04:55:33.251439962Z",
  "checksum_sha256": "3e853d19e1d4a3687dfcf7bde5fc6bb192c19345462c5f4ae05c481acf26968b",
  "product": "CCKM"
}
```

Note down the "id" value, "26304f13-5626-419b-bd81-459fda7fdb68". It is required when applying the migration data. Now, you need to apply the migration data to CCKM Embedded (refer to [Migrate Complete Data](#)).

Apply Migration Data

The migration data can be applied to CCKM Embedded in two ways:

- [Migrate Data of Set of Containers](#)
- [Migrate Complete Data](#)

Note

- In a clustered CipherTrust Manager environment, apply the migration data on one node only. Migrated data is automatically replicated to other nodes of the cluster.
- After migration, all the source keys will be migrated to the current domain on the CipherTrust Manager (where the migration is being performed).
- Even if you migrate specific KMS containers, all of the CipherTrust Manager source keys will be migrated.

Migrate Data of Set of Containers

This method requires a list of containers. After the exported file is uploaded to the CipherTrust Manager, you can get the list of containers and select the containers to be migrated.

To migrate the data of a set of containers:

1. List the migrated containers.

Run the command:

```
./ksctl-linux-amd64 migrations containers list -id <export-file-id>
```

Here,

- `--id`: ID of the uploaded export file. Refer to [Upload Export File](#) for the ID of the export file.

Example:

```
./ksctl-linux-amd64 migrations containers list --id 26304f13-5626-419b-bd81-459fda7fdb68
```

Output:

```
{
  "Containers": [
    {
      "id": "b10ba3f2-b835-485d-b09a-07d4c002ff97",
      "type": "aws",
      "name": "805182230944",
```

```

        "tenant": "805182230944"
    },
    {
        "id": "0de664f3-7c70-4fae-9213-4e16bb99a4fc",
        "type": "aws",
        "name": "556782317223",
        "tenant": "556782317223"
    },
    {
        "id": "3f2b09ba-485d-b835-b09a-07d4c002ff97",
        "type": "salesforce",
        "name": "0056A0000011XDvQAM",
        "tenant": "0056A0000011XDvQAM"
    },
    {
        "id": "4f4b11ba-c385-456d-b09a-07d4c002ff97",
        "type": "salesforcesandbox",
        "name": "0056A0000011ABvZCM",
        "tenant": "0056A0000011ABvZCM"
    },
    {
        "id": "0941754d-9cb9-4b32-a241-59f7a4703756",
        "type": "azure",
        "name": "pkv-premium",
        "tenant": "d27d849e-e487-4b0e-a54c-a71e67687d10"
    }
}

```

This command returns the list of containers with their IDs. From this list, you can select the containers to be migrated.

2. Select the desired containers to be migrated. In this example, we will migrate the containers with the IDs `"b10ba3f2-b835-485d-b09a-07d4c002ff97"` and `"0941754d-9cb9-4b32-a241-59f7a4703756"`.
3. Apply the migration to migrate the desired container.

Run the command:

```

./ksctl-linux-amd64 migrations apply --id <export-file-id> --private-key-id <private-key-id> --migrate-cckm-source-keys --containers "<container1-id>,<container2-id>,...,<containerN-id>"

```

Here,

- `--id`: ID of the uploaded export file. Refer to [Upload Export File](#) for the ID of the export file.
- `--private-key-id`: ID of the private key of the RSA key pair. Refer to [Generate RSA Key Pair](#) for the private key ID.
- `--migrate-cckm-source-keys`: Tag to migrate the source keys.
- `--containers`: Comma-separated container IDs.

Example:

```
./ksctl-linux-amd64 migrations apply --id 26304f13-5626-419b-bd81-459fda7fdb68
--private-key-id b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c3
3c1 --migrate-cckm-source-keys --containers "b10ba3f2-b835-485d-b09a-07d4c002f
f97,0941754d-9cb9-4b32-a241-59f7a4703756",
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 22940401,
  "created_at": "2021-05-06T14:19:54.76812635Z",
  "status": "In progress",
  "checksum_sha256": "3386289c00e7e8bf7ad70a5ca12d0852913c0fb3a54f584c28460a
a9cd65383e",
  "product": "CCKM"
}
```

The container migration data is applied. It might take some time to complete. Verify the migration status. Refer to [Check Migration Status](#) for details.

Migrate Complete Data

To migrate the complete data, run the command:

```
./ksctl-linux-amd64 migrations apply --id <export-file-id> --private-key-id <priv
ate-key-id> --migrate-cckm-source-keys
```

Here,

- `--id`: ID of the uploaded export file. Refer to [Upload Export File](#) for the ID of the export file.
- `--private-key-id`: ID of the private key of the RSA key pair. Refer to [Generate RSA Key Pair](#) for the private key ID.

- `--migrate-cckm-source-keys`: Tag to migrate the source keys.

Example:

```
./ksctl-linux-amd64 migrations apply --id 26304f13-5626-419b-bd81-459fda7fdb68 --
private-key-id b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1
--migrate-cckm-source-keys
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 22940401,
  "created_at": "2021-05-06T14:19:54.76812635Z",
  "status": "In progress",
  "checksum_sha256": "3386289c00e7e8bf7ad70a5ca12d0852913c0fb3a54f584c28460aa9c
d65383e",
  "product": "CCKM"
}
```

Check Migration Status

After you have applied the migration data, verify the migration status by running the `ksctl migrations status` command.

Example:

```
./ksctl-linux-amd64 migrations status
```

Output:

```
{
  "id": "9c9149ad-901b-405f-aefd-b279e6257f97",
  "overall_status": "Completed",
  "source": "CCKM",
  "cckm_azure_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_sfmc_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_aws_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_dsm_keys": {
```

```

    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 }
  }

```

In the sample output above, `"overall_status": "Completed"` indicates that migration of CCKM Appliance to CCKM Embedded is successful. Also, verify that migration status of different clouds and key sources such as `"cckm_azure_keys"`, `"cckm_aws_keys"`, and `"cckm_dsm_keys"` is `"Completed"`.

In the output, `"num_ignored"` indicates the number of Azure keys for which blobs are present on the CipherTrust Manager but the keys have been deleted.

After the migration status is `"Completed"`, you can add connection to supported clouds and key sources to perform operations on the added data.

After migration, you can see the migrated keys on the Keys page of the CipherTrust Manager GUI.

Add Connection

Add connection to the desired clouds on the CipherTrust Manager. Refer to [Connection Manager](#) for details on connection management.

Update Containers

After the connection is created, update connection in AWS KMS, Azure key vaults, Salesforce organizations, and Google key rings on the Containers tab. After the update, keys in supported clouds and key sources can be managed from CCKM Embedded.

For **Salesforce**, you also need to synchronize certificates and update endpoints linked with the organizations.

- **Synchronize certificates:** After migration, the certificates linked with the cache-only keys and schedulers are not synchronized. You need to manually synchronize them on CCKM. Refer to [Prepare the Cache-Only Key Integration](#) for details.
- **Update endpoints:** The endpoint URLs used in Named Credentials for cache-only keys before migration will not work after migration. This is because the Callout URL check performed by Salesforce while fetching the cache-only keys will fail. So, you need to update the endpoint URLs in Named Credential on Salesforce and create the same endpoints on CCKM.

Alternatively, you can create new endpoint URLs on Salesforce and CCKM and use them.

Refer to [Prepare the Cache-Only Key Integration](#) and [Edit a Salesforce Cache-Only Key Endpoint](#) for details.

5. Cloud keys already migrated, but kept DSM as key source, now migrate DSM keys to a local CM

This section provides instructions to migrate the DSM source keys from CCKM Appliance to the CipherTrust Manager. This section assumes that you have already migrated only the cloud keys to the CipherTrust Manager.

Note

The user who performs the migration becomes the owner of the migrated keys.

The steps above apply to the root domain only. To perform migration on a child domain, refer to [Migration from CCKM Appliance to Child Domain](#).

Generate RSA Key Pair

Creating migration data from CCKM Appliance requires an RSA key pair (public and private) on the CipherTrust Manager. The public key is used to encrypt the data while the private key is used to decrypt the migrated data.

To generate an RSA key pair, run the `ksctl keys create` command:

```
./ksctl-linux-amd64 keys create --name <rsa-key-name> --alg <key-algorithm> --size <key-size>
```

Here,

- `--name`: Name for RSA key pair.
- `--alg`: Algorithm for the RSA key pair.
- `--size`: Size for the key pair.

Example:

```
./ksctl-linux-amd64 keys create --name rsa-key --alg RSA --size 4096
```

Output:

```
{
  "id": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1",
  "uri": "kylo:kylo:vault:keys:rsa-key-v0",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-05-07T03:55:47.519466Z",
  "name": "rsa-key",
  "updatedAt": "2021-05-07T03:55:47.519466Z",
  "usage": "sign",
  "usageMask": 3,
  "meta": {
    "ownerId": "local|5e3b45c6-6f26-4413-9752-e6fd15418a61"
  },
  "version": 0,
  "algorithm": "RSA",
  "size": 4096,
```



```

    "unexportable": false,
    "undeletable": false,
    "neverExported": true,
    "neverExportable": false,
    "emptyMaterial": false,
    "publickey": "-----BEGIN PUBLIC KEY-----\nMIICijANBgkqhkiG9w0BAQEFAAOCAg8AMII
CCgKCAGeAwfd4fDEgdJydUP0kPpAL\nriQW+IpWm9oSte//pv45UXpw0wkag1FbSmEUQQMf02KdRW5s
o+4jrkX44gQhmDTA\nnamrpijweJa0HzkaqnTkMtCutFP9nbx33JiWRSYSKqXsEJho+P9SqXz45uxf7iis
5\n4NF0SpZSIYi3COH4xtJ7hK+6BXbLXBZHVpfQ6LN8p/+WDwcIIkSyWGQxj4V0xTwV\nnfNBNoQrXvanr
EX+nide28vuX1bJ1UzghUwcFT12VZL8KIrkviCKMwkBNiuiiXgh\nbntYGBY84ZbPjREgaodbaU45vj3
8/dpusL75Q2hkUdv5mYvTqN+OPVbJrTTQFzGfw\nmM3Pt86iBFfu3XH/ZMH4dbV3HHXJP7+mHI3c0hUlvo
jwx9hnKygn3fY4Darx/N0yr\nniCp6Sz7FI3sExAAIEF+AJ7zqyXK6a/NGve5gAqt1w3fn0YIFeD8f6oX0
YBFFniu3\n3uX//4WcNdgyTXKXhDsZAtaLqmHv9jIwGZ0pTlj8xefZPbkoDN0N3uC92b0tzI7F\nn7+Iq0
iEf5bg4huU/EJh8emYgU8mPZGpwPtPVUFiKm0Y7EbvHS1C6RIqRE1hnCZAa\nnZSMup6LLzZGvk6SM0339
c5gDJuS+kGkYK/f0wuWJ7q05m+T/27J1IoNna6JuZ9el\nnZDMxs7Rqj4cdezaa3CTV4l8CAwEAA
Q==\n-----END PUBLIC KEY-----\n",
    "defaultIV": "78f83dddc0ee01a2ab3ff579c908a33a",
    "sha1Fingerprint": "878bcd84e81c4170",
    "sha256Fingerprint": "c9c2d321b21d34a3e82460df8839e55f3ebca977766658d830d5100
fb29bed75",
    "objectType": "Private Key",
    "activationDate": "2021-05-07T03:55:43.229267Z",
    "state": "Active",
    "aliases": [
      {
        "alias": "rsa-key",
        "type": "string",
        "index": 0
      }
    ],
    "links": [
      {
        "id": "6dc578f7-1864-43a7-899a-5035d54f1772",
        "uri": "kylo:kylo:vault:links:6dc578f7-1864-43a7-899a-5035d54f1772",
        "account": "kylo:kylo:admin:accounts:kylo",
        "application": "ncryptify:gemalto:admin:apps:kylo",
        "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
        "createdAt": "2021-05-07T03:55:47.531762Z",
        "updatedAt": "2021-05-07T03:55:47.531762Z",
        "type": "publicKey",
        "source": "kylo:kylo:vault:keys:rsa-key-v0",
        "sourceID": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202
bd4c33c1",
        "target": "kylo:kylo:vault:keys:rsa-key-pub-v0",
        "targetID": "bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c213856
9d99e395",
        "index": 0
      }
    ],
    "uuid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3",
    "muid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3e129ed33-317d-4584-9cbf-d0e882f5
8fca"
  }

```

In the sample output above, "sourceID":

"b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1" is the private key ID. The "targetID":

"bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c2138569d99e395" under "links" is

the public key ID.

Create Migration Data

Create the migration data for the DSM key source. Specify `--key-source` as `dsm`.

Run the command:

```
./ksctl-linux-amd64 migrations create --key-source dsm --public-key-id <public-key-id>
```

Here,

- `--key-source`: Specify `dsm` as the key source.

Example:

```
./ksctl-linux-amd64 migrations create --key-source dsm --public-key-id c9b6922153e74c1f9be4bf9344ebf8eed827aa281be947a2b249b57f9f0c5d1c
```

Output:

```
{
  "status": "In progress"
}
```

Get the uploadID

After you have initiated the creation of migration data for the DSM key source, get the `uploadID` by running the `ksctl migrations status` command.

Example:

```
./ksctl-linux-amd64 migrations status
```

Output:

```
{
  "id": "9c9149ad-901b-405f-aefd-b279e6257f97",
  "overall_status": "Completed",
  "source": "CCKM",
  "cckm_azure_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_sfdc_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
}
```

```

    "cckm_aws_keys": {
      "status": "Completed",
      "num_processed": 100,
      "num_failed": 0,
      "num_ignored": 0 },
    "cckm_dsm_keys": {
      "status": "Completed",
      "num_processed": 100,
      "num_failed": 0,
      "num_ignored": 0 },
    "cckm_gcp_keys": {
      "status": "",
      "num_processed": 0,
      "num_failed": 0,
      "num_ignored": 0},
    "cckm_generate_migration": {
      "status": "Completed",
      "uploadID": "f915a761-9fa8-449d-a969-122601ef244e"
    }
  }
}

```

Note down the "uploadID" value, "f915a761-9fa8-449d-a969-122601ef244e". It is required when applying the migration data. Now, you need to apply the migration data to CCKM Embedded (refer to [Apply Migration Data](#)).

Apply Migration Data

Note

- In a clustered CipherTrust Manager environment, apply the migration data on one node only. Migrated data is automatically replicated to other nodes of the cluster.
- After migration, all the source keys from DSM will be migrated to the current domain on the CipherTrust Manager (where the migration is being performed).
- Even if you migrate specific KMS containers, all of the DSM source keys will be migrated.
- After all the source keys are moved to the CipherTrust Manager, you can remove DSM from CCKM Embedded, if needed. Refer to [Deleting DSM Domains](#) and [Deleting a DSM Connection](#).

To apply the migration data, run the command:

```
./ksctl-linux-amd64 migrations apply --id <uploadID> --private-key-id <private-key-id>
```

Here,

- `--id`: uploadID returned in [Check Status of uploadID](#).

- `--private-key-id`: ID of the private key of the RSA key pair. Refer to [Generate RSA Key Pair](#) for the private key ID.

Example:

```
./ksctl-linux-amd64 migrations apply --id f915a761-9fa8-449d-a969-122601ef244e --private-key-id b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1
```

Output:

```
{
  "id": "f915a761-9fa8-449d-a969-122601ef244e",
  "file_size": 70697,
  "created_at": "2022-11-28T04:24:28.004216933Z",
  "status": "In progress",
  "checksum_sha256": "8b5839e47dfbb68b9dadbd1f31e416321a5033db2b7956ddc03e07748e58258a8",
  "product": "CCKM"
}
```

Check Migration Status

After you have applied the migration data, verify the migration status by running the `ksctl migrations status` command.

Example:

```
./ksctl-linux-amd64 migrations status
```

Output:

```
{
  "id": "f915a761-9fa8-449d-a969-122601ef244e",
  "overall_status": "Completed",
  "source": "CCKM",
  "cckm_azure_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_sfdc_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_aws_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_dsm_keys": {
```

```

    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
    "cckm_generate_migration": {
    "status": "Completed",
    "uploadID": "f915a761-9fa8-449d-a969-122601ef244e"}
  }

```

In the sample output above, `"overall_status": "Completed"` indicates that migration of DSM source keys from CCKM Appliance to CCKM Embedded is successful.

6. Cloud keys already migrated, but kept DSM as the key source, now migrate DSM keys to an external CipherTrust Manager

Before migration, ensure that both the local and external CipherTrust Managers are running the identical build of the same CipherTrust Manager version (2.20 or later).

Migrate the DSM source keys from CipherTrust Manager to external CipherTrust Manager

Tip

Perform this step multiple times - once for each external CipherTrust Manager domain. This ensures that all DSM source keys are properly migrated from all DSM domains to their respective CipherTrust Manager domains.

This section provides instructions to migrate the DSM source keys from CipherTrust Manager to external CipherTrust Manager. This section assumes that you have already migrated the cloud keys to the CipherTrust Manager.

Note

The user who performs the migration becomes the owner of the migrated keys.

The steps below apply to the root domain only. To perform migration on a child domain, refer to [Migration from CCKM Appliance to Child Domain](#).

1. Generate RSA Key Pair

Creating migration data from local CipherTrust Manager requires an RSA key pair (public and private) on the external CipherTrust Manager. The public key is used to encrypt the data while the private key is used to decrypt the migrated data.

To generate an RSA key pair, run the `ksctl keys create` command on external CipherTrust Manager:

```
./ksctl-linux-amd64 keys create --name <rsa-key-name> --alg <key-algorithm> --size <key-size>
```

Here,

- **--name**: Name for RSA key pair.
- **--alg**: Algorithm for the RSA key pair.
- **--size**: Size for the key pair.

Example:

```
./ksctl-linux-amd64 keys create --name rsa-key --alg RSA --size 4096
```

Output:

```
{
  "id": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1",
  "uri": "kylo:kylo:vault:keys:rsa-key-v0",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-05-07T03:55:47.519466Z",
  "name": "rsa-key",
  "updatedAt": "2021-05-07T03:55:47.519466Z",
  "usage": "sign",
  "usageMask": 3,
  "meta": {
    "ownerId": "local|5e3b45c6-6f26-4413-9752-e6fd15418a61"
  },
  "version": 0,
  "algorithm": "RSA",
  "size": 4096,
  "unexportable": false,
  "undeletable": false,
  "neverExported": true,
  "neverExportable": false,
  "emptyMaterial": false,
  "publickey": "-----BEGIN PUBLIC KEY-----\nMIICIjANBgkqhkiG9w0BAQEFAAOCAg8A\nMIICGgKCAgEAWfd4fDEgdJydUP0kPpAL\nnriQW+IpmW9oSte//pv45UXpw0wkag1FbSmEUQQMf02Kd\nRW5so+4jrkX44gQhmdTA\nnamrpijweJa0HzkaqnTkMtCUTFP9nbx33JiWRSYSKqXsEJho+P9SqXz45"
```

```

uxf7iis5\n4NF0SpZSIYi3C0H4xtJ7hK+6BXbLXBZHVpfQ6LN8p/+wDwcIIkSyWGQxj4V0xTwV\nfN
BNoQrXvanrEX+nide28vuX1bJ1UzglhUwcFT12VZL8KIrkviCKMwkBNIuuiXgh\nbtYGBY84ZbPjRE
gaodbaU45vj38/dpusL75Q2hkUdv5mYvTqN+OPVbJrTTQFzGfw\nM3Pt86iBFfu3XH/ZMH4dbV3HHX
JP7+mHI3c0hUlvojw9hnKygn3fY4Darx/N0yr\niCp6Sz7FI3sExAAIeF+AJ7zqyXK6a/NGve5gAq
t1w3fn0YIFeD8f6oX0YBFFniu3\n3uX//4WcNdgyTXKXhDsZAtaLqmHv9jIwGZ0pTlj8xefZPbkoDN
ON3uC92b0tzI7F\n7+Iq0iEf5bg4huU/EJh8emYgU8mPZGpwPtPVUfiKm0Y7EbvHS1C6RIqRE1hnCZ
Aa\nZSMup6LLzZGvk6SM0339c5gDJuS+kGkYK/f0wuWJ7q05m+T/27J1IoNna6JuZ9eL\nZDMxs7Rq
j4cdezaa3CTV4l8CAwEAAQ==\n-----END PUBLIC KEY-----\n",
  "defaultIV": "78f83dddc0ee01a2ab3ff579c908a33a",
  "sha1Fingerprint": "878bcd84e81c4170",
  "sha256Fingerprint": "c9c2d321b21d34a3e82460df8839e55f3ebca977766658d830d5
100fb29bed75",
  "objectType": "Private Key",
  "activationDate": "2021-05-07T03:55:43.229267Z",
  "state": "Active",
  "aliases": [
    {
      "alias": "rsa-key",
      "type": "string",
      "index": 0
    }
  ],
  "links": [
    {
      "id": "6dc578f7-1864-43a7-899a-5035d54f1772",
      "uri": "kylo:kylo:vault:links:6dc578f7-1864-43a7-899a-5035d54f177
2",
      "account": "kylo:kylo:admin:accounts:kylo",
      "application": "ncryptify:gemalto:admin:apps:kylo",
      "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
      "createdAt": "2021-05-07T03:55:47.531762Z",
      "updatedAt": "2021-05-07T03:55:47.531762Z",
      "type": "publicKey",
      "source": "kylo:kylo:vault:keys:rsa-key-v0",
      "sourceID": "b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd
202bd4c33c1",
      "target": "kylo:kylo:vault:keys:rsa-key-pub-v0",
      "targetID": "bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c213
8569d99e395",
      "index": 0
    }
  ],
  "uuid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3",
  "muid": "4b9e7c53-40d7-44b7-9fa1-31cf8d0237d3e129ed33-317d-4584-9cbf-d0e88

```

```
2f58fca"
}
```

In the sample output above, "sourceID":

"b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1" is the private key ID. The "targetID":

"bd3e3bfa246f470ea6327646b3db359fcb882a6e2a6d4f839c2138569d99e395" under "links" is the public key ID. The "publickey" is the public key material.

2. Download Migration Data

Download the migration data to migrate DSM keys to an external CipherTrust Manager.

Run the command on local CipherTrust Manager:

```
./ksctl-linux-amd64 migrations download --public-key-material "<public-key-material>" --download-dsm-externalcm-mig-data
```

Here,

- **--public-key-material**: public-key-material is the public key generated from the external CipherTrust Manager to encrypt the migration data.
- **--download-dsm-externalcm-mig-data**: With this flag, the migration data will be downloaded to the migration of source keys from DSM to external CipherTrust Manager when the cloud keys from CCKM Enterprise are already migrated to CipherTrust Manager

Example:

```
./ksctl-linux-amd64 migrations download --public-key-material "-----BEGIN PUBLIC KEY-----MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAgEAAoIBAQDXrrAQ01aKDRY9-----END PUBLIC KEY-----" --download-dsm-externalcm-mig-data
```

Output:

```
{
  File is downloading...
  File downloaded successfully: migration_20250508143834.json
}
```

3. Upload Migration Data to the External CipherTrust Manager

As a CipherTrust Manager administrator, upload the downloaded export file to the external CipherTrust Manager.

Run the command on external CipherTrust Manager:

```
./ksctl-linux-amd64 migrations upload --file <export-file-name>
```

Here,

- **--file**: Name of the export file.

Example:

```
./ksctl-linux-amd64 migrations upload --file migration_20210621143427.json
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 21712358,
  "created_at": "2021-05-07T04:55:33.251439962Z",
  "checksum_sha256": "3e853d19e1d4a3687dfcf7bde5fc6bb192c19345462c5f4ae05c481
acf26968b",
  "product": "CCKM"
}
```

Note down the **"id"** value, **"26304f13-5626-419b-bd81-459fda7fdb68"**. It is required when applying the migration data. Now, you need to apply the migration data to external CipherTrust Manager (refer to [Migrate Complete Data](#)).

4. Apply Migration Data

The migration data can be applied to external CipherTrust Manager by [migrating data of set of containers](#).

Note

- In a clustered CipherTrust Manager environment, apply the migration data on one node only. Migrated data is automatically replicated to other nodes of the cluster.
- After migration, all the source keys from DSM will be migrated to the current domain on the external CipherTrust Manager (where the migration is being performed).
- After all the source keys are moved to the external CipherTrust Manager, you can remove DSM from

CCKM Embedded, if needed. Refer to [Deleting DSM Domains](#) and [Deleting a DSM Connection](#).

To apply the migration data, run the command on external CipherTrust Manager:

```
./ksctl-linux-amd64 migrations apply --id <uploadID> --private-key-id <private-key-id> --migrate-dsmkeys-to-externalcm-from-cm
```

Here,

- **--id**: uploadID returned in [Check Status of uploadID](#).
- **--private-key-id**: ID of the private key of the RSA key pair.

The public key material of the RSA key pair is used to download the migration data.

- **--migrate-dsmkeys-to-externalcm-from-cm**: This optional parameter is used only while migrating source keys from DSM to external CipherTrust Manager when the cloud keys from CCKM Enterprise are already migrated to CipherTrust Manager. With this flag, the source keys will be migrated from DSM to external CipherTrust Manager.

Example:

```
./ksctl-linux-amd64 migrations apply --id f915a761-9fa8-449d-a969-122601ef244e --private-key-id b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c33c1 --migrate-dsmkeys-to-externalcm-from-cm
```

Output:

```
{
  "id": "f915a761-9fa8-449d-a969-122601ef244e",
  "file_size": 70697,
  "created_at": "2022-11-28T04:24:28.004216933Z",
  "status": "In progress",
  "checksum_sha256": "8b5839e47dfbb68b9dadb1f31e416321a5033db2b7956ddc03e07748e58258a8",
  "product": "CCKM"
}
```

5. Check Migration Status

After you have applied the migration data, verify the migration status by running the `ksctl migrations status` command.

Example:

```
./ksctl-linux-amd64 migrations status
```

Output:

```
{
  "id": "f915a761-9fa8-449d-a969-122601ef244e",
  "overall_status": "Completed",
  "source": "CCKM",
  "cckm_azure_keys": {
    "status": "not started",
    "num_processed": 0,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_sfmc_keys": {
    "status": "not started",
    "num_processed": 0,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_aws_keys": {
    "status": "not started",
    "num_processed": 0,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_dsm_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
  "cckm_generate_migration": {
    "status": "not started"}
}
```

In the sample output above, `"overall_status": "Completed"` indicates that the migration of DSM source keys from local CipherTrust Manager to external CipherTrust Manager is successful.

Migrate to Update the Source Key Linkings in Cloud Keys

1. Prerequisites for Migration

- Source keys from DSM are migrated to the external CipherTrust Manager.
- After migrating the DSM source keys to the external CipherTrust Manager, assign the **Key Admins** permission to the client CipherTrust Manager. Refer to [Refer to Add the Client](#) for details.

2. Migration Steps

On the CipherTrust Manager, configure the *external* CipherTrust Manager as a key source.

- Add the external CipherTrust Manager connection. Refer to [CM Connection/External CM](#).
- Add the *external* CipherTrust Manager domains where the DSM source keys were migrated.
- Refresh the CipherTrust domains added to the CipherTrust Manager.

3. Download Migration Data

Download the migration data from CipherTrust Manager to update the source key linkings.

Run the command on local CipherTrust Manager:

```
./ksctl-linux-amd64 migrations download --public-key-material "<public-key-material>" --download-dsm-externalcm-mig-data
```

Here,

- **--public-key-material**: Public key generated from local CipherTrust Manager to encrypt migration data.
- **--download-dsm-externalcm-mig-data**: With this flag, the migration data will be downloaded to the migration of source keys from DSM to external CipherTrust Manager when the cloud keys from CCKM Enterprise are already migrated to CipherTrust Manager.

Example:

```
./ksctl-linux-amd64 migrations download --public-key-material "-----BEGIN PUBLIC KEY-----MIIEvGIBADANBgkqhkiG9w0BAQEFAASCBAgEAAoIBAQDXrrAQ01aKDRY9-----END PUBLIC KEY-----" --download-dsm-externalcm-mig-data
```

Output:

```
{
  File is downloading...
  File downloaded successfully: migration_20250508160025.json
}
```

4. Upload Migration Data to the Local CipherTrust Manager

As a CipherTrust Manager administrator, upload the downloaded export file to the local CipherTrust Manager.

Run the command on local CipherTrust Manager:

```
./ksctl-linux-amd64 migrations upload --file <export-file-name>
```

Here,

- **--file**: Name of the export file.

Example:

```
./ksctl-linux-amd64 migrations upload --file migration_20210621143427.json
```

Output:

```
{
  "id": "26304f13-5626-419b-bd81-459fda7fdb68",
  "file_size": 21712358,
  "created_at": "2021-05-07T04:55:33.251439962Z",
  "checksum_sha256": "3e853d19e1d4a3687dfcf7bde5fc6bb192c19345462c5f4ae05c481acf26968b",
  "product": "CCKM"
}
```

Note down the **"id"** value, **"26304f13-5626-419b-bd81-459fda7fdb68"**. It is required when applying the migration data. Now, you need to apply the migration data to the local CipherTrust Manager (refer to [Migrate Complete Data](#)).

5. Apply Migration Data

To apply the migration data, run the command on local CipherTrust Manager:

```
./ksctl-linux-amd64 migrations apply --id <uploadID> --private-key-id <private-key-id> --update-cloudkeys-with-externalcm-keys
```

Here,

- **--id**: uploadID returned in [Check Status of uploadID](#).
- **--private-key-id**: ID of the private key of the RSA key pair.

The public key material of the RSA key pair is used to download the migration data.

- `--update-cloudkeys-with-externalcm-keys`: This optional parameter is used only while migrating source keys from DSM to external CipherTrust Manager when the cloud keys from CCKM Enterprise are already migrated to CipherTrust Manager. If this flag is set to true, the cloud keys will be updated with the corresponding migrated external CipherTrust Manager keys.

Example:

```
./ksctl-linux-amd64 migrations apply --id f915a761-9fa8-449d-a969-122601ef244e
--private-key-id b4336425a98541b68a105326be8abd777ac994f789ac46c2a79dd202bd4c3
3c1 --update-cloudkeys-with-externalcm-keys
```

Output:

```
{
  "id": "f915a761-9fa8-449d-a969-122601ef244e",
  "file_size": 70697,
  "created_at": "2022-11-28T04:24:28.004216933Z",
  "status": "In progress",
  "checksum_sha256": "8b5839e47dfbb68b9dadb1f31e416321a5033db2b7956ddc03e077
48e58258a8",
  "product": "CCKM"
}
```

6. Check Migration Status

After you have applied the migration data, verify the migration status by running the `ksctl migrations status` command.

Example:

```
./ksctl-linux-amd64 migrations status
```

Output:

```
{
  "id": "f915a761-9fa8-449d-a969-122601ef244e",
  "overall_status": "Completed",
  "source": "CCKM",
  "cckm_azure_keys": {
    "status": "not started",
```

```

    "num_processed": 0,
    "num_failed": 0,
    "num_ignored": 0 },
    "cckm_sfkc_keys": {
    "status": "not started",
    "num_processed": 0,
    "num_failed": 0,
    "num_ignored": 0 },
    "cckm_aws_keys": {
    "status": "not started",
    "num_processed": 0,
    "num_failed": 0,
    "num_ignored": 0 },
    "cckm_dsm_keys": {
    "status": "Completed",
    "num_processed": 100,
    "num_failed": 0,
    "num_ignored": 0 },
    "cckm_generate_migration": {
    "status": "Completed"}
}

```

In the sample output above, `"overall_status": "Completed"` indicates that migration of DSM source keys linking to external CipherTrust Manager is successful.

Note

If the migration data contains Salesforce schedules, then after migration, update the external CipherTrust Manager domain and Certificate in the Salesforce schedules.

Migration from CCKM Appliance to Child Domain

To migrate the CCKM Appliance data to CCKM Embedded in a child domain.

1. Set `export KSCTL_DOMAIN=<child-domain>` on the machine where you will run the ksctl commands.
2. Perform the migration steps, as described in sections above.

Note

The steps above show examples of `ksctl` commands. Switch to the child domain `<child-domain>` to:

- Check the migration status using the API.
- Verify or access the migrated data on the CipherTrust Manager GUI.

Backup and Restore

CCKM resources can be backed up and restored on the CipherTrust Manager. CipherTrust Manager supports system backups and domain scoped backups. Refer to [Backups](#) for details.

Only the CCKM EKM resources are part of the domain backup. Support for other CCKM resources will be added in a future release. This section describes the behavior of CCKM resources while restoring domain backups.

Note

- The backup and restore of CCKM resources (Projects, Google EKM Endpoints and Cryptospaces) in a domain only works among the domains of different CipherTrust Managers. This feature does not support backup and restore among different domains of the same CipherTrust Manager.
- Currently, partial domain backup and restore of CCKM resources (Projects, Cryptospaces, EKM Endpoint) are *not* supported.

Restore Behavior of CCKM Resources

Note

It is recommended to restore the backup of CCKM resources in an empty domain.

After restore, make sure to update the **Key URI Hostname** of the restored EKM endpoints.

The following table depicts the behavior of CCKM resources. We recommend you to get familiar with the below terms used in the table:

Resource is skipped: the resource will be ignored and not restored.

Conflicting resource: at least any one of the unique attributes (for example, name) of the resource is the same in the source domain (backup file) and the target domain.

Conflicting Resource	Behavior
Project	1. This project will be skipped. 2. All cryptospaces created under this project will be skipped.

Conflicting Resource	Behavior
	3. All EKM endpoints created under each cryptospace in step 2 will be skipped.
Cryptospace	1. This cryptospace will be skipped. 2. All EKM endpoints created under this cryptospace will be skipped.
EKM Endpoint	1. This EKM Endpoint will be skipped.

Product Brief

CipherTrust Cloud Key Management

Cloud Encryption Key
Lifecycle Management

cpl.thalesgroup.com

THALES
Building a future we can all trust

Industry best practices as defined by the Cloud Security Alliance (CSA) require that keys be stored and managed outside of the cloud service provider and the associated encryption operations¹. Cloud Service Providers (CSPs) can comply with best practices by offering Bring Your Own Key (BYOK) or Hold Your Own Key (HYOK) services to enable customer control of the keys used to encrypt their data. Customer control of the keys allows for the separation, creation, ownership and control, including revocation, of encryption keys or tenant secrets used to create the keys.

CipherTrust Cloud Key Management (CCKM) increases efficiency by reducing the operational burden—even when all of the cloud keys are native keys. Giving customers lifecycle control, centralized management within and among clouds, and visibility of cloud encryption keys reduces key management complexity and operational costs. Customers report that they stepped away from managing keys across a heterogeneous environment and invested in CCKM to enable them to move securely to the cloud—and their cloud use is growing exponentially, reducing management overhead and the potential for security holes.

Control Cloud Encryption Keys

- Leverage the value of BYOK and HYOK services with full- lifecycle cloud encryption key management
- Gain higher efficiency with centralized key management across hybrid, single- and multi-cloud environments, including key discovery, management of native cloud keys and automated key rotation
- Comply with the most stringent data protection mandates with secure key origination
- Amplify the benefits of native keys by using a robust multi-cloud platform with outstanding UI

Best Practices

Industry and internal data protection mandates require increasing protection for sensitive cloud data. Meanwhile, CSA and industry analysts state that cloud encryption keys should be managed by customers. Key management systems (KMSs) can grow to hundreds of master keys per KMS—requiring knowledge of how, when, and by whom encryption keys are used, in addition to backing up, monitoring, rotating, expiring, archiving, suspending, restoring, and destroying keys as needed. CCKM provides comprehensive key lifecycle management to provide visibility on a single pane of glass, and automate and fulfill requirements for safe, comprehensive key management across multiple clouds.

Improved Efficiency

CCKM offers multiple capabilities in support of increasing efficiency:

- Secure centralized key management for Native, BYOK and HYOK cloud keys from a single browser window, across multiple clouds, regions, accounts, subscriptions, projects, applications, org ids, etc.

- Automated synchronization ensuring that cloud console-specific key operations are visible in centralized key management—even if you have already created thousands of native cloud keys at your cloud provider
- Automated key rotation with support for expiring keys ensuring compliance while saving valuable time
- Metadata is collected and laid out in the same order for every cloud provider, removing the need to look for data in disparate places

Single Pane of Glass

Access to each cloud provider from a single console, across multiple accounts, regions, subscriptions and projects makes it easier for organizations to understand how their workloads across different clouds are protected. We continually increase key visibility to make it easier for administrators to manage and control access to the keys in minutes instead of days, and stop threats faster.

Strong Encryption Key Security

Customer key control requires secure key generation and storage. CCKM leverages the security of the CipherTrust Manager, [Luna Network HSM](#), or the Vormetric [Data Security Manager \(DSM\)](#) to create keys with up to FIPS 140-2 Level 3 security.

Hold Your Own Key

CCKM services respond to encryption key requests from cloud providers, support many of the emerging HYOK offerings: AWS External Key Store (XKS); Google Cloud External Key Management (EKM), Google EKM Ubiquitous Data Encryption (UKE), Microsoft 365, and Google Workspace Client-side Encryption for Drive, Gmail, Google Calendar and Google Meet; and Salesforce Cached Keys.

The Compliance Tools You Need

Visibility reports demonstrate compliance to the regulator for mission-critical workloads. Logs may also be directed to a syslog server or Security Information and Event Management (SIEM) tool.

Automation Tools Support Your Initiatives

CCKM capabilities are available programmatically using RESTful APIs, enabling the power of centralized cloud encryption management to work with your automation and self-service initiatives. Interactively explore the APIs with our built-in API playground.

Flexible Deployment Options

Deployment environments include: public cloud, private cloud, hybrid cloud, physical appliances and an as-a cloud-based subscription service.

Regardless of how and where CCKM is deployed, CCKM can manage keys and access to the keys, in any supported cloud.

To meet different organization’s needs, CCKM is available in virtual and physical form factors: Virtual CCKM is an all-software offering easily deployed and can be run in the cloud or on premises and may be found in several cloud provider marketplaces for fast instantiation. Physical appliances are available for customers who prefer an on-prem solution. [CCKM service](#) is available on-demand through the [Thales Data Protection on Demand \(DPoD\) marketplace](#) and in CSP marketplaces.

Multi-cloud Data Security Solutions

CCKM simplifies the need to control and manage encryption keys for cloud services, a critical solution for fulfilling industry and organizational data protection mandates. CCKM and many additional Thales multi-cloud security products, including various [Bring Your Own Advanced Encryption](#) offerings, all with centralized key management, offer you choices on how best to protect all your cloud data. [It’s your data in their cloud.](#)

About Thales

Today’s businesses and governments depend on the cloud, data and software to deliver trusted digital services. That is why the most recognized brands and organizations around the world rely on Thales to help them protect sensitive information and software wherever it is created, stored or accessed – from the cloud and data centers to devices and across networks. As the global leader in data security and software licensing, our solutions enable organizations to move to the cloud securely, achieve compliance with confidence, create more value from their software and deliver seamless digital experiences for millions of consumers every day.

Supported clouds and key management ownership models:

Increasing customer control

Amazon Web Services (AWS) KMS	Native	BYOK	
AWS CloudHSM	Native		
AWS XKS			HYOK
AWS China	Native	BYOK	
AWS GovCloud	Native	BYOK	HYOK
Google Cloud Platform CMEK	Native	BYOK	
Google Cloud Platform EKM			HYOK
Google Cloud Platform EKM UDE			HYOK-CC*
Google Workspace CSE			HYOK
IBM Cloud HPCS	Native	BYOK	
IBM Cloud Key Protect	Native	BYOK	
Microsoft Azure Cloud	Native	BYOK	
Microsoft Azure China	Native	BYOK	
Microsoft Azure GovCloud	Native	BYOK	
Microsoft Azure Managed HSMs	Native	BYOK	
Microsoft 365		BYOK	HYOK
Oracle Cloud Infrastructure	Native	BYOK	HYOK
Salesforce.com	Native	BYOK	HYOK**
Salesforce GovCloud Plus	Native	BYOK	HYOK**
Salesforce Sandbox	Native	BYOK	HYOK**
SAP Data Custodian	Native	BYOK	

*HYOK-CC is HYOK for Confidential Computing

** Cache-only Key Service



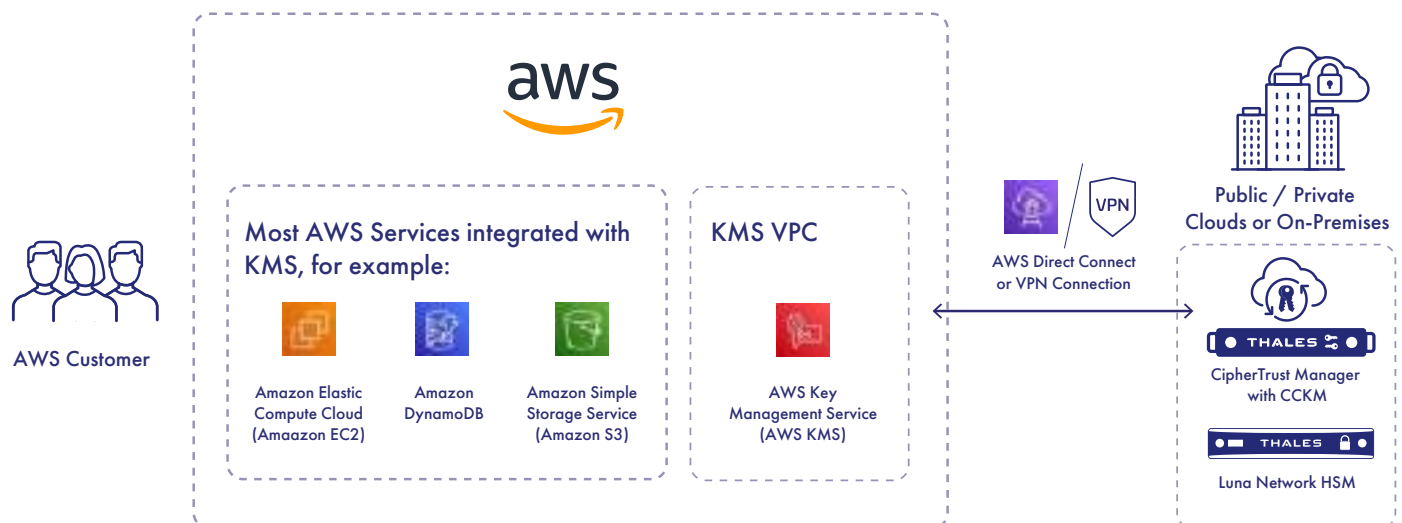
CipherTrust Cloud Key Management Solutions for Amazon Web Services

AWS Key Management Service (AWS KMS) External Key Store (XKS), the new Hold Your Own Key (HYOK) service from Amazon Web Services (AWS), allows customers to protect their data residing in AWS using encryption keys stored outside AWS.

Thales is collaborating with AWS to accelerate the ability of organizations to migrate sensitive data and workloads to the cloud. AWS customers can block and unblock access to encrypted data with keys stored and managed outside of AWS.

CipherTrust Cloud Key Manager

Thales and AWS innovated to create a versatile, feature-rich implementation, providing customers with choices in managing their keys. The collaboration extends the existing key management ownership model of Bring Your Own Key (BYOK) with a Hold Your Own Key (HYOK) offer. With external key store and Thales CipherTrust Cloud Key Manager (CCKM), customers can now choose to have data protected with keys physically located outside of AWS Cloud. The externally stored keys are only accessible via explicit customer authorization. External key store supports most AWS services already integrated with AWS KMS.



Benefits

- Enhance compliance with GDPR in the post Schrems II ruling world by aligning with guidelines from European Data Protection Board (EDPB)
- Improve data sovereignty by providing explicit control over encryption keys outside of the AWS cloud
- Maximize choice from a single console, manage Native, BYOK, HYOK keys across clouds
- Reduce threat surface by centralizing control of keys outside of cloud providers
- Reduce costs by simplifying key management
- Faster time to value by speeding up migration to the cloud

Gain Strong Key Control and Security

External key store enables customers to separate key management from AWS controlled encryption, offering a crucial layer of separation of duty and control. Thales CCKM delivers key generation, reporting, and key lifecycle management that help fulfill internal and industry data protection mandates, with optional FIPS 140-2-certified hardware.

Meet Organizational Needs with Flexible Deployment Options

Thales CCKM is available in virtual and physical form factors: Virtual CCKM is an all-software offering easily deployed and can be run in the cloud or on premises and may be found in several cloud provider marketplaces for fast instantiation. Deployment environments include: public cloud, private cloud, hybrid cloud, and physical appliances. Physical appliances are available for customers who prefer an on-prem solution. Regardless of how and where CCKM is deployed, CCKM can manage keys, and access to the keys, in any reachable, supported cloud.

Increased Efficiency

Thales CCKM centralizes encryption key management from multiple environments, presenting all supported clouds, and even multiple cloud accounts, in a single pane of glass. Advanced cloud key management services and capabilities include automated key rotation, key expiration handling, and cloud key vault synchronization — dramatically reducing the time required for cloud key life cycle management.

In addition to HYOK and BYOK, CCKM supports full key lifecycle management of native cloud keys, including keys stored in AWS CloudHSM.

Align with Best Practices

Deliver compliance, best security practices and control of your data by separating encryption key control from data encryption and decryption operations. Gain operational insights on encryption key usage with reports and logs provided by Thales CCKM.

Integrates With Your Automation Initiatives

CCKM capabilities are available programmatically using RESTful APIs, enabling DevOps and IT teams the power of centralized cloud encryption management to work with the organization's automation and self-service initiatives.

How it Works

External key store is an option within AWS KMS. Once an externally managed key is linked to a KMS key ID using XKS, the externally managed key can be used to protect data in any of the AWS services that integrate with AWS KMS.

External key stores let you control data sovereignty. Data Encryption Keys (DEK) encrypted under KMS keys in the External key store can be decrypted only in the Thales CCKM under your control. When you revoke access to CCKM, by blocking the key or disconnecting the external key store, workloads running in AWS lose all access to your encryption keys, and data encrypted under your keys cannot be decrypted; it is crypto-shredded.

Thales is Here to Help

Contact Thales or [try CCKM for free today](#) to help you assess and define the data protection strategy that best suits your organizational requirements, and for [integration guides](#) to help accelerate your deployment.

About Amazon Web Services

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud offering, with more than 200 fully featured services available from data centers globally. Millions of customers — including the fastest-growing startups, largest enterprises, and leading government agencies — are using AWS to lower costs, increase security, become more agile, and innovate faster.

About Thales

As the global leader in data security, Thales helps the most trusted brands and organizations around the world protect their most sensitive data and software, secure the cloud, provide seamless digital experiences, and achieve compliance through our industry-leading data encryption, identity and access management, and software licensing solutions.

CipherTrust Cloud Key Manager

Cloud Encryption Key Lifecycle Management



Industry best practices as defined by the Cloud Security Alliance (CSA), require that keys be stored and managed outside of the cloud service provider and the associated encryption operations (See CSA CCM EKM-04). Cloud Service Providers (CSPs) can comply with best practices by offering Bring Your Own Key (BYOK) or Hold Your Own Key (HYOK) services to enable customer control of the keys used to encrypt their data. Customer key control allows for the separation, creation, ownership and control, including revocation, of encryption keys or tenant secrets used to create the keys.

CipherTrust Cloud Key Manager (CCKM) reduces key management complexity and operational costs by giving customers lifecycle control of cloud encryption keys with centralized management and visibility by leveraging cloud provider BYOK APIs or servicing HYOK requests.

Control Cloud Encryption Keys

- Leverage the value of BYOK and HYOK services with full-lifecycle cloud encryption key management
- Gain higher efficiency with centralized key management across hybrid, single- and multi-cloud environments, including key discovery, management of native cloud keys and automated key rotation
- Comply with the most stringent data protection mandates with secure key origination

Supported clouds and key management ownership models:

Amazon Web Services (AWS) KMS	Native	BYOK	
AWS CloudHSM	Native		
AWS XKS			HYOK
AWS China	Native	BYOK	
AWS GovCloud	Native	BYOK	HYOK
Google Cloud Platform CMEK	Native	BYOK	
Google Cloud Platform EKM			HYOK
Google Cloud Platform EKM UDE			HYOK-CC *
Google Workspace CSE			HYOK
IBM Cloud HPCS	Native	BYOK	
IBM Cloud Key Protect	Native	BYOK	
Microsoft Azure Cloud	Native	BYOK	
Microsoft Azure China	Native	BYOK	
Microsoft Azure GovCloud	Native	BYOK	
Microsoft Azure Managed HSMs	Native	BYOK	
Oracle Cloud Infrastructure	Native	BYOK	
Salesforce.com	Native	BYOK	HYOK **
Salesforce GovCloud Plus	Native	BYOK	HYOK **
Salesforce Sandbox	Native	BYOK	HYOK **
SAP Data Custodian	Native	BYOK	

*HYOK-CC is HYOK for Confidential Computing

** Cache-only Key Service

Best Practices

Industry and internal data protection mandates require increasing protection for sensitive cloud data. Meanwhile, CSA and industry analysts state that cloud encryption keys should be managed by customers. Key management can grow to hundreds of master keys per KMS—requiring knowledge of how, when, and by whom encryption keys are used, in addition to backing up, monitoring, rotating, expiring, archiving, suspending, restoring, and destroying keys as needed. CCKM provides comprehensive key lifecycle management to provide visibility on a single pane of glass, and automate and fulfill requirements for safe, comprehensive key management across multiple clouds.

Single Pane of Glass

Access to each cloud provider from a single browser window, across multiple accounts, regions, subscriptions and projects.

Improved Efficiency

CCKM offers multiple capabilities in support of increasing efficiency:

- Full management of cloud keys enabling centralized multicloud key management—even if that management is for native keys
- Automated synchronization ensuring that cloud console-specific key operations are visible in centralized key management
- Automated key rotation with support for expiring keys ensuring compliance while saving valuable time
- Key operations presented in the semantics of the cloud provider, removing the need to learn additional key technologies and terminology

Strong Encryption Key Security

Customer key control requires secure key generation and storage. CCKM leverages the security of the [CipherTrust Manager](#), [Luna Network HSM](#), or the Vormetric [Data Security Manager \(DSM\)](#) to create keys with up to FIPS 140-2 Level 3 security.

Hold Your Own Key

CCKM services respond to encryption key requests from cloud providers, supporting many of the emerging HYOK offerings: AWS External Key Store (XKS); Google Cloud External Key Management (EKM), Google EKM Ubiquitous Data Encryption (UDE), and Google Workspace Client-side Encryption for Drive, Gmail, Google Calendar and Google Meet; and Salesforce Cached Keys.

The Compliance Tools You Need

CipherTrust Cloud Key Manager visibility reports enable confirmation of compliance. Logs may also be directed to a syslog server or Security Information and Event Management (SIEM) tool.

Automation Tools Support Your Initiatives

CCKM capabilities are available programmatically using RESTful APIs, enabling the power of centralized cloud encryption management to work with your automation and self-service initiatives. Interactively explore the APIs with our built-in API playground.

Flexible Deployment Options

Deployment environments include: public cloud, private cloud, hybrid cloud, and physical appliances.

Regardless of how and where CCKM is deployed, CCKM can manage keys, and access to the keys, in any supported cloud.

To meet different organization's needs, CCKM is available in virtual and physical form factors: Virtual CCKM is an all-software offering easily deployed and can be run in the cloud or on premises and may be found in several cloud provider marketplaces for fast instantiation. Physical appliances are available for customers who prefer an on-prem solution.

Multi-cloud Data Security Solutions

CCKM simplifies the need to control and manage encryption keys for cloud services, a critical solution for fulfilling industry and organizational data protection mandates. CCKM and many additional Thales multi-cloud security products, including various [Bring Your Own Advanced Encryption](#) offerings, all with centralized key management, offer you choices on how best to protect all your cloud data. [It's your data in their cloud.](#)

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.



Solution Brief

Thales CipherTrust Cloud Key Management for **Salesforce**

Providing customers with
encryption key options

cpl.thalesgroup.com

THALES
Building a future we can all trust

Due to a growing number of requirements, including security and sovereignty mandates, organizations need to control their data. This is particularly challenging for enterprises using SaaS-based solutions since they do not directly manage, host, or even own the cloud-based platform containing their data.

Managing encryption keys and storing them securely is the cornerstone to meeting data security and sovereignty requirements. This is because encryption keys (also called cryptographic keys) are the strings of bits generated to encode and decode data. Whoever owns and manages the keys, therefore also ultimately owns the data. This is why organizations must prove they manage and control their encryption keys to be compliant with many digital sovereignty and data protection requirements.

Additionally, ensuring control over encryption keys can bolster security for SaaS-based applications in the event of a cyber-attack since the keys are stored and managed separately from the data.

The Solution

Thales and Salesforce have partnered to provide enterprise customers with choices in managing their data encryption keys for Salesforce data.

External Key Management

Thales can enable digital sovereignty for organizations using Salesforce by ensuring they retain more control and management over their encryption keys with our CipherTrust Cloud Key Management (CCKM) offering. By using Thales CCKM, organizations can safely store, manage, and maintain the tenant secrets used to derive the encryption keys that protect data within the Salesforce environment.

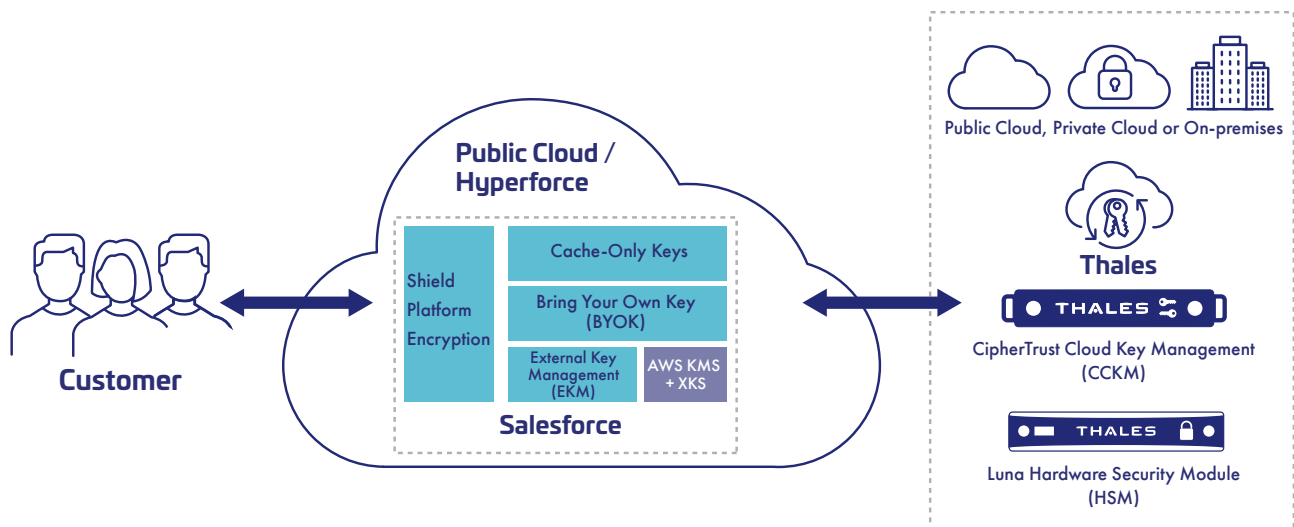
Thales CCKM delivers key generation, reporting, and key lifecycle management that helps fulfill internal and industry data protection mandates. To securely store encryption keys, customers can leverage Thales CipherTrust Manager with optional FIPS 140-2 Level 3 certified

HSM card or Luna Network Hardware Security Modules (HSMs) that securely protect cryptographic keys in a high-assurance, tamper-resistant, network-attached appliance that is FIPS 140-2 Level 3 certified.

By using Thales CCKM, Salesforce customers can implement Cache-Only Keys, Bring-Your-Own-Key (BYOK) and Hold-Your-Own-Key (HYOK) capabilities. By leveraging these options, customers retain more control of their encryption keys - and therefore also their encrypted data. This helps organizations meet compliance and best practice requirements for encryption key management, data security, and digital sovereignty.

Additionally, Thales supports Salesforce External Key Management (EKM) service for Salesforce Shield Platform Encryption, allowing users to provide and manage their own encryption keys that protect their Salesforce content. The combined Salesforce and Thales solution leverages the AWS External Key Store (XKS) service so customers can protect their Salesforce data using encryption keys stored physically outside the cloud.

Thales Encryption Key Management Solutions with Salesforce



Salesforce and Thales - Solution Overview

Flexible Deployment Options

Both Thales CCKM and Luna HSMs are available in both virtual and physical form factors. Virtual CCKM is an all-software offering easily deployed and can be run in the cloud or on-premises and may be found in several cloud provider marketplaces including AWS and Salesforce Hyperforce. Deployment environments include public cloud, private cloud, hybrid cloud, and physical appliances. Physical appliances are available for customers who prefer an on-premises solution.

Increased Efficiency

Thales CCKM centralizes encryption key management across all supported clouds, and multiple cloud accounts, regions, subscriptions, org IDs and projects, from a single pane of glass. Advanced cloud key management services and capabilities include automated key rotation, key expiration handling, and cloud key inventory--dramatically reducing the time required for cloud key life cycle management.

Best Practices

Together, Thales and Salesforce offer security best practices (such as CSA CCM EKM-04), control of customer data by separating key control from data encryption/decryption operations, and help gain operational insights on encryption key usage with reports and logs provided by Thales CCKM.

About Salesforce

Salesforce empowers companies of every size and industry to connect with their customers through the power of CRM + AI + Data + Trust. For more information about Salesforce, visit: www.salesforce.com.

About Thales

Today's businesses and governments depend on the cloud, data and software to deliver trusted digital services. That is why the most recognized brands and organizations around the world, rely on Thales to help them protect sensitive information and software wherever it is created, stored or accessed – from the cloud and data centers to devices and across networks. As the global leader in data security, identity & access management, and software licensing, our solutions enable organizations to move to the cloud securely, achieve compliance with confidence, create more value from their software and deliver seamless digital experiences for millions of consumers every day.

White Paper

NIST 800-57 **Recommendations** **for Key Management** **Requirements** **Analysis**

cpl.thalesgroup.com

THALES
Building a future we can all trust

The National Institute of Standards and Technology (NIST) Special Publication (SP) 800-57, Recommendations for Key Management Part 1 (Rev 5) provides guidance for cryptographic key management for U.S. Federal Government agencies. Part 1 of the publication outlines best practices for the management of cryptographic keys and discusses key management issues that must be addressed with using cryptography.

Importance of Securing Cryptographic Keys

The security of cryptographic processes is dependent on the security of the cryptographic keys used to encrypt the data. If the keys used to encrypt data are stolen with the encrypted data, the data is not secure because it can be deciphered and read in plain text.

NIST emphasizes the importance of protecting cryptographic keys in the publication, "The proper management of cryptographic keys is essential to the effective use of cryptography for security. Poor key management may easily compromise strong algorithms."¹ NIST states that:

*Ultimately, the security of information protected by cryptography directly depends on the strength of the keys, the effectiveness of cryptographic mechanisms and protocols associated with the keys, and the protection provided to the keys. Secret and private keys need to be protected against unauthorized disclosure, and all keys need to be protected against modification.*¹

For encryption to successfully secure sensitive data, the cryptographic keys themselves must be secured, managed and controlled by your organization and not a third-party or cloud provider. As agencies deploy ever-increasing numbers of siloed encryption solutions, they find themselves managing inconsistent policies, different levels of protection, and escalating costs.

The simplest path through this maze is to transition to a centralized key management model. Encryption key management involves administering the full lifecycle of cryptographic keys and protecting them from loss or misuse. Keys have a life cycle: They're created, live useful lives, and are retired. Key lifecycle management includes generating, using, storing, distributing, archiving, and deleting keys.

CipherTrust Data Security Platform

CipherTrust Data Security Platform (CDSP) offers a unified, scalable solution for protecting sensitive data across on-premises, cloud, and hybrid environments. Designed to simplify and strengthen data security, the platform centralizes critical capabilities such as data discovery, encryption, access controls, and compliance reporting.

CDSP is managed by CipherTrust Manager (CM), which is a centralized key management solution for enterprises, providing a unified console for managing encryption keys across on-premises and cloud environments. It simplifies key lifecycle management, including generation, backup, and deletion, while enforcing role-based access control and security policies. It also offers features like multi-tenancy support, robust auditing, and a REST API for custom solutions.

A cornerstone of CDSP is its robust key management functionality, which empowers organizations to securely create, store, rotate, and manage encryption keys across diverse systems and services. With support for FIPS 140-2 L1-compliant hardware security modules (HSMs) and integrations with major cloud providers, CDSP ensures that encryption keys remain under enterprise control—reducing risk and helping meet regulatory mandates.

CDSP streamlines operations with automated policies and centralized oversight, allowing security teams to confidently enforce data protection without disrupting performance. Whether securing databases, files, containers, or applications, CDSP delivers a comprehensive, policy-driven framework for managing and safeguarding critical data and the cryptographic keys that protect it.

Platform Benefits

Robust Key Management, Encryption and Tokenization

CDSP offers encryption solutions to protect data while it is stored on-premises, in the cloud, or in backups. CDSP supports transparent encryption, which means that data can be encrypted without requiring changes to applications or workflows. CDSP provides centralized key

¹<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NISTSP800-57pt1r5.pdf>

management to ensure that encryption keys are securely generated, stored, and managed. Tokenization replaces sensitive data with non-sensitive tokens, making it difficult for unauthorized individuals to access or misuse the data. CDSP supports dynamic data masking which allows sensitive data to be masked or redacted in real time, preventing unauthorized access even when data is in use.

Take Control of Your Sensitive Data Across Clouds

CDSP provides solutions for protecting data in public, private, and hybrid cloud environments. The platform integrates with popular cloud platforms such as AWS, Azure, and GCP. Centralizes multi cloud key management for BYOK, HYOK and cloud native encryption keys across any combination of clouds and on-premises with a single UI.

Integrate Security into Development improving DevSecOps efficiency

CDSP can be integrated into DevOps workflows to ensure that security is built into applications from the beginning. The platform provides solutions for protecting sensitive data during development, testing, and deployment. It rapidly secures, deploys and runs cloud-native workloads across cloud service providers and speed-up continuous integration and continuous delivery processes.

Managing Compliance and Risk

As part of CDSP, Data Risk Intelligence provides visibility into where your data sources that contain sensitive data are. It identifies the encryption status of each data source and offers recommendations on how to meet encryption goals. CDSP helps organizations comply with data protection regulations such as GDPR, HIPAA, and PCI DSS. By protecting sensitive data, CDSP can help organizations reduce the risk of data breaches and other security incidents. File Activity Monitoring (FAM) delivers real-time visibility into file usage, encryption status, and sensitive data access, enabling proactive threat detection and data protection across your organization.

Manage and Automate Access to Secrets

CipherTrust Secrets Management (CSM) offers a range of capabilities to securely store, manage, and distribute secrets across different developer environments, ensuring that they are protected from unauthorized access and misuse. It improves developer efficiency and centralizes management for all secret types with scalable SaaS.

Discovering and Classifying your Sensitive Data

CDSP helps organizations discover and classify sensitive data across their entire infrastructure, including files, databases, and big data. It provides visibility into where sensitive data resides and how it is being accessed, helping organizations assess their risk and prioritize protection efforts. It automatically scans and detects code containing sensitive information (API keys, tokens, passwords, etc.) mistakenly included by developers.

NIST SP 800-57 Requirements Mapping

Focusing on the capabilities needed to meet the requirements outlined in NIST SP 800-57, the following table provides details on CipherTrust Platform.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
6	Protection Requirements for Key Information		
6.1	Protection and Assurance Requirements	✓	CipherTrust Manager is available in both virtual and physical appliances, offering FIPS 140-2 L2 certifications to meet diverse security requirements. Virtual deployments can integrate with various network-accessible Hardware Security Modules (HSMs) to establish a FIPS 140 Level 3 root of trust. CipherTrust Manager supports deployment on-premises and across private or public cloud environments. For enhanced protection, its disk can be fully encrypted, with encryption initiated either during initial launch or applied to an already running instance.
6.1.1	Summary of Protection and Assurance Requirements for Cryptographic Keys	✓	
6.1.2	Summary of Protection Requirements for Other Related Information	✓	

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
6.2	Protection Mechanisms	✓	CipherTrust Manager is available as both virtual and physical appliances, offering FIPS 140-2 L2 certifications. The appliance supports full disk encryption and can be deployed in high-availability, Active/Active clustered configurations. Real-time replication of keys, policies, and configurations enables seamless disaster recovery and business continuity across multiple instances.
6.2.1	Protection Mechanisms for Key Information in Transit	✓	
6.2.1.1	Availability	✓	CipherTrust Manager ensures high availability through clustered deployment options and redundant communication paths between nodes.
6.2.1.2	Integrity	✓	Supports secure transmission of key materials and policies using TLS/SSL to maintain integrity during transit.
6.2.1.3	Confidentiality	✓	Utilizes TLS-encrypted communications and optional client authentication to ensure confidentiality of key exchanges and administrative sessions.
6.2.1.4	Association with Usage or Application	✓	Supports PKCS #11, JCE, .NET, and the Key Management Interoperability Protocol (KMIP) to bind keys to specific usage contexts and applications. Additionally, the SafeNet REST API allows integration with custom applications using enterprise-grade key management
6.2.1.5	Association with Other Entities	✓	Keys can be explicitly associated with users, applications, or services through role-based and attribute-based access controls.
6.2.1.6	Association with Other Related Key Information	✓	Offers granular access control via Attribute-Based Access Control (ABAC), ensuring that keys are only used in authorized contexts. Also supports: - Secure key distribution using SSL - Secure storage of key encryption keys in Luna Network HSMs
6.2.2	Protection Mechanisms for Key Information in Storage	✓	
6.2.2.1	Availability	✓	High-availability clustering and replication ensure keys remain accessible across failovers and maintenance events.
6.2.2.2	Integrity	✓	Key material is stored securely and verified using checksums and cryptographic validation mechanisms.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
6.2.2.3	Confidentiality	✓	When integrated with CipherTrust Transparent Encryption (CTE), keys are securely passed over encrypted channels. Admin access occurs via HTTPS, and communication between agents and the manager is encrypted using ephemeral session keys, minimizing exposure and enhancing confidentiality.
6.2.2.4	Association with Usage or Application	✓	CipherTrust Manager binds keys to specific applications or data sets through policy-driven associations and APIs, ensuring precise usage control.
6.2.2.5	Association with Other Entities	✓	Keys are mapped to users, roles, or services through defined policies and fine-grained access controls.
6.2.2.6	Association with Other Related Key Information	✓	Relationships between keys and associated metadata (e.g., rotation dates, ownership, tags) are preserved and managed via policy
6.2.3	Metadata for Keys	✓	Keys managed by CipherTrust Manager contain a single metadata record, which can link to multiple versions of the key. Each version is uniquely tracked, providing a detailed audit trail for compliance and lifecycle management.
7	Key States and Transitions		
7.1	Pre-activation State	✓	CipherTrust Manager provides centralized key lifecycle management for all CipherTrust Data Security Platform products. Built on an extensible microservices architecture, it supports all standard key states—from generation through retirement—ensuring secure and auditable transitions throughout the lifecycle. Key management capabilities include: - State transitions such as activation, suspension, deactivation, compromise, and destruction - Role-based access control (RBAC) for fine-grained key and policy management - Multi-tenancy support to isolate keys and operations across users or departments - Comprehensive auditing and reporting for key usage, operational changes, and policy enforcement CipherTrust Manager also supports secure key generation, backup and recovery, and policy-driven actions to manage compromised or decommissioned keys.
7.2	Active State	✓	
7.3	Suspended State	✓	
7.4	Deactivated State	✓	
7.5	Compromised State	✓	
7.6	Destroyed State	✓	

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8	Key-Management Phases and Functions		
8.1	Pre-operational Phase	✓	The CipherTrust Data Security Platform supports comprehensive key management functions that are logically separated from standard operations. These functions include domain creation, key creation, host registration, system initialization, and audit logging.
8.1.1	Entity Registration Function	✓	In multi-domain environments, users and administrators are logically isolated. Admins can select different encryption algorithms and key lengths per domain, increasing flexibility and control. Supported algorithms include AES-128 and AES-256.
8.1.2	System Initialization Function	✓	
8.1.3	Initialization Function	✓	
8.1.4	Keying-Material Installation Function	✓	Encrypted communications between CipherTrust Manager (CM) and agents are configurable.
8.1.5	Key Establishment Function	✓	The platform supports TLS 1.2+ using its REST API to ensure secure key delivery and API access. Key establishment is handled through CipherTrust Transparent Encryption (CTE) or agent-based solutions, where selectable encryption methods are applied per deployment requirement.
8.1.5.1	Generation and Distribution of Asymmetric Key Pairs	✓	Secure control paths exist between: - The daemon on the host - The SecFS kernel module responsible for system-level security Steps for key pair generation: - CM creates a public/private key pair - A Certificate Signing Request (CSR) is generated - The signed certificate is returned and stored in CM's secure database When operating in agent mode, the user space portion of the agent creates a local key pair and sends a CSR to CM. The signed certificate is returned and installed to complete registration. The kernel space portion also generates a key pair, and sends its public key to CM to complete secure host provisioning.
8.1.5.1.1	Distribution of Public Keys	✓	
8.1.5.1.1.1	Distribution of a Trust Anchor's Public Key in a PKI	✓	
8.1.5.1.1.2	Submission to a Registration Authority or Certification Authority	✓	
8.1.5.1.1.3	General Distribution of Static Public Keys	✓	
8.1.5.1.2	Distribution of Ephemeral Public Keys	✓	
8.1.5.1.3	Distribution of Centrally Generated Key Pairs	✓	

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8.1.5.2	Generation and Distribution of Symmetric Keys	✓	Key generation supports standards-compliant symmetric algorithms and configurable key lengths
8.1.5.2.1	Key Generation	✓	Key distribution is supported via: Manual distribution (for controlled environments)
8.1.5.2.2	Key Distribution	✓	Automated distribution and key wrapping
8.1.5.2.2.1	Manual Key Distribution	✓	Keys are associated with policies that define lifecycle states, access controls, and allowed usage
8.1.5.2.2.2	Automated Key Distribution/Key Transport/Key Wrapping	✓	
8.1.5.2.2.3	Key Agreement	✓	Support for key agreement protocols can be configured via CipherTrust Manager's API and agent-based workflows, depending on the environment and policy enforcement needs.
8.1.5.3	Generation and Distribution of Other Keying Material	✓	CipherTrust Manager supports the secure generation and distribution of other cryptographic materials, such as initialization vectors, shared secrets, RBG seeds, random numbers, and passwords. These are passed between the CipherTrust Manager and hosts via a one-time-use AES-256 random key generated by the Manager. The random key encrypts the desired encryption keys. This AES key is then encrypted using the kernel-space public key of the host. The encrypted payload is delivered to the host, where the kernel-space private key decrypts the random key. The decrypted random key is used to decrypt the actual encryption keys, which are then applied to the target file, directory, or executable. This method ensures secure in-transit handling of sensitive materials like initialization vectors, shared
8.1.5.3.1	Domain Parameters	✓	
8.1.5.3.2	Initialization Vectors	✓	
8.1.5.3.3	Shared Secrets	✓	
8.1.5.3.4	RBG Seeds	✓	
8.1.5.3.5	Other Public and Secret Information	✓	
8.1.5.3.6	Intermediate Results	✓	
8.1.5.3.7	Random Bits/Numbers	✓	
8.1.5.3.8	Passwords	✓	
8.1.6	Key Registration Functiony	✓	CipherTrust Manager supports automated key registration and lifecycle management, allowing keys to be registered during generation or securely imported from external systems. Registered keys are tagged, versioned, and governed by policies such as expiration, usage, and rotation.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8.2	Operational Phase	✓	CipherTrust Manager provides comprehensive tools for managing cryptographic keys throughout their operational lifecycle. This includes: • Key generation, import, export, and rotation • Support for symmetric and asymmetric key types • Control over key size, usage constraints, and policy enforcement All keys are stored in a centralized, hardware or virtual appliance, simplifying operations while enforcing strong access controls and auditability. CipherTrust Manager also supports Active/Active clustering for high availability, ensuring continuous uptime for encryption and key management services.
8.2.1	Normal Operational Storage Function	✓	
8.2.1.1	Cryptographic Module Storage	✓	All keys are securely stored inside CipherTrust Manager, a FIPS-certified cryptographic module, ensuring isolation, protection, and compliance with international standards.
8.2.1.2	Immediately Accessible Storage Media	✓	Keys are readily available to authorized services and applications during runtime, while access is tightly controlled through policy-based authorization, role-based access control (RBAC), and secure API communication.
8.2.2	Continuity of Operations Function	✓	CipherTrust Manager supports high-availability deployments, including Active/Active configurations, for continuous access to keys and services. This architecture allows for automatic failover and synchronized replication, ensuring that encryption and key management operations continue uninterrupted even during system events or outages.
8.2.2.1	Backup Storage	✓	CipherTrust Manager uses backup encryption keys to securely protect key backups. Administrators can: • Generate a new backup encryption key, or • Reuse an existing key when creating backups Backup keys are stored encrypted using a secure vault within CipherTrust Manager. Each backup may include critical data such as user keys or database dumps, all encrypted according to the system's configured key hierarchy. A final encrypted backup file is generated using the designated backup key, ensuring secure recovery and compliance with retention and protection policies.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8.2.2.2	Key Recovery Function	✓	CipherTrust Manager supports robust key recovery capabilities, including key generation, backup, restore, deactivation, and deletion. Keys are versioned and archived, supporting rapid recovery when needed. With version control and Thales Live Data Transformation (CTE-LDT), restored keys can be automatically aligned to historical data states. During recovery, keys retrieved from backup are securely applied to the relevant datasets, with encryption maintained using current cryptographic policies.
8.2.3	Key Change Function	✓	CipherTrust Manager supports:
8.2.3.1	Re-keying	✓	Key change for replacing compromised or outdated keys
8.2.3.2	Key Update Function	✓	Re-keying, enabling seamless replacement of keys without interrupting operations Key update workflows that apply newer keys to older datasets as part of a versioned lifecycle process All updates are governed by policy-based controls and audit logging for traceability.
8.2.4	Key Derivation Methods	✓	Enforces secure key derivation methods, centralizes control, automates policy enforcement, and ensures compliant key lifecycle management.
8.3	Post-Operational Phase	✓	
8.3.1	Key Archive and Key Recovery Functions	✓	Securely archiving keys, enabling authorized recovery, enforcing access controls, and ensuring compliance with cryptographic key management policies.
8.2.3	Key Change Function	✓	CipherTrust Manager supports:
8.2.3.1	Re-keying	✓	Key change for replacing compromised or outdated keys
8.2.3.2	Key Update Function	✓	- Re-keying, enabling seamless replacement of keys without interrupting operations - Key update workflows that apply newer keys to older datasets as part of a versioned lifecycle process All updates are governed by policy-based controls and audit logging for traceability.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8.2.4	Key Derivation Methods	✓	CipherTrust supports secure key derivation methods consistent with industry standards. Derived keys are created based on unique, application-specific inputs and comply with approved cryptographic standards for secure usage.
8.3	Post-Operational Phase	✓	CipherTrust Manager retains cryptographic materials per policy after their active use has ended, allowing for secure archival, auditing, or decommissioning.
8.3.1	Key Archive and Key Recovery Functions	✓	CipherTrust Manager supports secure archiving of retired or inactive keys. Archived keys remain accessible for approved recovery scenarios and are stored using encryption consistent with active key policies. Archived keys can be restored based on versioned metadata Policy controls govern access and retention
8.3.2	Entity De-registration Function	✓	CipherTrust Manager allows for full lifecycle control of clients (e.g., ProtectFile, ProtectV, CTE). Administrators can: • Register and manage client entities (excluding ProtectFile clients) • View, modify, revoke, or delete client registrations as needed Once a client is deregistered, all communication with CipherTrust Manager is terminated, ensuring immediate and complete disassociation.
8.3.3	Key De-registration Function	✓	Within CipherTrust Manager there is a Key Admins group. Key Administrators have permissions to managing keys on the system. They can: • create or modify their own keys • perform key management operations on keys created by all users on the system
8.3.4	Key Destruction Function	✓	
8.3.5	Key Revocation Function	✓	
8.4	Destroyed Phase	✓	There is a System Defined Group named "CTE Admins". Users within the "CTE Admins" group are CTE Administrators. Only users of the "CTE Admins" group can delete CTE keys. Depending on the Key Management usecase a specified admin within CipherTrust Manager will be able to manage keys or delete keys that are not in use.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
8.3.3	Key De-registration Function	✓	Within CipherTrust Manager, a designated Key Admins group manages key de-registration. Admins have permissions to: • Create and manage their own keys • Perform operations on other keys based on their assigned privileges Key de-registration removes keys from operational use but retains metadata for audit if required.
8.3.4	Key Destruction Function	✓	Only users in the System Defined Group: "CTE Admins" have permission to permanently destroy keys (such as CTE policies and associated keys). Keys marked for deletion are securely removed based on policy and usage state Only unused or expired keys can be deleted, ensuring cryptographic continuity
8.3.5	Key Revocation Function	✓	CipherTrust Manager supports key revocation based on usage status or compromise events. Revoked keys are: • Immediately marked as inactive • Prevented from being used in future encryption or decryption operations • Tracked in audit logs for compliance and incident response
8.4	Destroyed Phase	✓	The Destroyed Phase ensures that cryptographic keys are irretrievably deleted when no longer needed. Only authorized administrators (e.g., CTE Admins) can initiate destruction Keys must be inactive or non-operational before destruction Destruction operations are logged to maintain audit integrity
9	Additional Considerations		
9.1	Access Control and Identity Authentication	✓	CipherTrust Manager enhances access control through kernel-level agents and AES-256 encryption, surpassing typical OS-level controls. It enforces least privilege access, allowing only authenticated administrators and clients to communicate with the Manager. Integration with role-based access controls ensures secure, policy-driven identity management.

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
9.2	Inventory Management	✓	CipherTrust Manager provides centralized inventory visibility and lifecycle management for all cryptographic keys and certificates across the CipherTrust Data Security Platform. Its microservices-based architecture simplifies operations such as key generation, rotation, backup, archival, and deletion.
9.2.1	Key Inventories	✓	All cryptographic keys are cataloged with metadata such as key type, usage, status, creation date, and expiration, enabling full visibility and control across environments.
9.2.2	Certificate Inventories	✓	CipherTrust Manager tracks certificates used for encryption, authentication, and digital signing. Each certificate is indexed for quick retrieval and lifecycle management.
9.3	Accountability	✓	All user and administrative actions within CipherTrust Manager are fully auditable and attributable through detailed logs. These logs include timestamps, user roles, affected objects (e.g., keys, policies), and the nature of each action, helping ensure accountability and support for internal or external audits.
9.4	Audit	✓	CipherTrust Manager generates comprehensive audit logs capturing all key lifecycle operations (e.g., generation, access, modification, rotation, deletion). Logs are tamper-resistant and can be forwarded to external SIEM systems or retained locally for compliance. This capability ensures full traceability for audit and compliance purposes.
9.5	Key-Management System Survivability	✓	CipherTrust Manager supports high availability through clustered deployments with real-time replication of keys, policies, and configuration data across appliances. This architecture supports: - Seamless failover and disaster recovery - No disruption to cryptographic operations - Centralized key management across distributed environments with minimal performance impact

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
9.5.1	Backed Up and Archived Key	✓	CipherTrust Manager automatically creates versioned backups of all cryptographic keys. Backups: • Are stored securely on the appliance • Remain available until explicitly deleted • Can be restored on the same or another appliance • Are always encrypted using a designated backup key If a specific backup key is not provided, a default backup key is used to ensure data protection and future recovery.
9.5.2	Key Recovery	✓	During key recovery, backup files—encrypted using the assigned or default backup key—can be: • Downloaded from the appliance • Re-uploaded to the same or a different CipherTrust Manager appliance • Decrypted and restored using the required backup key This enables secure restoration of archived encryption keys for operational continuity and compliance with retention policies.
9.5.3	System Redundancy/Contingency Planning	✓	CipherTrust Manager supports robust system redundancy and contingency planning through clustered deployments with real-time replication of keys, policies, and configurations across multiple appliances. This setup ensures high availability, seamless failover, and disaster recovery Organizations with multiple encryption endpoints can centrally manage keys without disrupting system performance Business continuity is preserved even during infrastructure failure scenarios

NIST 800-57 Reference	Requirement	Requirement Met	CipherTrust Platform
9.5.3.1	General Principles	✓	CipherTrust Manager supports Active/Active high availability deployments to ensure minimal downtime. Designed for 24x7 operations with full support for uninterrupted key management and encryption services All keys and related policies are versioned and securely stored, allowing seamless recovery in the event of failure or compromise
9.5.3.2	Cryptography and Key-Management-Specific Recovery Issues	✓	CipherTrust Manager maintains comprehensive backups and versioning of keys. These capabilities allow: • Recovery of archived encryption keys aligned with the original version of protected data • Secure application of keys to historical datasets through Live Data Transformation, preserving data integrity • Controlled and auditable recovery procedures in line with cryptographic best practices
9.5.4	Compromise Recovery	✓	In the event of a key compromise, CipherTrust Manager enables recovery via Live Data Transformation, which allows data encrypted with a compromised key to be seamlessly re-encrypted using a new, secure key. This eliminates the need to re-ingest or decrypt raw data manually Recovery operations are automated, secure, and aligned with enterprise-grade data protection policies

About Thales

Thales is a global leader in cybersecurity, helping the most trusted organizations protect critical applications, data, identities, and software anywhere, at scale. Through Thales' integrated platforms, customers achieve better visibility of risks, defend against cyber threats, close compliance gaps, and deliver trusted digital experiences for billions of consumers every day.



Contact us

For contact information,
please visit cpl.thalesgroup.com/contact-us

cpl.thalesgroup.com



CipherTrust Cloud Key Manager REST API Documentation

Copyright 2009 - 2019. Thales e-Security, Inc. All rights reserved.
More information: <http://help.thalesecurity.com>
Contact Info: support@thalesecurity.com
Version: 1.6.1
BasePath:/kmaas
All rights reserved
<http://apache.org/licenses/LICENSE-2.0.html>

Access

Methods

[[Jump to Models](#)]

Table of Contents

[AggregationController](#)

- [GET /aggregate](#)

[AwsKeyEntity](#)

- [PUT /rest/awsKeys/{id}/autorotate](#)
- [PUT /rest/awsKeys/{id}/cancelDeletion](#)
- [POST /rest/awsKeys/create](#)
- [PUT /rest/awsKeys/{id}/delete](#)
- [GET /rest/awsKeys](#)
- [GET /rest/awsKeys/{id}](#)
- [GET /rest/awsKeys/keyByKeyId](#)
- [GET /rest/awsKeys/keys](#)
- [PUT /rest/awsKeys/{id}/import](#)
- [POST /rest/awsKeys/rotate](#)
- [POST /rest/awsKeys/synchronize](#)
- [PUT /rest/awsKeys/{id}/updateKeyPolicy](#)
- [PUT /rest/awsKeys/{id}/update](#)
- [POST /rest/awsKeys/upload](#)
- [POST /rest/awsKeys/verifyAlias](#)

[AwsScheduleEntity](#)

- [DELETE /awsSchedules/{id}](#)
- [POST /awsSchedules](#)
- [GET /rest/awsSchedules](#)
- [GET /rest/awsSchedules/search/findByUsernameAndJobTypeAndTenant](#)
- [GET /rest/awsSchedules/{id}](#)
- [GET /awsSchedules/schedule](#)
- [PATCH /rest/awsSchedules/{id}](#)
- [POST /rest/awsSchedules](#)
- [PUT /rest/awsSchedules/{id}](#)
- [PUT /awsSchedules/{id}](#)

[AzureKeyEntity](#)

- [PUT /rest/azureKeys/{id}/autorotate](#)
- [POST /rest/azureKeys/create](#)
- [PUT /rest/azureKeys/{id}/deleteBackup](#)
- [PUT /rest/azureKeys/{id}/delete](#)
- [GET /rest/azureKeys](#)
- [GET /rest/azureKeys/search/findByAutoRotateUserAndAutoRotate](#)
- [GET /rest/azureKeys/search/findByAutoRotateUserAndAutoRotateOnExpire](#)
- [GET /rest/azureKeys/search/findByAutoRotateUser](#)
- [GET /rest/azureKeys/search/findById](#)
- [GET /rest/azureKeys/search/findByKeyId](#)
- [GET /rest/azureKeys/search/findByKeyIdIgnoreCase](#)
- [GET /rest/azureKeys/search/findByName](#)
- [GET /rest/azureKeys/search/findByNameRegexAndKeyVaultRegex](#)
- [GET /rest/azureKeys/search/findByNameRegexOrKeyVaultRegexOrVersionsVersionRegexOrVersionsTagsUploadedFrom](#)
- [GET /rest/azureKeys/findKeyByKeyId](#)
- [GET /rest/azureKeys/{id}](#)
- [GET /rest/azureKeys/getIdByKeyId](#)
- [PUT /rest/azureKeys/{id}/purge](#)
- [PUT /rest/azureKeys/{id}/recover](#)
- [PUT /rest/azureKeys/{id}/restore](#)
- [GET /rest/azureKeys/search/searchAndSortKeys](#)
- [GET /rest/azureKeys/keys](#)
- [POST /rest/azureKeys/synchronize](#)
- [PUT /rest/azureKeys/{id}/update](#)
- [POST /rest/azureKeys/uploadKeyFileBLOB](#)
- [POST /rest/azureKeys/upload](#)

[AzureKeyVaultController](#)

- [GET /rest/azureKeyVaults/{name}](#)
- [GET /rest/azureKeyVaults](#)
- [GET /rest/azureKeyVaults/getSubscriptionList](#)

[AzureScheduleEntity](#)

- [POST /azureSchedules](#)
- [DELETE /azureSchedules/{id}](#)
- [GET /rest/azureSchedules](#)
- [GET /rest/azureSchedules/search/findByUsernameAndJobTypeAndTenant](#)
- [GET /rest/azureSchedules/{id}](#)
- [GET /azureSchedules/schedule](#)
- [PATCH /rest/azureSchedules/{id}](#)
- [POST /rest/azureSchedules](#)
- [PUT /rest/azureSchedules/{id}](#)
- [PUT /azureSchedules/{id}](#)

[CloudServiceInfoController](#)

- [GET /csp](#)

[KeyProducerController](#)

- [GET /rest/keyProducers](#)

KeyVaultController

- [POST /keyvaultkey](#)
- [DELETE /keyvaultkey/{id}](#)
- [GET /keyvaultkey/key/{id}](#)
- [GET /keyvaultkey/keys](#)

LicenseController

- [GET /license](#)

ReportEntity

- [GET /rest/reports](#)
- [GET /rest/reports/search/findByIdOrName](#)
- [GET /rest/reports/search/findById](#)
- [GET /rest/reports/search/findByNameAndTenant](#)
- [GET /rest/reports/search/findByName](#)
- [GET /rest/reports/{id}](#)

SalesforceKeyEntity

- [POST /salesforceKeys/create](#)
- [DELETE /salesforceKeys/{id}](#)
- [PUT /salesforceKeys/{id}/destroy](#)
- [POST /salesforceKeys/synchronize](#)
- [GET /rest/salesforceKeys](#)
- [GET /rest/salesforceKeys/search/findByName](#)
- [GET /rest/salesforceKeys/search/findByStatus](#)
- [GET /rest/salesforceKeys/search/findByTenantSecret_Id](#)
- [GET /rest/salesforceKeys/search/findByTenantSecret_StatusAndType](#)
- [GET /rest/salesforceKeys/search/findByType](#)
- [GET /rest/salesforceKeys/{id}](#)
- [GET /salesforceKeys/{id}](#)
- [GET /salesforceKeys/keys](#)
- [PUT /salesforceKeys/{id}/import](#)
- [PUT /salesforceKeys/{id}](#)
- [POST /salesforceKeys/upload](#)

SalesforceScheduleEntity

- [POST /salesforceSchedules](#)
- [DELETE /salesforceSchedules/{id}](#)
- [GET /rest/salesforceSchedules](#)
- [GET /rest/salesforceSchedules/search/findByUsernameAndJobTypeAndTenant](#)
- [GET /rest/salesforceSchedules/search/findByUsername](#)
- [GET /rest/salesforceSchedules/{id}](#)
- [GET /salesforceSchedules/schedule](#)
- [PATCH /rest/salesforceSchedules/{id}](#)
- [POST /rest/salesforceSchedules](#)
- [PUT /rest/salesforceSchedules/{id}](#)
- [PUT /salesforceSchedules/{id}/toggleAutoRotate](#)
- [PUT /salesforceSchedules/{id}](#)

SessionDataController

- [GET /rest/serviceSessions/myAzureCurrentDirectory](#)
- [GET /rest/serviceSessions/search/findByCloudService](#)
- [GET /rest/serviceSessions/search/findByUser](#)
- [GET /rest/serviceSessions/search/findByTenant](#)
- [GET /rest/serviceSessions/{id}](#)
- [GET /rest/serviceSessions/myAzureTenantIds](#)
- [GET /rest/serviceSessions](#)

SystemInfoController

- [GET /systemInfo](#)

UserInfoController

- [GET /me](#)

AggregationController

GET /aggregate

Up

Gets the aggregation metrics (runAggregationUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- * / *

Responses

200

OK [String](#)
401
Unauthorized
403
Forbidden
404
Not Found

AwsKeyEntity

PUT [Up](#) `/rest/awsKeys/{id}/autorotate`

Enable or disable an AWS key to be auto rotated in this user's KEY_ROTATION schedule (`autorotateUsingPUT`)

Path parameters

id (required)
Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

data [AutoRotateRequestData](#) (required)
Body Parameter — data

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `*/*`

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT [Up](#) `/rest/awsKeys/{id}/cancelDeletion`

Cancel a scheduled deletion of AWS key (`cancelDeletionUsingPUT`)

Path parameters

id (required)
Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `*/*`

Responses

200
OK [String](#)
201

Created
401
Unauthorized
403
Forbidden
404
Not Found

POST /rest/awsKeys/create

Up

Create an AWS key. The key material will be generated in AWS KMS (createKeysUsingPOST)

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

data [UploadKeyRequestData](#) (required)
Body Parameter — data

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[AwsKey](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [AwsKey](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /rest/awsKeys/{id}/delete

Up

Delete AWS key material, or schedule a deletion of AWS key (deleteAwsKeyUsingPUT)

Path parameters

id (required)
Path Parameter — id

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

deleteRequest [DeleteRequestBody](#) (required)
Body Parameter — deleteRequest

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
201
Created
401

GET /rest/awsKeys

Up

findAllAwsKey (findAllAwsKeyUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

page (optional)
Query Parameter — page
size (optional)
Query Parameter — size
sort (optional)
Query Parameter — sort

Return type

[Resources-AwsKey-](#)

Example data

Content-Type: application/json

```
{
  "_embedded": {
    "awsKeys": [
      {
        "aliases": [
          {
            "arn": "string",
            "name": "string"
          }
        ],
        "autoRotate": true,
        "autoRotateDisableEncrypt": true,
        "autoRotateKeyProvider": "DSM",
        "autoRotateOnExpire": true,
        "autoRotateUser": "string",
        "availableInKeyVault": true,
        "backup": {
          "algorithm": "string",
          "allTenant": true,
          "blob": "string",
          "blobUUID": "string",
          "cckmId": "string",
          "certificate": "string",
          "createdAt": "2019-03-25T23:09:53.167Z",
          "createdBy": "string",
          "deleted": true,
          "description": "string",
          "hashType": "string",
          "id": {
            "counter": 0,
            "date": "2019-03-25T23:09:53.167Z",
            "machineIdentifier": 0,
            "processIdentifier": 0,
            "time": 0,
            "timeSecond": 0,
            "timestamp": 0
          },
          "keySource": "KMaaS",
          "modifiedAt": "2019-03-25T23:09:53.167Z",
          "modifiedBy": "string",
          "nShieldKeyBlob": "string",
          "name": "string",
          "privateKey": {
            "hash": "string",
            "length": 0,
            "value": "string"
          },
          "protected": true,
          "provider": "DSM",
          "publicKey": {
            "hash": "string",
            "length": 0,
            "value": "string"
          },
          "secretKey": {
            "hash": "string",
            "length": 0,
            "value": "string"
          },
          "service": "NONE",
          "serviceKeyPage": {
            "content": [
              {
                "availableInKeyVault": true,
                "createdAt": "2019-03-25T23:09:53.167Z",
                "createdBy": "string",
                "id": {
                  "counter": 0,
                  "date": "2019-03-25T23:09:53.167Z",
                  "machineIdentifier": 0,
                  "processIdentifier": 0,
                  "time": 0,
                  "timeSecond": 0,
                  "timestamp": 0
                }
              }
            ]
          }
        }
      }
    ]
  }
}
```

```

    },
    "keyType": "AES256",
    "modifiedAt": "2019-03-25T23:09:53.167Z",
    "modifiedBy": "string",
    "name": "string",
    "source": "KMaas",
    "status": "CREATED",
    "syncedAt": "2019-03-25T23:09:53.167Z",
    "tenant": {
      "cloudService": "string",
      "createdAt": "2019-03-25T23:09:53.167Z",
      "createdBy": "string",
      "id": {
        "counter": 0,
        "date": "2019-03-25T23:09:53.167Z",
        "machineIdentifier": 0,
        "processIdentifier": 0,
        "time": 0,
        "timeSecond": 0,
        "timestamp": 0
      },
      "modifiedAt": "2019-03-25T23:09:53.167Z",
      "modifiedBy": "string",
      "name": "string"
    },
    "uploadedAt": "2019-03-25T23:09:53.167Z",
    "uploadedBy": "string"
  }
},
"first": true,
"last": true,
"number": 0,
"numberOfElements": 0,
"size": 0,
"sort": {},
"totalElements": 0,
"totalPages": 0
},
"sourceKeyTier": "string",
"tenant": {
  "cloudService": "string",
  "createdAt": "2019-03-25T23:09:53.167Z",
  "createdBy": "string",
  "id": {
    "counter": 0,
    "date": "2019-03-25T23:09:53.167Z",
    "machineIdentifier": 0,
    "processIdentifier": 0,
    "time": 0,
    "timeSecond": 0,
    "timestamp": 0
  },
  "modifiedAt": "2019-03-25T23:09:53.167Z",
  "modifiedBy": "string",
  "name": "string"
}
},
"createdAt": "2019-03-25T23:09:53.167Z",
"createdBy": "string",
"grants": [
  {
    "constraints": {
      "encryptionContextEquals": {
        "additionalProp1": "string",
        "additionalProp2": "string",
        "additionalProp3": "string"
      },
      "encryptionContextSubset": {
        "additionalProp1": "string",
        "additionalProp2": "string",
        "additionalProp3": "string"
      }
    },
    "creationDate": "2019-03-25T23:09:53.167Z",
    "grantId": "string",
    "granteePrincipal": "string",
    "issuingAccount": "string",
    "keyId": "string",
    "name": "string",
    "operations": [
      "string"
    ],
    "retiringPrincipal": "string"
  }
],
"id": {
  "counter": 0,
  "date": "2019-03-25T23:09:53.167Z",
  "machineIdentifier": 0,
  "processIdentifier": 0,
  "time": 0,
  "timeSecond": 0,
  "timestamp": 0
},
"keyMetadata": {
  "arn": "string",
  "awsaccountId": "string",
  "creationDate": "2019-03-25T23:09:53.167Z",
  "deletionDate": "2019-03-25T23:09:53.167Z",
  "description": "string",
  "enabled": true,
  "expirationModel": "string",
  "keyId": "string",
  "keyManager": "string",
  "keyState": "string",
  "keyUsage": "string",
  "origin": "string",
  "validTo": "2019-03-25T23:09:53.167Z"
},
"keyType": "AES256",
"modifiedAt": "2019-03-25T23:09:53.167Z",
"modifiedBy": "string",
"name": "string",
"originalKey": {
  "algorithm": "string",
  "allTenant": true,

```



```

"blob": "string",
"blobUUID": "string",
"ckcmId": "string",
"certificate": "string",
"createdAt": "2019-03-25T23:09:53.167Z",
"createdBy": "string",
"deleted": true,
"description": "string",
"hashType": "string",
"id": {
  "counter": 0,
  "date": "2019-03-25T23:09:53.167Z",
  "machineIdentifier": 0,
  "processIdentifier": 0,
  "time": 0,
  "timeSecond": 0,
  "timestamp": 0
},
"keySource": "KMaaS",
"modifiedAt": "2019-03-25T23:09:53.167Z",
"modifiedBy": "string",
"nShieldKeyBlob": "string",
"name": "string",
"privateKey": {
  "hash": "string",
  "length": 0,
  "value": "string"
},
"protected": true,
"provider": "DSM",
"publicKey": {
  "hash": "string",
  "length": 0,
  "value": "string"
},
"secretKey": {
  "hash": "string",
  "length": 0,
  "value": "string"
},
"service": "NONE",
"serviceKeyPage": {
  "content": [
    {
      "availableInKeyVault": true,
      "createdAt": "2019-03-25T23:09:53.167Z",
      "createdBy": "string",
      "id": {
        "counter": 0,
        "date": "2019-03-25T23:09:53.167Z",
        "machineIdentifier": 0,
        "processIdentifier": 0,
        "time": 0,
        "timeSecond": 0,
        "timestamp": 0
      },
      "keyType": "AES256",
      "modifiedAt": "2019-03-25T23:09:53.167Z",
      "modifiedBy": "string",
      "name": "string",
      "source": "KMaaS",
      "status": "CREATED",
      "syncedAt": "2019-03-25T23:09:53.167Z",
      "tenant": {
        "cloudService": "string",
        "createdAt": "2019-03-25T23:09:53.167Z",
        "createdBy": "string",
        "id": {
          "counter": 0,
          "date": "2019-03-25T23:09:53.167Z",
          "machineIdentifier": 0,
          "processIdentifier": 0,
          "time": 0,
          "timeSecond": 0,
          "timestamp": 0
        },
        "modifiedAt": "2019-03-25T23:09:53.167Z",
        "modifiedBy": "string",
        "name": "string"
      },
      "uploadedAt": "2019-03-25T23:09:53.167Z",
      "uploadedBy": "string"
    }
  ],
  "first": true,
  "last": true,
  "number": 0,
  "numberOfElements": 0,
  "size": 0,
  "sort": {},
  "totalElements": 0,
  "totalPages": 0
},
"sourceKeyTier": "string",
"tenant": {
  "cloudService": "string",
  "createdAt": "2019-03-25T23:09:53.168Z",
  "createdBy": "string",
  "id": {
    "counter": 0,
    "date": "2019-03-25T23:09:53.168Z",
    "machineIdentifier": 0,
    "processIdentifier": 0,
    "time": 0,
    "timeSecond": 0,
    "timestamp": 0
  },
  "modifiedAt": "2019-03-25T23:09:53.168Z",
  "modifiedBy": "string",
  "name": "string"
},
"policy": {
  "id": "string",
  "statements": [
    {

```

```

    "actions": [
      {
        "actionName": "string"
      }
    ],
    "conditions": [
      {
        "conditionKey": "string",
        "type": "string",
        "values": [
          "string"
        ]
      }
    ],
    "effect": "Allow",
    "id": "string",
    "principals": [
      {
        "pid": "string",
        "provider": "string"
      }
    ],
    "resources": [
      {
        "id": "string"
      }
    ]
  },
  "version": "string"
},
"region": "string",
"regionLabel": "string",
"rotationChainId": "string",
"rotationStatus": "ARCHIVED",
"source": "KMaaS",
"status": "CREATED",
"syncedAt": "2019-03-25T23:09:53.168Z",
"tags": [
  {
    "tagKey": "string",
    "tagValue": "string"
  }
],
"tenant": {
  "cloudService": "string",
  "createdAt": "2019-03-25T23:09:53.168Z",
  "createdBy": "string",
  "id": {
    "counter": 0,
    "date": "2019-03-25T23:09:53.168Z",
    "machineIdentifier": 0,
    "processIdentifier": 0,
    "time": 0,
    "timeSecond": 0,
    "timestamp": 0
  },
  "modifiedAt": "2019-03-25T23:09:53.168Z",
  "modifiedBy": "string",
  "name": "string"
},
"uploadedAt": "2019-03-25T23:09:53.168Z",
"uploadedBy": "string",
"uploadedFrom": "string"
}
},
"_links": {
  "additionalProp1": {
    "href": "string",
    "rel": "string",
    "templated": true
  },
  "additionalProp2": {
    "href": "string",
    "rel": "string",
    "templated": true
  },
  "additionalProp3": {
    "href": "string",
    "rel": "string",
    "templated": true
  }
}
}
}

```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json
- application/hal+json
- text/uri-list
- application/x-spring-data-compact+json

Responses

200

OK [Resources-AwsKey-](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/awsKeys/{id}

findOneAwsKey (findOneAwsKeyUsingGET)

Path parameters

id (required)

Path Parameter – id

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

[Resource~AwsKey~>](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [Resource~AwsKey~>](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/awsKeys/keyByKeyId

Up

Find AWS key by key id ([getKeyByKeyIdUsingGET](#))

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

keyId (required)

Query Parameter — keyId

Return type

[AwsKeyResponse](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [AwsKeyResponse](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/awsKeys/keys

Up

Get AWS keys by searching name, region, enabled, keyState, rotationStatus, creationDateFrom, etc. ([getKeyUsingGET](#))

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

data (optional)

Query Parameter — data

enabled (optional)

Query Parameter — enabled

keyMaterialOrigin (optional)

Query Parameter — keyMaterialOrigin

keyState (optional)

Query Parameter — keyState

name (optional)

Query Parameter — name

page (optional)
Query Parameter — format: int32

region (optional)
Query Parameter — region

rotationStatus (optional)
Query Parameter — rotationStatus

search (optional)
Query Parameter — search

size (optional)
Query Parameter — format: int32

sort (optional)
Query Parameter —

Return type
[Page](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Page](#)
401
Unauthorized
403
Forbidden
404
Not Found

PUT /rest/awsKeys/{id}/import

Import key material of a CCKM source key to AWS key ([importAwsKeyUsingPUT](#))

Up

Path parameters
id (required)
Path Parameter — id

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
data [ImportKeyRequestData](#) (optional)
Body Parameter — data

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

POST /rest/awsKeys/rotate

Rotate an Aws key. This upload a new AWS key, and move aliases from this to the new key ([rotateAwsKeyUsingPOST](#))

Up

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
data [RotateKeyRequestData](#) (required)
Body Parameter — data

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

POST /rest/awsKeys/synchronize Up

Synchronize all AwsKeys that this user has proper permissions in the given regions (`synchronizeAllKeysUsingPOST`)

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
`regions` [string](#) (optional)
Body Parameter — regions

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /rest/awsKeys/{id}/updateKeyPolicy Up

Update AWS key policy (`updateKeyPolicyUsingPUT`)

Path parameters
`id` (required)
Path Parameter — id

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
`data` [UpdateKeyPolicyRequestData](#) (required)
Body Parameter — data

Request headers
Cookie

Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /rest/awsKeys/{id}/update

Up

Update AWS key (updateKeysUsingPUT)

Path parameters

id (required)
Path Parameter — AwsKey id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

data [UpdateKeyRequestData](#) (required)
Body Parameter — data

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[AwsKey](#)

Example data
Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [AwsKey](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

POST /rest/awsKeys/upload

Up

Upload a CCKM source key to AWS (uploadKeysUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

data [UploadKeyRequestData](#) (required)
Body Parameter — data

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN

Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[AwsKey](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [AwsKey](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

POST /rest/awsKeys/verifyAlias [Up](#)

Verify AWS key alias ([verifyAliasInAwsUsingPOST](#))

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
data [VerifyAliasData](#) (required)
Body Parameter — data

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
Boolean

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Boolean](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

AwsScheduleEntity

DELETE /awsSchedules/{id} [Up](#)

Delete an AwsSchedule ([createUsingDELETE](#))

Path parameters
id (required)
Path Parameter — id

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [String](#)

204

No Content

401

Unauthorized

403

Forbidden

POST /awsSchedules

Up

Create an AwsSchedule (createUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

schedule [AwsSchedule](#) (optional)

Body Parameter – Required parameters:

jobType,

pause,

cron or (startDate, interval, and intervalUnit),

username,

accessKeyId,

password // AWS Secret Access Key

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [String](#)

201

Created

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/awsSchedules

Up

findAllAwsSchedule (findAllAwsScheduleUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

page (optional)

Query Parameter – page

size (optional)

Query Parameter – size

sort (optional)

Query Parameter – sort

Return type

[Resources-AwsSchedule-](#)

Example data

Content-Type: application/json


```

{
  "_embedded": {
    "awsSchedules": [
      {
        "accessKeyId": "string",
        "applyOnExpiration": true,
        "createdAt": "2019-03-25T23:38:36.281Z",
        "createdBy": "string",
        "cron": "string",
        "description": "string",
        "fixedRateSchedule": true,
        "humanReadableCron": "string",
        "id": {
          "counter": 0,
          "date": "2019-03-25T23:38:36.281Z",
          "machineIdentifier": 0,
          "processIdentifier": 0,
          "time": 0,
          "timeSecond": 0,
          "timestamp": 0
        },
        "interval": 0,
        "intervalKeyExpiration": 0,
        "intervalUnit": "Hour",
        "intervalUnitKeyExpiration": "Hour",
        "jobType": "KEY_ROTATION",
        "modifiedAt": "2019-03-25T23:38:36.281Z",
        "modifiedBy": "string",
        "name": "string",
        "password": "string",
        "pause": true,
        "startDate": "2019-03-25T23:38:36.281Z",
        "tenant": {
          "cloudService": "string",
          "createdAt": "2019-03-25T23:38:36.281Z",
          "createdBy": "string",
          "id": {
            "counter": 0,
            "date": "2019-03-25T23:38:36.281Z",
            "machineIdentifier": 0,
            "processIdentifier": 0,
            "time": 0,
            "timeSecond": 0,
            "timestamp": 0
          },
          "modifiedAt": "2019-03-25T23:38:36.281Z",
          "modifiedBy": "string",
          "name": "string"
        },
        "username": "string"
      }
    ]
  },
  "_links": {
    "additionalProp1": {
      "href": "string",
      "rel": "string",
      "templated": true
    },
    "additionalProp2": {
      "href": "string",
      "rel": "string",
      "templated": true
    },
    "additionalProp3": {
      "href": "string",
      "rel": "string",
      "templated": true
    }
  }
}

```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json
- application/hal+json
- text/uri-list
- application/x-spring-data-compact+json

Responses

200
OK [Resources-AwsSchedule](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/awsSchedules/search/findByUsernameAndJobTypeAndTenant

Up

findByUsernameAndJobTypeAndTenantAwsSchedule (findByUsernameAndJobTypeAndTenantAwsScheduleUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

jobType (optional)
Query Parameter — jobType
tenant (optional)

Query Parameter — tenant

username (optional)

Query Parameter — username

Return type

[Resources~AwsSchedule~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [Resources~AwsSchedule~](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/awsSchedules/{id}

Up

findOneAwsSchedule (findOneAwsScheduleUsingGET)

Path parameters

id (required)

Path Parameter — id

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

[Resource~AwsSchedule~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [Resource~AwsSchedule~](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /awsSchedules/schedule

Up

Get list of AwsSchedule and IDs for a user (getScheduleByUserNameUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

username (required)

Query Parameter — username

Return type

[AwsScheduleOutput](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [AwsScheduleOutput](#)

401

Unauthorized
403
Forbidden
404
Not Found

PATCH /rest/awsSchedules/{id}

Up

saveAwsSchedule (saveAwsScheduleUsingPATCH)

Path parameters

id (required)
Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [AwsSchedule](#) (required)
Body Parameter — body

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

[Resource~AwsSchedule~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Resource~AwsSchedule~](#)
204
No Content
401
Unauthorized
403
Forbidden

POST /rest/awsSchedules

Up

saveAwsSchedule (saveAwsScheduleUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [AwsSchedule](#) (required)
Body Parameter — body

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

[Resource~AwsSchedule~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Resource~AwsSchedule~](#)
201
Created
401
Unauthorized
403
Forbidden
404

PUT /rest/awsSchedules/{id}

Up

saveAwsSchedule (saveAwsScheduleUsingPUT)

Path parameters

id (required)
Path Parameter – id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [AwsSchedule](#) (required)
Body Parameter – body

Request headers**Cookie**

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

[Resource~AwsSchedule~](#)

Example data

Content-Type:

Produces

This API produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [Resource~AwsSchedule~](#)

201

Created

401

Unauthorized

403

Forbidden

404

Not Found

PUT /awsSchedules/{id}

Up

Update an AwsSchedule (updateUsingPUT)

Path parameters

id (required)
Path Parameter – id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

data [AwsSchedule](#) (required)
Body Parameter – Parameters:
 jobType,
 description,
 pause,
 cron or (startDate, interval, and intervalUnit),
 username,
 accessKeyId,
 password // AWS Secret Access Key

Request headers**Cookie**

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

AzureKeyEntity

PUT /rest/azureKeys/{id}/autorotate [Up](#)

Enable or disable the AzureKey to be auto rotated by this user's KEY_ROTATION schedule ([autorotateUsingPUT1](#))

Path parameters

id (required)
Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

data [AutoRotateRequestData](#) (required)
Body Parameter — data

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

POST /rest/azureKeys/create [Up](#)

Create a new AzureKey. Key material will be generated in Azure ([createUsingPOST1](#))

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

data [NativeKeyRequestData](#) (required)
Body Parameter — data

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /rest/azureKeys/{id}/deleteBackup [Up](#)

Delete the backup of an Azure key. This deletes the AzureKey in CCKM ([deleteBackupUsingPUT](#))

Path parameters

id (required)
Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /rest/azureKeys/{id}/delete [Up](#)

Delete an Azure key in Azure, including all versions. This does not delete AzureKey and key backup in CCKM ([deleteUsingPUT](#))

Path parameters

id (required)
Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [String](#)
201
Created
401
Unauthorized

GET /rest/azureKeys

Up

findAllAzureKey (findAllAzureKeyUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

page (optional)

Query Parameter — page

size (optional)

Query Parameter — size

sort (optional)

Query Parameter — sort

Return type

[Resources-AzureKey»](#)

Example data

Content-Type: application/json

```
{
  "_embedded": {
    "azureKeys": [
      {
        "autoRotate": true,
        "autoRotateDsm": true,
        "autoRotateHsm": true,
        "autoRotateKeyProvider": "DSM",
        "autoRotateKeyType": "AES256",
        "autoRotateOnExpire": true,
        "autoRotateUser": "string",
        "availableInKeyVault": true,
        "backup": {
          "algorithm": "string",
          "allTenant": true,
          "blob": "string",
          "blobUUID": "string",
          "cockId": "string",
          "certificate": "string",
          "createdAt": "2019-03-25T23:40:14.188Z",
          "createdBy": "string",
          "deleted": true,
          "description": "string",
          "hashType": "string",
          "id": {
            "counter": 0,
            "date": "2019-03-25T23:40:14.188Z",
            "machineIdentifier": 0,
            "processIdentifier": 0,
            "time": 0,
            "timeSecond": 0,
            "timestamp": 0
          },
          "keySource": "KMaas",
          "modifiedAt": "2019-03-25T23:40:14.188Z",
          "modifiedBy": "string",
          "nShieldKeyBlob": "string",
          "name": "string",
          "privateKey": {
            "hash": "string",
            "length": 0,
            "value": "string"
          },
          "protected": true,
          "provider": "DSM",
          "publicKey": {
            "hash": "string",
            "length": 0,
            "value": "string"
          },
          "secretKey": {
            "hash": "string",
            "length": 0,
            "value": "string"
          },
          "service": "NONE",
          "serviceKeyPage": {
            "content": [
              {
                "availableInKeyVault": true,
                "createdAt": "2019-03-25T23:40:14.188Z",
                "createdBy": "string",
                "id": {
                  "counter": 0,
                  "date": "2019-03-25T23:40:14.188Z",
                  "machineIdentifier": 0,
                  "processIdentifier": 0,
                  "time": 0,
                  "timeSecond": 0,
                  "timestamp": 0
                },
                "keyType": "AES256",
                "modifiedAt": "2019-03-25T23:40:14.188Z",
                "modifiedBy": "string",
                "name": "string",
```

```

"source": "KMaas",
"status": "CREATED",
"syncedAt": "2019-03-25T23:40:14.188Z",
"tenant": {
  "cloudService": "string",
  "createdAt": "2019-03-25T23:40:14.188Z",
  "createdBy": "string",
  "id": {
    "counter": 0,
    "date": "2019-03-25T23:40:14.188Z",
    "machineIdentifier": 0,
    "processIdentifier": 0,
    "time": 0,
    "timeSecond": 0,
    "timestamp": 0
  },
  "modifiedAt": "2019-03-25T23:40:14.188Z",
  "modifiedBy": "string",
  "name": "string"
},
"uploadedAt": "2019-03-25T23:40:14.188Z",
"uploadedBy": "string"
}
},
"first": true,
"last": true,
"number": 0,
"numberOfElements": 0,
"size": 0,
"sort": {},
"totalElements": 0,
"totalPages": 0
},
"sourceKeyTier": "string",
"tenant": {
  "cloudService": "string",
  "createdAt": "2019-03-25T23:40:14.188Z",
  "createdBy": "string",
  "id": {
    "counter": 0,
    "date": "2019-03-25T23:40:14.188Z",
    "machineIdentifier": 0,
    "processIdentifier": 0,
    "time": 0,
    "timeSecond": 0,
    "timestamp": 0
  },
  "modifiedAt": "2019-03-25T23:40:14.188Z",
  "modifiedBy": "string",
  "name": "string"
}
},
"backupAt": 0,
"createdAt": "2019-03-25T23:40:14.188Z",
"createdBy": "string",
"deleted": true,
"description": "string",
"id": {
  "counter": 0,
  "date": "2019-03-25T23:40:14.188Z",
  "machineIdentifier": 0,
  "processIdentifier": 0,
  "time": 0,
  "timeSecond": 0,
  "timestamp": 0
},
"keySoftDeletedInAzure": true,
"keyType": "AES256",
"keyVault": "string",
"keyVaultDisplayName": "string",
"keyVaultLocation": "string",
"keyVaultLocationDisplay": "string",
"kid": "string",
"modifiedAt": "2019-03-25T23:40:14.188Z",
"modifiedBy": "string",
"name": "string",
"originalKey": {
  "algorithm": "string",
  "allTenant": true,
  "blob": "string",
  "blobUUID": "string",
  "ckmId": "string",
  "certificate": "string",
  "createdAt": "2019-03-25T23:40:14.188Z",
  "createdBy": "string",
  "deleted": true,
  "description": "string",
  "hashType": "string",
  "id": {
    "counter": 0,
    "date": "2019-03-25T23:40:14.188Z",
    "machineIdentifier": 0,
    "processIdentifier": 0,
    "time": 0,
    "timeSecond": 0,
    "timestamp": 0
  },
},
"keySource": "KMaas",
"modifiedAt": "2019-03-25T23:40:14.188Z",
"modifiedBy": "string",
"nShieldKeyBlob": "string",
"name": "string",
"privateKey": {
  "hash": "string",
  "length": 0,
  "value": "string"
},
"protected": true,
"provider": "DSM",
"publicKey": {
  "hash": "string",
  "length": 0,
  "value": "string"
},
"secretKey": {
  "hash": "string",

```



```

    "length": 0,
    "value": "string"
  },
  "service": "NONE",
  "serviceKeyPage": {
    "content": [
      {
        "availableInKeyVault": true,
        "createdAt": "2019-03-25T23:40:14.188Z",
        "createdBy": "string",
        "id": {
          "counter": 0,
          "date": "2019-03-25T23:40:14.188Z",
          "machineIdentifier": 0,
          "processIdentifier": 0,
          "time": 0,
          "timeSecond": 0,
          "timestamp": 0
        },
        "keyType": "AES256",
        "modifiedAt": "2019-03-25T23:40:14.188Z",
        "modifiedBy": "string",
        "name": "string",
        "source": "KMaaS",
        "status": "CREATED",
        "syncedAt": "2019-03-25T23:40:14.188Z",
        "tenant": {
          "cloudService": "string",
          "createdAt": "2019-03-25T23:40:14.188Z",
          "createdBy": "string",
          "id": {
            "counter": 0,
            "date": "2019-03-25T23:40:14.188Z",
            "machineIdentifier": 0,
            "processIdentifier": 0,
            "time": 0,
            "timeSecond": 0,
            "timestamp": 0
          },
          "modifiedAt": "2019-03-25T23:40:14.189Z",
          "modifiedBy": "string",
          "name": "string"
        },
        "uploadedAt": "2019-03-25T23:40:14.189Z",
        "uploadedBy": "string"
      }
    ],
    "first": true,
    "last": true,
    "number": 0,
    "numberOfElements": 0,
    "size": 0,
    "sort": {},
    "totalElements": 0,
    "totalPages": 0
  },
  "sourceKeyTier": "string",
  "tenant": {
    "cloudService": "string",
    "createdAt": "2019-03-25T23:40:14.189Z",
    "createdBy": "string",
    "id": {
      "counter": 0,
      "date": "2019-03-25T23:40:14.189Z",
      "machineIdentifier": 0,
      "processIdentifier": 0,
      "time": 0,
      "timeSecond": 0,
      "timestamp": 0
    },
    "modifiedAt": "2019-03-25T23:40:14.189Z",
    "modifiedBy": "string",
    "name": "string"
  }
},
"softDeleteEnabled": true,
"source": "KMaaS",
"status": "CREATED",
"subscriptionId": "string",
"syncedAt": "2019-03-25T23:40:14.189Z",
"tenant": {
  "cloudService": "string",
  "createdAt": "2019-03-25T23:40:14.189Z",
  "createdBy": "string",
  "id": {
    "counter": 0,
    "date": "2019-03-25T23:40:14.189Z",
    "machineIdentifier": 0,
    "processIdentifier": 0,
    "time": 0,
    "timeSecond": 0,
    "timestamp": 0
  },
  "modifiedAt": "2019-03-25T23:40:14.189Z",
  "modifiedBy": "string",
  "name": "string"
},
"uploadedAt": "2019-03-25T23:40:14.189Z",
"uploadedBy": "string",
"uploadedFrom": "string",
"versions": [
  {
    "attributes": {
      "created": 0,
      "enabled": true,
      "exp": 0,
      "nbf": 0,
      "recoveryLevel": "string",
      "updated": 0
    },
    "crv": "string",
    "description": "string",
    "e": "string",
    "keyVault": "string",
    "key_ops": [
      "sign"
    ]
  }
]

```

```

    },
    "key_size": 0,
    "kid": "string",
    "kty": "string",
    "n": "string",
    "source": "KMaas",
    "tags": {
      "additionalProp1": "string",
      "additionalProp2": "string",
      "additionalProp3": "string"
    },
    "uploadedAt": "2019-03-25T23:40:14.189Z",
    "uploadedBy": "string",
    "uploadedFrom": "string",
    "version": "string"
  }
}
]
}
},
"links": {
  "additionalProp1": {
    "href": "string",
    "rel": "string",
    "templated": true
  },
  "additionalProp2": {
    "href": "string",
    "rel": "string",
    "templated": true
  },
  "additionalProp3": {
    "href": "string",
    "rel": "string",
    "templated": true
  }
}
}
}

```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json
- application/hal+json
- text/uri-list
- application/x-spring-data-compact+json

Responses

200

OK [Resources-AzureKey-](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/azureKeys/search/findByAutoRotateUserAndAutoRotate

Up

findByAutoRotateUserAndAutoRotateAzureKey (findByAutoRotateUserAndAutoRotateAzureKeyUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

autoRotate (optional)

Query Parameter — autoRotate

autoRotateUser (optional)

Query Parameter — autoRotateUser

Return type

[Resources-AzureKey-](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [Resources-AzureKey-](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/azureKeys/search/findByAutoRotateUserAndAutoRotateOnExpire

Up

findByAutoRotateUserAndAutoRotateOnExpireAzureKey (findByAutoRotateUserAndAutoRotateOnExpireAzureKeyUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

autoRotateOnExpire (optional)
Query Parameter — autoRotateOnExpire
autoRotateUser (optional)
Query Parameter — autoRotateUser

Return type
[Resources~AzureKey~](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resources~AzureKey~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/search/findByAutoRotateUser

findByAutoRotateUserAzureKey (findByAutoRotateUserAzureKeyUsingGET)

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters
autoRotateUser (optional)
Query Parameter — autoRotateUser

Return type
[Resources~AzureKey~](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resources~AzureKey~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/search/findById

findByIdAzureKey (findByIdAzureKeyUsingGET)

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters
_id (optional)
Query Parameter — _id

Return type
[Resource~Optional~AzureKey~>](#)

Example data

Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resource-Optional-AzureKey»»](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/search/findByKid Up

findByKidAzureKey (findByKidAzureKeyUsingGET)

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters
kid (optional)
Query Parameter — kid

Return type
[Resource-Optional-AzureKey»»](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resource-Optional-AzureKey»»](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/search/findByKidIgnoreCase Up

findByKidIgnoreCaseAzureKey (findByKidIgnoreCaseAzureKeyUsingGET)

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters
kid (optional)
Query Parameter — kid

Return type
[Resource-Optional-AzureKey»»](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resource-Optional-AzureKey»»](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/search/findByName Up

findByNameAzureKey (findByNameAzureKeyUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

name (optional)
Query Parameter — name

Return type

[Resources~AzureKey~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Resources~AzureKey~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/search/findByNameRegexAndKeyVaultRegex

Up

findByNameRegexAndKeyVaultRegexAzureKey (findByNameRegexAndKeyVaultRegexAzureKeyUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

keyVault (optional)
Query Parameter — keyVault
name (optional)
Query Parameter — name
page (optional)
Query Parameter — format: int32
size (optional)
Query Parameter — format: int32
sort (optional)
Query Parameter —

Return type

[Resource~Page~AzureKey~>>](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Resource~Page~AzureKey~>>](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/search/findByNameRegexOrKeyVaultRegexOrVersionsVersionRegexOrVersionsTagsUploadedFromAzureKey

Up

findByNameRegexOrKeyVaultRegexOrVersionsVersionRegexOrVersionsTagsUploadedFromAzureKey
(findByNameRegexOrKeyVaultRegexOrVersionsVersionRegexOrVersionsTagsUploadedFromAzureKeyUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant

Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

page (optional)
Query Parameter — format: int32
search (optional)
Query Parameter — search
size (optional)
Query Parameter — format: int32
sort (optional)
Query Parameter —

Return type
[Resource~Page~AzureKey~>](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resource~Page~AzureKey~>](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/findKeyByKid Up

Get AzureKey by Azure kid (findKeyByKidUsingGET)

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters
kid (required)
Query Parameter — kid

Return type
[AzureKeyResponse](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [AzureKeyResponse](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/{id} Up

findOneAzureKey (findOneAzureKeyUsingGET)

Path parameters
id (required)
Path Parameter — id

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[Resource~AzureKey~>](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [Resource~AzureKey~](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/azureKeys/getIdByKid

Up

Get AzureKey id by Azure kid ([getIdByKidUsingGET](#))

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

kid (required)

Query Parameter — kid

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [String](#)

401

Unauthorized

403

Forbidden

404

Not Found

PUT /rest/azureKeys/{id}/purge

Up

Purge a soft-deleted Azure key ([purgeUsingPUT](#))

Path parameters

id (required)

Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [String](#)

201

Created

401

Unauthorized

403

PUT /rest/azureKeys/{id}/recover

Up

Recover a soft-deleted Azure key (`recoverUsingPUT`)

Path parameters

id (required)
Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `*/*`

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /rest/azureKeys/{id}/restore

Up

Restore a deleted Azure key in Azure, including all versions (`restoreUsingPUT`)

Path parameters

id (required)
Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `*/*`

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/search/searchAndSortKeys

Up

searchAndSortKeysAzureKey (searchAndSortKeysAzureKeyUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

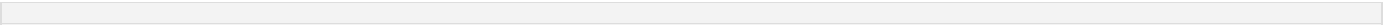
search (optional)
Query Parameter — search

Return type

[Resources~AzureKey-](#)

Example data

Content-Type:



Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Resources~AzureKey-](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeys/keys Up

Get AzureKeys by searching name, key vault, key version, exoFrom, expTo, and source key (searchAzureKeysUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

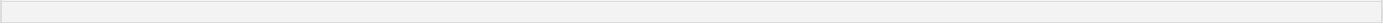
autoRotate (optional)
Query Parameter — autoRotate
backup (optional)
Query Parameter — backup
enabled (optional)
Query Parameter — enabled
inAzure (optional)
Query Parameter — inAzure
keyMaterialOrigin (optional)
Query Parameter — keyMaterialOrigin
keyVault (optional)
Query Parameter — keyVault
location (optional)
Query Parameter — location
name (optional)
Query Parameter — name
page (optional)
Query Parameter — format: int32
search (optional)
Query Parameter — search
size (optional)
Query Parameter — format: int32
sort (optional)
Query Parameter —
version (optional)
Query Parameter — version

Return type

[Page](#)

Example data

Content-Type:



Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Page](#)
401

Unauthorized
403
Forbidden
404
Not Found

POST /rest/azureKeys/synchronize

Up

Synchronize all AzureKeys that this user has proper permissions, download backups of Azure keys and store backups in CCKM (synchronizeAllKeysUsingPOST1)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [String](#)

201

Created

401

Unauthorized

403

Forbidden

404

Not Found

PUT /rest/azureKeys/{id}/update

Up

Update attributes of an Azure key (updateUsingPUT1)

Path parameters

id (required)

Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

data [KeyUpdateRequestData](#) (required)

Body Parameter — data

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [String](#)

201

Created

401

Unauthorized

403

Forbidden

404

Not Found

POST /rest/azureKeys/uploadKeyFileBLOB

Up

Upload an AzureKey from byok file that is stored in DSM as a source key (`uploadKeyFileBLOBUsingPOST`)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

data `KeyFileRequestData` (required)
Body Parameter — data

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `*/*`

Responses

200

OK [String](#)

201

Created

401

Unauthorized

403

Forbidden

404

Not Found

POST /rest/azureKeys/upload

Up

Upload a CCKM source key to Azure. This could create a new Azure key or a new version of an existing key. (`uploadUsingPOST`)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

data `KeyUploadRequestData` (required)
Body Parameter — data

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `*/*`

Responses

200

OK [String](#)

201

Created

401

Unauthorized

403

Forbidden

404

Not Found

AzureKeyVaultController

GET /rest/azureKeyVaults/{name}

Up

Gets the key vaults list by the name (`getKeyVaultUsingGET`)

Path parameters

name (required)
Path Parameter — Name of the keyVault to be searched

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[AzureKeyVault](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [AzureKeyVault](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeyVaults Up

Gets the key vaults which the user has access to ([listKeyVaultsUsingGET](#))

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
array[[AzureKeyVault](#)]

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureKeyVaults/getSubscriptionList Up

Gets all the subscriptions based on the current logged in user ([listSubscriptionsUsingGET](#))

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[AzureShortSubscriptionResponse](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200

OK [AzureShortSubscriptionResponse](#)
401
Unauthorized
403
Forbidden
404
Not Found

AzureScheduleEntity

POST /azureSchedules

Up

Create an AzureSchedule (createUsingPOST2)

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

schedule [AzureSchedule](#) (optional)
Body Parameter — Required parameters:
jobType,
pause,
cron or (startDate, interval, and intervalUnit),
username,
password

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

DELETE /azureSchedules/{id}

Up

Delete an AzureSchedule (deleteUsingDELETE)

Path parameters

id (required)
Path Parameter — id

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
204
No Content
401
Unauthorized
403
Forbidden

findAllAzureSchedule (findAllAzureScheduleUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

page (optional)

Query Parameter – page

size (optional)

Query Parameter – size

sort (optional)

Query Parameter – sort

Return type

[Resources-AzureSchedule-](#)

Example data

Content-Type: application/json

```
{
  "_embedded": {
    "azureSchedules": [
      {
        "applyOnExpiration": true,
        "createdAt": "2019-03-25T23:41:59.060Z",
        "createdBy": "string",
        "cron": "string",
        "description": "string",
        "fixedRateSchedule": true,
        "humanReadableCron": "string",
        "id": {
          "counter": 0,
          "date": "2019-03-25T23:41:59.060Z",
          "machineIdentifier": 0,
          "processIdentifier": 0,
          "time": 0,
          "timeSecond": 0,
          "timestamp": 0
        },
        "interval": 0,
        "intervalKeyExpiration": 0,
        "intervalUnit": "Hour",
        "intervalUnitKeyExpiration": "Hour",
        "jobType": "KEY_ROTATION",
        "modifiedAt": "2019-03-25T23:41:59.060Z",
        "modifiedBy": "string",
        "name": "string",
        "password": "string",
        "pause": true,
        "startDate": "2019-03-25T23:41:59.060Z",
        "tenant": {
          "cloudService": "string",
          "createdAt": "2019-03-25T23:41:59.060Z",
          "createdBy": "string",
          "id": {
            "counter": 0,
            "date": "2019-03-25T23:41:59.060Z",
            "machineIdentifier": 0,
            "processIdentifier": 0,
            "time": 0,
            "timeSecond": 0,
            "timestamp": 0
          },
          "modifiedAt": "2019-03-25T23:41:59.060Z",
          "modifiedBy": "string",
          "name": "string"
        },
        "username": "string"
      }
    ]
  },
  "_links": {
    "_additionalProp1": {
      "href": "string",
      "rel": "string",
      "templated": true
    },
    "additionalProp2": {
      "href": "string",
      "rel": "string",
      "templated": true
    },
    "additionalProp3": {
      "href": "string",
      "rel": "string",
      "templated": true
    }
  }
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json
- application/hal+json
- text/uri-list
- application/x-spring-data-compact+json

Responses
200
OK [Resources~AzureSchedule~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureSchedules/search/findByUsernameAndJobTypeAndTenant

findByUsernameAndJobTypeAndTenantAzureSchedule (findByUsernameAndJobTypeAndTenantAzureScheduleUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)

X-XSRF-TOKEN
Provide the token information [String](#)

x-sessionid
Provide the service session id [String](#)

x-tenant
Provide the cloud service name [String](#)

x-user
Provide the user name [String](#)

Query parameters

jobType (optional)
Query Parameter — jobType

tenant (optional)
Query Parameter — tenant

username (optional)
Query Parameter — username

Return type
[Resources~AzureSchedule~](#)

Example data
Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- * / *

Responses
200
OK [Resources~AzureSchedule~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/azureSchedules/{id}

findOneAzureSchedule (findOneAzureScheduleUsingGET)

Path parameters

id (required)
Path Parameter — id

Request headers

Cookie
Provide the JsessionId [String](#)

X-XSRF-TOKEN
Provide the token information [String](#)

x-sessionid
Provide the service session id [String](#)

x-tenant
Provide the cloud service name [String](#)

x-user
Provide the user name [String](#)

Return type
[Resource~AzureSchedule~](#)

Example data
Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- * / *

Responses
200
OK [Resource~AzureSchedule~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /azureSchedules/schedule

Get list of AzureSchedule and IDs for a user (getScheduleByUserNameUsingGET1)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

username (required)
Query Parameter — username

Return type

[AzureScheduleOutput](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [AzureScheduleOutput](#)
401
Unauthorized
403
Forbidden
404
Not Found

PATCH /rest/azureSchedules/{id} Up

saveAzureSchedule (saveAzureScheduleUsingPATCH)

Path parameters

id (required)
Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [AzureSchedule](#) (required)
Body Parameter — body

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

[Resource~AzureSchedule~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Resource~AzureSchedule~](#)
204
No Content
401
Unauthorized
403
Forbidden

POST /rest/azureSchedules Up

saveAzureSchedule (saveAzureScheduleUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [AzureSchedule](#) (required)
Body Parameter — body

Request headers

Cookie

Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[Resource~AzureSchedule~](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resource~AzureSchedule~](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /rest/azureSchedules/{id} [Up](#)

saveAzureSchedule (saveAzureScheduleUsingPUT)

Path parameters
id (required)
Path Parameter — id

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
body [AzureSchedule](#) (required)
Body Parameter — body

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[Resource~AzureSchedule~](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resource~AzureSchedule~](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /azureSchedules/{id} [Up](#)

Update an AzureSchedule (updateUsingPUT2)

Path parameters
id (required)
Path Parameter — id

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
data [AzureSchedule](#) (required)
Body Parameter — Parameters:
jobType,

description,
pause,
cron or (startDate, interval, and intervalUnit),
username,
password

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [String](#)

201

Created

401

Unauthorized

403

Forbidden

404

Not Found

CloudServiceInfoController

GET /csp

Up

Gets the cloud service information ([cspUsingGET](#))

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

array[String]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

401

Unauthorized

403

Forbidden

404

Not Found

KeyProducerController

GET /rest/keyProducers

Up

Gets the list of key producers ([keyProducersUsingGET](#))

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

array[[KeyProducer](#)]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK
401
Unauthorized
403
Forbidden
404
Not Found

KeyVaultController

POST /keyvaultkey

Up

Create source (DSM or nshield) key (createKeyUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

key [KeyVaultKey](#) (required)
Body Parameter — Source key parameters (service, cckmld, name, algorithm)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

page (optional)
Query Parameter — format: int32

sid (optional)
Query Parameter — sid

size (optional)
Query Parameter — format: int32

sort (optional)
Query Parameter —

Return type

[KeyVaultKey](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [KeyVaultKey](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

DELETE /keyvaultkey/{id}

Up

Delete source key with given id (deleteKeyUsingDELETE)

Path parameters

id (required)
Path Parameter — Internal id of source key

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

[KeyOperationResponse](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [KeyOperationResponse](#)
204
No Content
401
Unauthorized
403
Forbidden

GET /keyvaultkey/key/{id}

Get source key by id ([getKeyByIdUsingGET](#))

Path parameters

id (required)
Path Parameter — Internal id of source key

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

[KeyVaultKey](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [KeyVaultKey](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /keyvaultkey/keys

Get all source keys ([getKeyNotDeletedUsingGET](#))

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

page (optional)
Query Parameter — format: int32
provider (optional)
Query Parameter — Source key provider default: DSM
search (optional)
Query Parameter — search pattern
size (optional)
Query Parameter — format: int32
sort (optional)
Query Parameter —

Return type

[Page~KeyVaultKeyResponse~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Page~KeyVaultKeyResponse~](#)
401
Unauthorized

LicenseController

GET /license

Up

Gets the license information of the user (`getUsingGET`)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

[License](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [License](#)
401
Unauthorized
403
Forbidden
404
Not Found

ReportEntity

GET /rest/reports

Up

findAllReport (`findAllReportUsingGET`)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

page (optional)
Query Parameter — page
size (optional)
Query Parameter — size
sort (optional)
Query Parameter — sort

Return type

[Resources-Report-](#)

Example data

Content-Type: application/json

```
{
  "_embedded": {
    "reports": [
      {
        "createdAt": "2019-03-25T23:43:51.959Z",
        "createdBy": "string",
        "csp": "aws",
        "from": "2019-03-25T23:43:51.959Z",
        "id": {
          "counter": 0,
          "date": "2019-03-25T23:43:51.959Z",
          "machineIdentifier": 0,
          "processIdentifier": 0,
          "time": 0,
          "timeSecond": 0,
          "timestamp": 0
        },
        "modifiedAt": "2019-03-25T23:43:51.959Z",
        "modifiedBy": "string",
        "name": "string",
        "status": "REQUESTED",
        "tenant": {
          "cloudService": "string",
          "createdAt": "2019-03-25T23:43:51.959Z",
          "createdBy": "string",
          "id": {
            "counter": 0,
            "date": "2019-03-25T23:43:51.959Z",
```

```
        "machineIdentifier": 0,
        "processIdentifier": 0,
        "time": 0,
        "timeSecond": 0,
        "timestamp": 0
      },
      "modifiedAt": "2019-03-25T23:43:51.959Z",
      "modifiedBy": "string",
      "name": "string"
    },
    "to": "2019-03-25T23:43:51.959Z",
    "type": "KEY_360_RECONCILIATION"
  }
}
},
"_links": {
  "additionalProp1": {
    "href": "string",
    "rel": "string",
    "templated": true
  },
  "additionalProp2": {
    "href": "string",
    "rel": "string",
    "templated": true
  },
  "additionalProp3": {
    "href": "string",
    "rel": "string",
    "templated": true
  }
}
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json
- application/hal+json
- text/uri-list
- application/x-spring-data-compact+json

Responses

200

OK [Resources-Report-](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/reports/search/findByIdOrName

Up

findByIdOrNameReport (findByIdOrNameReportUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

counter (optional)

Query Parameter — format: int32

date (optional)

Query Parameter — format: date-time

machineIdentifier (optional)

Query Parameter — format: int32

param1 (optional)

Query Parameter — param1

processIdentifier (optional)

Query Parameter — format: int32

time (optional)

Query Parameter — format: int64

timeSecond (optional)

Query Parameter — format: int32

timestamp (optional)

Query Parameter — format: int32

Return type

[Resource-Optional-List-Report<>>>](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [Resource-Optional-List-Report<>>>](#)

401

Unauthorized

403

Forbidden

404

GET /rest/reports/search/findById

Up

findByIdReport (findByIdReportUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

counter (optional)
Query Parameter — format: int32
date (optional)
Query Parameter — format: date-time
machineIdentifier (optional)
Query Parameter — format: int32
processIdentifier (optional)
Query Parameter — format: int32
time (optional)
Query Parameter — format: int64
timeSecond (optional)
Query Parameter — format: int32
timestamp (optional)
Query Parameter — format: int32

Return type

[Resource<Optional<Report>>](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Resource<Optional<Report>>](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/reports/search/findByNameAndTenant

Up

findByNameAndTenantReport (findByNameAndTenantReportUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

name (optional)
Query Parameter — name
tenant (optional)
Query Parameter — tenant

Return type

[Resources<Report>](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Resources<Report>](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/reports/search/findByName

Up

findByNameReport (findByNameReportUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

name (optional)

Query Parameter — name

Return type

[Resource-Optional-Report»](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- * / *

Responses

200

OK [Resource-Optional-Report»](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/reports/{id}

Up

findOneReport (findOneReportUsingGET)

Path parameters

id (required)

Path Parameter — id

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

[Resource-Report-](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- * / *

Responses

200

OK [Resource-Report-](#)

401

Unauthorized

403

Forbidden

404

Not Found

SalesforceKeyEntity

POST /salesforceKeys/create

Up

Generate a new tenant secret in Salesforce, download and store the backup in CCKM (createKeyUsingPOST1)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

data [KeyUploadRequestData](#) (required)

Body Parameter — Required parameters:

keyDerivationMode - default to PBKDF2

type

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

DELETE /salesforceKeys/{id} [Up](#)

Delete the backup of a SalesforceKey from CCKM (deleteKeyUsingDELETE1)

Path parameters

id (required)
Path Parameter — id

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

deleteinfo (optional)
Query Parameter — deleteinfo

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
204
No Content
401
Unauthorized
403
Forbidden

PUT /salesforceKeys/{id}/destroy [Up](#)

Destroy a SalesforceKey, e.g. Salesforce tenant secret (destroyTenantSecretUsingPUT)

Path parameters

id (required)
Path Parameter — id

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

POST /salesforceKeys/synchronize

Up

Synchronize all SalesforceKeys, download backups of archived keys and store backups in CCKM ([downloadKeyUsingPOST](#))

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/salesforceKeys

Up

findAllSalesforceKey ([findAllSalesforceKeyUsingGET](#))

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

page (optional)
Query Parameter — page
size (optional)
Query Parameter — size
sort (optional)
Query Parameter — sort

Return type
[Resources-SalesforceKey-](#)

Example data
Content-Type: application/json

```
{
  "_embedded": {
    "salesforceKeys": [
      {
        "availableInKeyVault": true,
        "backup": {
          "algorithm": "string",
```

```

    "allTenant": true,
    "blob": "string",
    "blobUUID": "string",
    "cckmId": "string",
    "certificate": "string",
    "createdAt": "2019-03-25T23:45:07.473Z",
    "createdBy": "string",
    "deleted": true,
    "description": "string",
    "hashType": "string",
    "id": {
      "counter": 0,
      "date": "2019-03-25T23:45:07.473Z",
      "machineIdentifier": 0,
      "processIdentifier": 0,
      "time": 0,
      "timeSecond": 0,
      "timestamp": 0
    },
    "keySource": "KMaas",
    "modifiedAt": "2019-03-25T23:45:07.473Z",
    "modifiedBy": "string",
    "nShieldKeyBlob": "string",
    "name": "string",
    "privateKey": {
      "hash": "string",
      "length": 0,
      "value": "string"
    },
    "protected": true,
    "provider": "DSM",
    "publicKey": {
      "hash": "string",
      "length": 0,
      "value": "string"
    },
    "secretKey": {
      "hash": "string",
      "length": 0,
      "value": "string"
    },
    "service": "NONE",
    "serviceKeyPage": {
      "content": [
        {
          "availableInKeyVault": true,
          "createdAt": "2019-03-25T23:45:07.473Z",
          "createdBy": "string",
          "id": {
            "counter": 0,
            "date": "2019-03-25T23:45:07.473Z",
            "machineIdentifier": 0,
            "processIdentifier": 0,
            "time": 0,
            "timeSecond": 0,
            "timestamp": 0
          },
          "keyType": "AES256",
          "modifiedAt": "2019-03-25T23:45:07.473Z",
          "modifiedBy": "string",
          "name": "string",
          "source": "KMaas",
          "status": "CREATED",
          "syncedAt": "2019-03-25T23:45:07.473Z",
          "tenant": {
            "cloudService": "string",
            "createdAt": "2019-03-25T23:45:07.473Z",
            "createdBy": "string",
            "id": {
              "counter": 0,
              "date": "2019-03-25T23:45:07.473Z",
              "machineIdentifier": 0,
              "processIdentifier": 0,
              "time": 0,
              "timeSecond": 0,
              "timestamp": 0
            },
            "modifiedAt": "2019-03-25T23:45:07.473Z",
            "modifiedBy": "string",
            "name": "string"
          },
          "uploadedAt": "2019-03-25T23:45:07.473Z",
          "uploadedBy": "string"
        }
      ],
      "first": true,
      "last": true,
      "number": 0,
      "numberOfElements": 0,
      "size": 0,
      "sort": {},
      "totalElements": 0,
      "totalPages": 0
    },
    "sourceKeyTier": "string",
    "tenant": {
      "cloudService": "string",
      "createdAt": "2019-03-25T23:45:07.473Z",
      "createdBy": "string",
      "id": {
        "counter": 0,
        "date": "2019-03-25T23:45:07.473Z",
        "machineIdentifier": 0,
        "processIdentifier": 0,
        "time": 0,
        "timeSecond": 0,
        "timestamp": 0
      },
      "modifiedAt": "2019-03-25T23:45:07.473Z",
      "modifiedBy": "string",
      "name": "string"
    }
  },
  "createdAt": "2019-03-25T23:45:07.473Z",
  "createdBy": "string",
  "description": "string",

```

```

"displayOrder": 0,
"id": {
  "counter": 0,
  "date": "2019-03-25T23:45:07.473Z",
  "machineIdentifier": 0,
  "processIdentifier": 0,
  "time": 0,
  "timeSecond": 0,
  "timestamp": 0
},
"keyDerivationMode": "string",
"keyType": "AES256",
"modifiedAt": "2019-03-25T23:45:07.473Z",
"modifiedBy": "string",
"name": "string",
"originalKey": {
  "algorithm": "string",
  "allTenant": true,
  "blob": "string",
  "blobUUID": "string",
  "cckmId": "string",
  "certificate": "string",
  "createdAt": "2019-03-25T23:45:07.473Z",
  "createdBy": "string",
  "deleted": true,
  "description": "string",
  "hashType": "string",
  "id": {
    "counter": 0,
    "date": "2019-03-25T23:45:07.473Z",
    "machineIdentifier": 0,
    "processIdentifier": 0,
    "time": 0,
    "timeSecond": 0,
    "timestamp": 0
  },
  "keySource": "KMaaS",
  "modifiedAt": "2019-03-25T23:45:07.473Z",
  "modifiedBy": "string",
  "nShieldKeyBlob": "string",
  "name": "string",
  "privateKey": {
    "hash": "string",
    "length": 0,
    "value": "string"
  },
  "protected": true,
  "provider": "DSM",
  "publicKey": {
    "hash": "string",
    "length": 0,
    "value": "string"
  },
  "secretKey": {
    "hash": "string",
    "length": 0,
    "value": "string"
  },
  "service": "NONE",
  "serviceKeyPage": {
    "content": [
      {
        "availableInKeyVault": true,
        "createdAt": "2019-03-25T23:45:07.473Z",
        "createdBy": "string",
        "id": {
          "counter": 0,
          "date": "2019-03-25T23:45:07.473Z",
          "machineIdentifier": 0,
          "processIdentifier": 0,
          "time": 0,
          "timeSecond": 0,
          "timestamp": 0
        },
        "keyType": "AES256",
        "modifiedAt": "2019-03-25T23:45:07.473Z",
        "modifiedBy": "string",
        "name": "string",
        "source": "KMaaS",
        "status": "CREATED",
        "syncedAt": "2019-03-25T23:45:07.473Z",
        "tenant": {
          "cloudService": "string",
          "createdAt": "2019-03-25T23:45:07.473Z",
          "createdBy": "string",
          "id": {
            "counter": 0,
            "date": "2019-03-25T23:45:07.473Z",
            "machineIdentifier": 0,
            "processIdentifier": 0,
            "time": 0,
            "timeSecond": 0,
            "timestamp": 0
          },
          "modifiedAt": "2019-03-25T23:45:07.473Z",
          "modifiedBy": "string",
          "name": "string"
        },
        "uploadedAt": "2019-03-25T23:45:07.473Z",
        "uploadedBy": "string"
      }
    ],
    "first": true,
    "last": true,
    "number": 0,
    "numberOfElements": 0,
    "size": 0,
    "sort": {},
    "totalElements": 0,
    "totalPages": 0
  },
  "sourceKeyTier": "string",
  "tenant": {
    "cloudService": "string",
    "createdAt": "2019-03-25T23:45:07.473Z",
    "createdBy": "string",

```

```

        "id": {
            "counter": 0,
            "date": "2019-03-25T23:45:07.473Z",
            "machineIdentifier": 0,
            "processIdentifier": 0,
            "time": 0,
            "timeSecond": 0,
            "timestamp": 0
        },
        "modifiedAt": "2019-03-25T23:45:07.473Z",
        "modifiedBy": "string",
        "name": "string"
    }
},
"source": "KMaaS",
"status": "CREATED",
"syncedAt": "2019-03-25T23:45:07.473Z",
"tenant": {
    "cloudService": "string",
    "createdAt": "2019-03-25T23:45:07.473Z",
    "createdBy": "string",
    "id": {
        "counter": 0,
        "date": "2019-03-25T23:45:07.473Z",
        "machineIdentifier": 0,
        "processIdentifier": 0,
        "time": 0,
        "timeSecond": 0,
        "timestamp": 0
    },
    "modifiedAt": "2019-03-25T23:45:07.473Z",
    "modifiedBy": "string",
    "name": "string"
},
"tenantSecret": {
    "CreatedById": "string",
    "Description": "string",
    "Id": "string",
    "IsDeleted": true,
    "KeyDerivationMode": "string",
    "LastModifiedById": "string",
    "LastModifiedByUsername": "string",
    "LastModifiedDate": "2019-03-25T23:45:07.473Z",
    "SecretValueCertificate": "string",
    "SecretValueHash": "string",
    "SystemModstamp": "2019-03-25T23:45:07.473Z",
    "Type": "string",
    "attributes": {
        "type": "string",
        "url": "string"
    },
    "createdByUsername": "string",
    "createdDate": "2019-03-25T23:45:07.473Z",
    "keyvaultName": "string",
    "source": "string",
    "status": "ACTIVE",
    "version": 0
},
"type": "Data",
"uploadedAt": "2019-03-25T23:45:07.473Z",
"uploadedBy": "string",
"uploadedFrom": "string",
"uploadedToSalesforceBy": "string"
}
}
},
"links": {
    "additionalProp1": {
        "href": "string",
        "rel": "string",
        "templated": true
    },
    "additionalProp2": {
        "href": "string",
        "rel": "string",
        "templated": true
    },
    "additionalProp3": {
        "href": "string",
        "rel": "string",
        "templated": true
    }
}
}
}

```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json
- application/hal+json
- text/uri-list
- application/x-spring-data-compact+json

Responses

200

OK [Resources-SalesforceKey-](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /rest/salesforceKeys/search/findByName

findByNameSalesforceKey (findByNameSalesforceKeyUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

name (optional)
Query Parameter — name

Return type
[Resource<Optional<SalesforceKey>>](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resource<Optional<SalesforceKey>>](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/salesforceKeys/search/findByStatus

Up

findByStatusSalesforceKey (findByStatusSalesforceKeyUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

keyStatus (optional)
Query Parameter — keyStatus

Return type
[Resources<SalesforceKey>](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resources<SalesforceKey>](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/salesforceKeys/search/findByTenantSecret_Id

Up

findByTenantSecret_IdSalesforceKey (findByTenantSecretIdSalesforceKeyUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

Id (optional)
Query Parameter — Id

Return type
[Resource<SalesforceKey>](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

• */*

Responses

200
OK [Resource~SalesforceKey~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/salesforceKeys/search/findByTenantSecret_StatusAndType

Up

findByTenantSecret_StatusAndTypeSalesforceKey (findByTenantSecretStatusAndTypeSalesforceKeyUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

status (optional)
Query Parameter — status

type (optional)
Query Parameter — type

Return type

[Resources~SalesforceKey~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

• */*

Responses

200
OK [Resources~SalesforceKey~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/salesforceKeys/search/findByType

Up

findByTypeSalesforceKey (findByTypeSalesforceKeyUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

page (optional)
Query Parameter — format: int32

size (optional)
Query Parameter — format: int32

sort (optional)
Query Parameter —

type (optional)
Query Parameter — type

Return type

[Resource~Page~SalesforceKey~>>](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

• */*

Responses

200
OK [Resource~Page~SalesforceKey~>>](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/salesforceKeys/{id}

Up

findOneSalesforceKey (findOneSalesforceKeyUsingGET)

Path parameters

id (required)
Path Parameter — id

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

[Resource~SalesforceKey~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [Resource~SalesforceKey~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /salesforceKeys/{id}

Up

Get a SalesforceKey from CCKM (getKeyUsingGET)

Path parameters

id (required)
Path Parameter — id

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [String](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /salesforceKeys/keys

Up

Get Salesforce keys by searching search, type, name, version, status, generatedBy, etc. (getKeysUsingGET1)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

backup (optional)

Query Parameter – backup

data (optional)
Query Parameter – data

generatedBy (optional)
Query Parameter – generatedBy

keyDerivation (optional)
Query Parameter – keyDerivation

keyId (optional)
Query Parameter – keyId

keyMaterialOrigin (optional)
Query Parameter – keyMaterialOrigin

name (optional)
Query Parameter – name

page (optional)
Query Parameter – format: int32

search (optional)
Query Parameter – search

size (optional)
Query Parameter – format: int32

sort (optional)
Query Parameter –

status (optional)
Query Parameter – status

type (required)
Query Parameter – type

uploaded (optional)
Query Parameter – uploaded

Return type
[Page](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Page](#)
401
Unauthorized
403
Forbidden
404
Not Found

PUT /salesforceKeys/{id}/import [Up](#)

Restore a destroyed SalesforceKey from the backup stored in CCKM (**importKeyUsingPUT**)

Path parameters

id (required)
Path Parameter – id

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /salesforceKeys/{id}

Up

Update a SalesforceKey ([modifyKeyUsingPUT](#))

Path parameters

id (required)
Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- [application/json](#)

Request body

data [KeyModifyRequestData](#) (required)
Body Parameter — data

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- [*/*](#)

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

POST /salesforceKeys/upload

Up

Upload a CCKM source key to Salesforce as a new tenant secret ([uploadKeyUsingPOST](#))

Consumes

This API call consumes the following media types via the Content-Type request header:

- [application/json](#)

Request body

data [KeyUploadRequestData](#) (required)
Body Parameter — Required parameters:
keyVaultKeyId - ID of the source key to be uploaded,
certificateLabel - the certificate for importing Salesforce tenant secret,
keyDerivationMode - default to PBKDF2
type

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- [*/*](#)

Responses

200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

SalesforceScheduleEntity

POST /salesforceSchedules

Up

Create a SalesforceSchedule (createUsingPOST3)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

schedule [SalesforceSchedule](#) (optional)

Body Parameter – Required parameters:

jobType (KEY_ROTATION or KEY_SYNCHRONIZATION),
pause,
cron or (startDate, interval, and intervalUnit),
username,
password,
certificate if jobType is KEY_ROTATION

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [String](#)

201

Created

401

Unauthorized

403

Forbidden

404

Not Found

DELETE /salesforceSchedules/{id}

Up

Delete a SalesforceSchedule (deleteUsingDELETE1)

Path parameters

id (required)

Path Parameter – id

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [String](#)

204

No Content

401

Unauthorized

403

Forbidden

GET /rest/salesforceSchedules

Up

findAllSalesforceSchedule (findAllSalesforceScheduleUsingGET)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

page (optional)

Query Parameter — page

size (optional)

Query Parameter — size

sort (optional)

Query Parameter — sort

Return type

[Resources-SalesforceSchedule](#)

Example data

Content-Type: application/json

```
{
  "_embedded": {
    "salesforceSchedules": [
      {
        "applyOnExpiration": true,
        "certificate": "string",
        "createdAt": "2019-03-25T23:46:36.325Z",
        "createdBy": "string",
        "cron": "string",
        "description": "string",
        "fixedRateSchedule": true,
        "humanReadableCron": "string",
        "id": {
          "counter": 0,
          "date": "2019-03-25T23:46:36.325Z",
          "machineIdentifier": 0,
          "processIdentifier": 0,
          "time": 0,
          "timeSecond": 0,
          "timestamp": 0
        },
        "interval": 0,
        "intervalKeyExpiration": 0,
        "intervalUnit": "Hour",
        "intervalUnitKeyExpiration": "Hour",
        "jobType": "KEY_ROTATION",
        "keyDerivationMode": "string",
        "modifiedAt": "2019-03-25T23:46:36.325Z",
        "modifiedBy": "string",
        "name": "string",
        "password": "string",
        "pause": true,
        "startDate": "2019-03-25T23:46:36.325Z",
        "tenant": {
          "cloudService": "string",
          "createdAt": "2019-03-25T23:46:36.325Z",
          "createdBy": "string",
          "id": {
            "counter": 0,
            "date": "2019-03-25T23:46:36.325Z",
            "machineIdentifier": 0,
            "processIdentifier": 0,
            "time": 0,
            "timeSecond": 0,
            "timestamp": 0
          },
          "modifiedAt": "2019-03-25T23:46:36.325Z",
          "modifiedBy": "string",
          "name": "string"
        },
        "types": [
          "Data"
        ],
        "username": "string"
      }
    ]
  },
  "_links": {
    "additionalProp1": {
      "href": "string",
      "rel": "string",
      "templated": true
    },
    "additionalProp2": {
      "href": "string",
      "rel": "string",
      "templated": true
    },
    "additionalProp3": {
      "href": "string",
      "rel": "string",
      "templated": true
    }
  }
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json
- application/hal+json
- text/uri-list
- application/x-spring-data-compact+json

Responses

200

OK [Resources-SalesforceSchedule~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/salesforceSchedules/search/findByUsernameAndJobTypeAndTenant

Up

findByUsernameAndJobTypeAndTenantSalesforceSchedule (findByUsernameAndJobTypeAndTenantSalesforceScheduleUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

jobType (optional)
Query Parameter — jobType
tenant (optional)
Query Parameter — tenant
username (optional)
Query Parameter — username

Return type

[Resources-SalesforceSchedule~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

• */*

Responses

200
OK [Resources-SalesforceSchedule~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/salesforceSchedules/search/findByUsername

Up

findByUsernameSalesforceSchedule (findByUsernameSalesforceScheduleUsingGET)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

username (optional)
Query Parameter — username

Return type

[Resources-SalesforceSchedule~](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

• */*

Responses

200
[Resources-SalesforceSchedule~](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/salesforceSchedules/{id}

Up

findOneSalesforceSchedule (findOneSalesforceScheduleUsingGET)

Path parameters

id (required)

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Return type

[Resource<SalesforceSchedule>](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK [Resource<SalesforceSchedule>](#)

401

Unauthorized

403

Forbidden

404

Not Found

GET /salesforceSchedules/schedule

Up

Get list of SalesforceSchedule and IDs for a user ([getSchedulesByUserNameUsingGET](#))

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)

x-user

Provide the user name [String](#)

Query parameters

username (required)

Query Parameter — name

Return type

array[[SalesforceScheduleOutput](#)]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

401

Unauthorized

403

Forbidden

404

Not Found

PATCH /rest/salesforceSchedules/{id}

Up

saveSalesforceSchedule ([saveSalesforceScheduleUsingPATCH](#))

Path parameters

id (required)

Path Parameter — id

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [SalesforceSchedule](#) (required)

Body Parameter — body

Request headers

Cookie

Provide the JsessionId [String](#)

X-XSRF-TOKEN

Provide the token information [String](#)

x-sessionid

Provide the service session id [String](#)

x-tenant

Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[Resource~SalesforceSchedule~](#)

Example data
Content-Type:

Produces
This API produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resource~SalesforceSchedule~](#)
204
No Content
401
Unauthorized
403
Forbidden

POST /rest/salesforceSchedules [Up](#)

saveSalesforceSchedule (saveSalesforceScheduleUsingPOST)

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
body [SalesforceSchedule](#) (required)
Body Parameter — body

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[Resource~SalesforceSchedule~](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resource~SalesforceSchedule~](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /rest/salesforceSchedules/{id} [Up](#)

saveSalesforceSchedule (saveSalesforceScheduleUsingPUT)

Path parameters
id (required)
Path Parameter — id

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
body [SalesforceSchedule](#) (required)
Body Parameter — body

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[Resource<SalesforceSchedule>](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [Resource<SalesforceSchedule>](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /salesforceSchedules/{id}/toggleAutoRotate Up

Specify what data type will be rotated by this KEY_ROTATION SalesforceSchedule (toggleAutoRotateForTypeUsingPUT)

Path parameters
id (required)
Path Parameter — id

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
data ToggleAutoRotateData (required)
Body Parameter — Required parameters:
enable,
type

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

PUT /salesforceSchedules/{id} Up

Update a SalesforceSchedule (updateUsingPUT3)

Path parameters
id (required)
Path Parameter — id

Consumes
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body
scheduleModifications SalesforceSchedule (required)
Body Parameter — Parameters:
jobType (KEY_ROTATION or KEY_SYNCHRONIZATION),
pause,
cron or (startDate, interval, and intervalUnit),
username,
password,
certificate if jobType is KEY_ROTATION

Request headers
Cookie

Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
201
Created
401
Unauthorized
403
Forbidden
404
Not Found

SessionDataController

GET /rest/serviceSessions/myAzureCurrentDirectory [Up](#)

Gets current session Azure tenant ID that the user has access to ([getCurrentDirectoryUsingGET](#))

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
String

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [String](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/serviceSessions/search/findByCloudService [Up](#)

List the cloud service sessions by cloud service name ([getSessionDataByCloudServiceUsingGET](#))

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters
cloudService (required)
Query Parameter – cloudService

Return type
array[[SessionData](#)]

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/serviceSessions/search/findByUser

Up

List the cloud service sessions by username (`getSessionDataByNameUsingGET`)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

user (required)
Query Parameter — user

Return type

array[[SessionData](#)]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/serviceSessions/search/findByTenant

Up

List the cloud service sessions by tenant ID (`getSessionDataByTenantUsingGET`)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

tenant (required)
Query Parameter — tenant

Return type

array[[SessionData](#)]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/serviceSessions/{id}

Up

Get the cloud service session by id (`getSessionDataUsingGET`)

Path parameters

id (required)
Path Parameter — id

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
[SessionData](#)

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK [SessionData](#)
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/serviceSessions/myAzureTenantIds [Up](#)

Gets Azure tenant IDs that the user has access to ([listMyTenantsUsingGET](#))

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
array[String]

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK
401
Unauthorized
403
Forbidden
404
Not Found

GET /rest/serviceSessions [Up](#)

List the cloud service sessions the user has currently logged in ([listSessionDataUsingGET](#))

Request headers
Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type
array[[SessionData](#)]

Example data
Content-Type:

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses
200
OK
401
Unauthorized
403
Forbidden

SystemInfoController

GET /systemInfo

Up

Gets the version and relate information of the system (`getSystemInfoUsingGET`)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Return type

[SystemInfo](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [SystemInfo](#)
401
Unauthorized
403
Forbidden
404
Not Found

UserInfoController

GET /me

Up

This is used to get the user related information (`userUsingGET`)

Request headers

Cookie
Provide the JsessionId [String](#)
X-XSRF-TOKEN
Provide the token information [String](#)
x-sessionid
Provide the service session id [String](#)
x-tenant
Provide the cloud service name [String](#)
x-user
Provide the user name [String](#)

Query parameters

name (optional)
Query Parameter —

Return type

[PrincipalAndUser](#)

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200
OK [PrincipalAndUser](#)
401
Unauthorized
403
Forbidden
404
Not Found

Models

[[Jump to Methods](#)]

Table of Contents

1. [AccessPolicy](#) - [AccessPolicy](#)
2. [Action](#) - [Action](#)
3. [Attribute](#) - [Attribute](#)
4. [Authentication](#) - [Authentication](#)
5. [AutoRotateRequestData](#) - [AutoRotateRequestData](#)
6. [AwsKey](#) - [AwsKey](#)
7. [AwsKeyAlias](#) - [AwsKeyAlias](#)
8. [AwsKeyResponse](#) - [AwsKeyResponse](#)
9. [AwsPolicy](#) - [AwsPolicy](#)
10. [AwsSchedule](#) - [AwsSchedule](#)
11. [AwsScheduleOutput](#) - [AwsScheduleOutput](#)

12. [AzureDirectory - AzureDirectory](#)
13. [AzureKey - AzureKey](#)
14. [AzureKeyAttributes - AzureKeyAttributes](#)
15. [AzureKeyResponse - AzureKeyResponse](#)
16. [AzureKeyVault - AzureKeyVault](#)
17. [AzureKeyVersion - AzureKeyVersion](#)
18. [AzureSchedule - AzureSchedule](#)
19. [AzureScheduleOutput - AzureScheduleOutput](#)
20. [AzureShortSubscriptionResponse - AzureShortSubscriptionResponse](#)
21. [Condition - Condition](#)
22. [DBObject - DBObject](#)
23. [DeleteRequestBody - DeleteRequestBody](#)
24. [EmbeddedCollection«AwsKey» - EmbeddedCollection«AwsKey»](#)
25. [EmbeddedCollection«AwsSchedule» - EmbeddedCollection«AwsSchedule»](#)
26. [EmbeddedCollection«AzureKey» - EmbeddedCollection«AzureKey»](#)
27. [EmbeddedCollection«AzureSchedule» - EmbeddedCollection«AzureSchedule»](#)
28. [EmbeddedCollection«Report» - EmbeddedCollection«Report»](#)
29. [EmbeddedCollection«SalesforceKey» - EmbeddedCollection«SalesforceKey»](#)
30. [EmbeddedCollection«SalesforceSchedule» - EmbeddedCollection«SalesforceSchedule»](#)
31. [GrantConstraints - GrantConstraints](#)
32. [GrantListEntry - GrantListEntry](#)
33. [GrantedAuthority - GrantedAuthority](#)
34. [ImportKeyRequestData - ImportKeyRequestData](#)
35. [Key - Key](#)
36. [KeyFileRequestData - KeyFileRequestData](#)
37. [KeyMetadata - KeyMetadata](#)
38. [KeyModifyRequestData - KeyModifyRequestData](#)
39. [KeyOperationResponse - KeyOperationResponse](#)
40. [KeyProducer - KeyProducer](#)
41. [KeyUpdateRequestData - KeyUpdateRequestData](#)
42. [KeyUploadRequestData - KeyUploadRequestData](#)
43. [KeyVaultKey - KeyVaultKey](#)
44. [KeyVaultKeyResponse - KeyVaultKeyResponse](#)
45. [KeyVaultProperties - KeyVaultProperties](#)
46. [License - License](#)
47. [Link - Link](#)
48. [NativeKeyRequestData - NativeKeyRequestData](#)
49. [OAuth2AccessToken - OAuth2AccessToken](#)
50. [OAuth2RefreshToken - OAuth2RefreshToken](#)
51. [ObjectId - ObjectId](#)
52. [Optional«AzureKey» - Optional«AzureKey»](#)
53. [Optional«List«Report»» - Optional«List«Report»»](#)
54. [Optional«Report» - Optional«Report»](#)
55. [Optional«SalesforceKey» - Optional«SalesforceKey»](#)
56. [Page - Page](#)
57. [Page«AzureKey» - Page«AzureKey»](#)
58. [Page«DBObject» - Page«DBObject»](#)
59. [Page«KeyVaultKeyResponse» - Page«KeyVaultKeyResponse»](#)
60. [Page«SalesforceKey» - Page«SalesforceKey»](#)
61. [Page«ServiceKey» - Page«ServiceKey»](#)
62. [Permission - Permission](#)
63. [PolicyPrincipal - PolicyPrincipal](#)
64. [PolicyStatement - PolicyStatement](#)
65. [PrincipalAndUser - PrincipalAndUser](#)
66. [Report - Report](#)
67. [Resource - Resource](#)
68. [ResourceSupport - ResourceSupport](#)
69. [Resources«AwsKey» - Resources«AwsKey»](#)
70. [Resources«AwsSchedule» - Resources«AwsSchedule»](#)
71. [Resources«AzureKey» - Resources«AzureKey»](#)
72. [Resources«AzureSchedule» - Resources«AzureSchedule»](#)
73. [Resources«Report» - Resources«Report»](#)
74. [Resources«SalesforceKey» - Resources«SalesforceKey»](#)
75. [Resources«SalesforceSchedule» - Resources«SalesforceSchedule»](#)
76. [Resource«AwsKey» - Resource«AwsKey»](#)
77. [Resource«AwsSchedule» - Resource«AwsSchedule»](#)
78. [Resource«AzureKey» - Resource«AzureKey»](#)
79. [Resource«AzureSchedule» - Resource«AzureSchedule»](#)
80. [Resource«Optional«AzureKey»» - Resource«Optional«AzureKey»»](#)
81. [Resource«Optional«List«Report»»» - Resource«Optional«List«Report»»»](#)
82. [Resource«Optional«Report»» - Resource«Optional«Report»»](#)
83. [Resource«Optional«SalesforceKey»» - Resource«Optional«SalesforceKey»»](#)
84. [Resource«Page«AzureKey»» - Resource«Page«AzureKey»»](#)
85. [Resource«Page«SalesforceKey»» - Resource«Page«SalesforceKey»»](#)
86. [Resource«Report» - Resource«Report»](#)
87. [Resource«SalesforceKey» - Resource«SalesforceKey»](#)
88. [Resource«SalesforceSchedule» - Resource«SalesforceSchedule»](#)
89. [RotateKeyRequestData - RotateKeyRequestData](#)
90. [SalesforceKey - SalesforceKey](#)
91. [SalesforceSchedule - SalesforceSchedule](#)
92. [SalesforceScheduleOutput - SalesforceScheduleOutput](#)
93. [SalesforceTenantSecret - SalesforceTenantSecret](#)
94. [ServiceKey - ServiceKey](#)
95. [SessionData - SessionData](#)
96. [Sku - Sku](#)
97. [Sort - Sort](#)
98. [SystemInfo - SystemInfo](#)
99. [Tag - Tag](#)
100. [Tenant - Tenant](#)
101. [ToggleAutoRotateData - ToggleAutoRotateData](#)
102. [UpdateKeyPolicyRequestData - UpdateKeyPolicyRequestData](#)
103. [UpdateKeyRequestData - UpdateKeyRequestData](#)
104. [UploadKeyRequestData - UploadKeyRequestData](#)
105. [User - User](#)
106. [VerifyAliasData - VerifyAliasData](#)

[AccessPolicy - AccessPolicy](#)

[Up](#)

objectId (optional)
[String](#)
permissions (optional)
[Permission](#)
tenantId (optional)

[String](#)

Action - Action

[Up](#)

actionName (optional)
[String](#)

Attribute - Attribute

[Up](#)

type (optional)
[String](#)
url (optional)
[String](#)

Authentication - Authentication

[Up](#)

authenticated (optional)
[Boolean](#)
authorities (optional)
[array\[GrantedAuthority\]](#)
credentials (optional)
[Object](#)
details (optional)
[Object](#)
name (optional)
[String](#)
principal (optional)
[Object](#)

AutoRotateRequestData - AutoRotateRequestData

[Up](#)

autoRotateOnExpire (optional)
[Boolean](#)
dsm (optional)
[Boolean](#)
enable (optional)
[Boolean](#)
hsm (optional)
[Boolean](#)
keyProvider (optional)
[String](#)
Enum:
DSM
DB
nShield
keyType (optional)
[String](#)
Enum:
AES256
RSA2048
RSA4096
RSA3072
ECP256
ECP384
ECP521
ECSECP256K1

AwsKey - AwsKey

[Up](#)

aliases (optional)
[array\[AwsKeyAlias\]](#)
autoRotate (optional)
[Boolean](#)
autoRotateDisableEncrypt (optional)
[Boolean](#)
autoRotateKeyProvider (optional)
[String](#)
Enum:
DSM
DB
nShield
autoRotateOnExpire (optional)
[Boolean](#)
autoRotateUser (optional)
[String](#)
availableInKeyVault (optional)
[Boolean](#)
backup (optional)
[KeyVaultKey](#)
createdAt (optional)
[Date](#) format: date-time
createdBy (optional)
[String](#)
grants (optional)
[array\[GrantListEntry\]](#)
id (optional)
[ObjectId](#)
keyMetadata (optional)
[KeyMetadata](#)
keyType (optional)
[String](#)
Enum:

AES256
RSA2048
RSA4096
RSA3072
ECP256
ECP384
ECP521
ECSECP256K1

modifiedAt (optional)
[Date](#) format: date-time

modifiedBy (optional)
[String](#)

name (optional)
[String](#)

originalKey (optional)
[KeyVaultKey](#)

policy (optional)
[AwsPolicy](#)

region (optional)
[String](#)

regionLabel (optional)
[String](#)

rotationChainId (optional)
[String](#)

rotationStatus (optional)
[String](#)

Enum:
ARCHIVED
ACTIVE

source (optional)
[String](#)

Enum:
KMaaS
Salesforce
Azure

status (optional)
[String](#)

Enum:
CREATED
UPLOADED
EXPORTED
ALREADY_DESTROYED

syncedAt (optional)
[Date](#) format: date-time

tags (optional)
[array\[Tag\]](#)

tenant (optional)
[Tenant](#)

uploadedAt (optional)
[Date](#) format: date-time

uploadedBy (optional)
[String](#)

uploadedFrom (optional)
[String](#)

AwsKeyAlias - AwsKeyAlias

[Up](#)

arn (optional)
[String](#)

name (optional)
[String](#)

AwsKeyResponse - AwsKeyResponse

[Up](#)

awsKey (optional)
[AwsKey](#)

id (optional)
[String](#)

sourceKeyId (optional)
[String](#)

AwsPolicy - AwsPolicy

[Up](#)

id (optional)
[String](#)

statements (optional)
[array\[PolicyStatement\]](#)

version (optional)
[String](#)

AwsSchedule - AwsSchedule

[Up](#)

accessKeyId (optional)
[String](#)

applyOnExpiration (optional)
[Boolean](#)

createdAt (optional)
[Date](#) format: date-time

createdBy (optional)
[String](#)

cron (optional)
[String](#)

description (optional)
[String](#)

fixedRateSchedule (optional)
[Boolean](#)

humanReadableCron (optional)
[String](#)

id (optional)
[ObjectId](#)

interval (optional)
[Integer](#) format: int32

intervalKeyExpiration (optional)
[Integer](#) format: int32

intervalUnit (optional)
[String](#)
Enum:
Hour
Day
Week
Month
Year
None

intervalUnitKeyExpiration (optional)
[String](#)
Enum:
Hour
Day
Week
Month
Year
None

jobType (optional)
[String](#)
Enum:
KEY_ROTATION
KEY_SYNCHRONIZATION
KEY_EXPIRATION

modifiedAt (optional)
[Date](#) format: date-time

modifiedBy (optional)
[String](#)

name (optional)
[String](#)

password (optional)
[String](#)

pause (optional)
[Boolean](#)

startDate (optional)
[Date](#) format: date-time

tenant (optional)
[Tenant](#)

username (optional)
[String](#)

AwsScheduleOutput - AwsScheduleOutput

[Up](#)

id (optional)
[String](#)

schedule (optional)
[AwsSchedule](#)

AzureDirectory - AzureDirectory

[Up](#)

countryCode (optional)
[String](#)

displayName (optional)
[String](#)

domains (optional)
[array\[String\]](#)

id (optional)
[String](#)

tenantId (optional)
[String](#)

AzureKey - AzureKey

[Up](#)

autoRotate (optional)
[Boolean](#)

autoRotateDsm (optional)
[Boolean](#)

autoRotateHsm (optional)
[Boolean](#)

autoRotateKeyProvider (optional)
[String](#)
Enum:
DSM
DB
nShield

autoRotateKeyType (optional)
[String](#)
Enum:
AES256
RSA2048
RSA4096

RSA3072
ECP256
ECP384
ECP521
ECSECP256K1

autoRotateOnExpire (optional)

[*Boolean*](#)

autoRotateUser (optional)

[*String*](#)

availableInKeyVault (optional)

[*Boolean*](#)

backup (optional)

[*KeyVaultKey*](#)

backupAt (optional)

[*Long*](#) format: int64

createdAt (optional)

[*Date*](#) format: date-time

createdBy (optional)

[*String*](#)

deleted (optional)

[*Boolean*](#)

description (optional)

[*String*](#)

id (optional)

[*ObjectId*](#)

keySoftDeletedInAzure (optional)

[*Boolean*](#)

keyType (optional)

[*String*](#)

Enum:

AES256
RSA2048
RSA4096
RSA3072
ECP256
ECP384
ECP521
ECSECP256K1

keyVault (optional)

[*String*](#)

keyVaultDisplayName (optional)

[*String*](#)

keyVaultLocation (optional)

[*String*](#)

keyVaultLocationDisplay (optional)

[*String*](#)

kid (optional)

[*String*](#)

modifiedAt (optional)

[*Date*](#) format: date-time

modifiedBy (optional)

[*String*](#)

name (optional)

[*String*](#)

originalKey (optional)

[*KeyVaultKey*](#)

softDeleteEnabled (optional)

[*Boolean*](#)

source (optional)

[*String*](#)

Enum:

KMaaS
Salesforce
Azure

status (optional)

[*String*](#)

Enum:

CREATED
UPLOADED
EXPORTED
ALREADY_DESTROYED

subscriptionId (optional)

[*String*](#)

syncedAt (optional)

[*Date*](#) format: date-time

tenant (optional)

[*Tenant*](#)

uploadedAt (optional)

[*Date*](#) format: date-time

uploadedBy (optional)

[*String*](#)

uploadedFrom (optional)

[*String*](#)

versions (optional)

[*array\[AzureKeyVersion\]*](#)

AzureKeyAttributes - AzureKeyAttributes

[Up](#)

created (optional)

[*Long*](#) format: int64

enabled (optional)

[*Boolean*](#)

exp (optional)
[Long](#) format: int64
nbf (optional)
[Long](#) format: int64
recoveryLevel (optional)
[String](#)
updated (optional)
[Long](#) format: int64

AzureKeyResponse - AzureKeyResponse

[Up](#)

azureKey (optional)
[AzureKey](#)
id (optional)
[String](#)

AzureKeyVault - AzureKeyVault

[Up](#)

id (optional)
[String](#)
location (optional)
[String](#)
locationDisplay (optional)
[String](#)
myPermissions (optional)
[array\[String\]](#)
name (optional)
[String](#)
properties (optional)
[KeyVaultProperties](#)
resourceGroup (optional)
[String](#)
subscription (optional)
[String](#)
tags (optional)
[map\[String, String\]](#)
type (optional)
[String](#)

AzureKeyVersion - AzureKeyVersion

[Up](#)

attributes (optional)
[AzureKeyAttributes](#)
crv (optional)
[String](#)
description (optional)
[String](#)
e (optional)
[String](#)
keyVault (optional)
[String](#)
key_ops (optional)
[array\[String\]](#)
Enum:
key_size (optional)
[Integer](#) format: int32
kid (optional)
[String](#)
kty (optional)
[String](#)
n (optional)
[String](#)
source (optional)
[String](#)
Enum:
KMaaS
Salesforce
Azure
tags (optional)
[map\[String, String\]](#)
uploadedAt (optional)
[Date](#) format: date-time
uploadedBy (optional)
[String](#)
uploadedFrom (optional)
[String](#)
version (optional)
[String](#)

AzureSchedule - AzureSchedule

[Up](#)

applyOnExpiration (optional)
[Boolean](#)
createdAt (optional)
[Date](#) format: date-time
createdBy (optional)
[String](#)
cron (optional)
[String](#)

description (optional) String fixedRateSchedule (optional) Boolean humanReadableCron (optional) String id (optional) ObjectId interval (optional) Integer format: int32 intervalKeyExpiration (optional) Integer format: int32 intervalUnit (optional) String Enum: Hour Day Week Month Year None intervalUnitKeyExpiration (optional) String Enum: Hour Day Week Month Year None jobType (optional) String Enum: KEY_ROTATION KEY_SYNCHRONIZATION KEY_EXPIRATION modifiedAt (optional) Date format: date-time modifiedBy (optional) String name (optional) String password (optional) String pause (optional) Boolean startDate (optional) Date format: date-time tenant (optional) Tenant username (optional) String	
AzureScheduleOutput - AzureScheduleOutput id (optional) String schedule (optional) AzureSchedule	Up
AzureShortSubscriptionResponse - AzureShortSubscriptionResponse displayName (optional) String subscriptionID (optional) String	Up
Condition - Condition conditionKey (optional) String type (optional) String values (optional) array[String]	Up
DBObject - DBObject partialObject (optional) Boolean	Up
DeleteRequestBody - DeleteRequestBody deleteKeyAfterDays (optional) Integer format: int32 deleteKeyMaterialNow (optional) Boolean	Up
EmbeddedCollection«AwsKey» - EmbeddedCollection«AwsKey» Embedded collection of AwsKey awsKeys	Up

array[AwsKey] Resource collection	
EmbeddedCollection«AwsSchedule» - EmbeddedCollection«AwsSchedule»	Up
Embedded collection of AwsSchedule	
awsSchedules array[AwsSchedule] Resource collection	
EmbeddedCollection«AzureKey» - EmbeddedCollection«AzureKey»	Up
Embedded collection of AzureKey	
azureKeys array[AzureKey] Resource collection	
EmbeddedCollection«AzureSchedule» - EmbeddedCollection«AzureSchedule»	Up
Embedded collection of AzureSchedule	
azureSchedules array[AzureSchedule] Resource collection	
EmbeddedCollection«Report» - EmbeddedCollection«Report»	Up
Embedded collection of Report	
reports array[Report] Resource collection	
EmbeddedCollection«SalesforceKey» - EmbeddedCollection«SalesforceKey»	Up
Embedded collection of SalesforceKey	
salesforceKeys array[SalesforceKey] Resource collection	
EmbeddedCollection«SalesforceSchedule» - EmbeddedCollection«SalesforceSchedule»	Up
Embedded collection of SalesforceSchedule	
salesforceSchedules array[SalesforceSchedule] Resource collection	
GrantConstraints - GrantConstraints	Up
encryptionContextEquals (optional) map[String, String]	
encryptionContextSubset (optional) map[String, String]	
GrantListEntry - GrantListEntry	Up
constraints (optional) GrantConstraints	
creationDate (optional) Date format: date-time	
grantId (optional) String	
granteePrincipal (optional) String	
issuingAccount (optional) String	
keyId (optional) String	
name (optional) String	
operations (optional) array[String]	
retiringPrincipal (optional) String	
GrantedAuthority - GrantedAuthority	Up
authority (optional) String	
ImportKeyRequestData - ImportKeyRequestData	Up
sourceKeyId (optional) String	
validTo (optional) Long format: int64	
Key - Key	Up
hash (optional) String	
length (optional) Integer format: int32	
value (optional) String	
KeyFileRequestData - KeyFileRequestData	Up
attributes (optional)	

[AzureKeyAttributes](#)
hsm (optional)
[Boolean](#)
keyId (optional)
[String](#)
keyName (optional)
[String](#)
keyVaultDisplayName (optional)
[String](#)
keyVaultKeyId (optional)
[String](#)
keyVaultName (optional)
[String](#)
key_ops (optional)
[array\[String\]](#)
pfxpassword (optional)
[String](#)
softDeleteEnabled (optional)
[Boolean](#)
sourceKeyName (optional)
[String](#)
tags (optional)
[map\[String, String\]](#)
type (optional)
[String](#)
user (optional)
[String](#)

KeyMetadata - KeyMetadata

[Up](#)

arn (optional)
[String](#)
awsaccountId (optional)
[String](#)
creationDate (optional)
[Date](#) format: date-time
deletionDate (optional)
[Date](#) format: date-time
description (optional)
[String](#)
enabled (optional)
[Boolean](#)
expirationModel (optional)
[String](#)
keyId (optional)
[String](#)
keyManager (optional)
[String](#)
keyState (optional)
[String](#)
keyUsage (optional)
[String](#)
origin (optional)
[String](#)
validTo (optional)
[Date](#) format: date-time

KeyModifyRequestData - KeyModifyRequestData

[Up](#)

description (optional)
[String](#)

KeyOperationResponse - KeyOperationResponse

[Up](#)

code (optional)
[Integer](#) format: int32
detailedMessage (optional)
[String](#)
message (optional)
[String](#)

KeyProducer - KeyProducer

[Up](#)

name (optional)
[String](#)
provider (optional)
[String](#)
Enum:
 DSM
 DB
 nShield

KeyUpdateRequestData - KeyUpdateRequestData

[Up](#)

attributes (optional)
[AzureKeyAttributes](#)
keyId (optional)
[String](#)

keyName (optional)
[String](#)

keyVaultKeyId (optional)
[String](#)

keyVaultName (optional)
[String](#)

key_ops (optional)
[array\[String\]](#)
Enum:

tags (optional)
[map\[String, String\]](#)

KeyUploadRequestData - KeyUploadRequestData

[Up](#)

certificateLabel (optional)
[String](#)

keyDerivationMode (optional)
[String](#)

keyPassphrase (optional)
[String](#)

keyVaultKeyId (optional)
[String](#)

type (optional)
[String](#)
Enum:
Data
SearchIndex
DeterministicData
EventBus
Analytics
None

KeyVaultKey - KeyVaultKey

[Up](#)

algorithm (optional)
[String](#)

allTenant (optional)
[Boolean](#)

blob (optional)
[String](#)

blobUUID (optional)
[String](#)

cckmId (optional)
[String](#)

certificate (optional)
[String](#)

createdAt (optional)
[Date](#) format: date-time

createdBy (optional)
[String](#)

deleted (optional)
[Boolean](#)

description (optional)
[String](#)

hashType (optional)
[String](#)

id (optional)
[ObjectId](#)

keySource (optional)
[String](#)
Enum:
KMaaS
Salesforce
Azure

modifiedAt (optional)
[Date](#) format: date-time

modifiedBy (optional)
[String](#)

nShieldKeyBlob (optional)
[String](#)

name (optional)
[String](#)

privateKey (optional)
[Key](#)

protected (optional)
[Boolean](#)

provider (optional)
[String](#)
Enum:
DSM
DB
nShield

publicKey (optional)
[Key](#)

secretKey (optional)
[Key](#)

service (optional)
[String](#)
Enum:
NONE
Salesforce

SalesforceSandbox
Azure
AWS
AzureGermany
AzureChina
AzureStack
AzureStackAdmin
AzureGovernment

serviceKeyPage (optional)
[Page<ServiceKey>](#)

sourceKeyTier (optional)
[String](#)

tenant (optional)
[Tenant](#)

KeyVaultKeyResponse - KeyVaultKeyResponse

[Up](#)

id (optional)
[String](#)

keyVaultKey (optional)
[KeyVaultKey](#)

KeyVaultProperties - KeyVaultProperties

[Up](#)

accessPolicies (optional)
[array\[AccessPolicy\]](#)

enableSoftDelete (optional)
[Boolean](#)

enabledForDeployment (optional)
[Boolean](#)

enabledForDiskencryption (optional)
[Boolean](#)

enabledForTemplateDeployment (optional)
[Boolean](#)

sku (optional)
[Sku](#)

tenantId (optional)
[String](#)

vaultUri (optional)
[String](#)

License - License

[Up](#)

awsLicenseExpired (optional)
[Boolean](#)

azureLicenseExpired (optional)
[Boolean](#)

byok_azure_applications (optional)
[String](#)

byok_license_expires_aws (optional)
[String](#)

byok_license_expires_azure (optional)
[String](#)

byok_license_expires_sfdc (optional)
[String](#)

byok_license_sfdc_org_ids (optional)
[String](#)

byok_license_type_aws (optional)
[String](#)

byok_license_type_azure (optional)
[String](#)

byok_license_type_sfdc (optional)
[String](#)

byok_max_azure_applications (optional)
[String](#)

byok_max_sfdc_orgs (optional)
[String](#)

sfdcLicenseExpired (optional)
[Boolean](#)

Link - Link

[Up](#)

href (optional)
[String](#)

rel (optional)
[String](#)

templated (optional)
[Boolean](#)

NativeKeyRequestData - NativeKeyRequestData

[Up](#)

attributes (optional)
[AzureKeyAttributes](#)

hsm (optional)
[Boolean](#)

keyId (optional)
[String](#)

keyName (optional)
[String](#)

keyType (optional) String Enum: - AES256 - RSA2048 - RSA4096 - RSA3072 - ECP256 - ECP384 - ECP521 - ECSECP256K1 keyVaultDisplayName (optional) String keyVaultKeyId (optional) String keyVaultName (optional) String key_ops (optional) array[String] softDeleteEnabled (optional) Boolean tags (optional) map[String, String]	
OAuth2AccessToken - OAuth2AccessToken additionalInformation (optional) Object expiration (optional) Date format: date-time expired (optional) Boolean expiresIn (optional) Integer format: int32 refreshToken (optional) OAuth2RefreshToken scope (optional) array[String] tokenType (optional) String value (optional) String	Up
OAuth2RefreshToken - OAuth2RefreshToken value (optional) String	Up
ObjectId - ObjectId counter (optional) Integer format: int32 date (optional) Date format: date-time machineIdentifier (optional) Integer format: int32 processIdentifier (optional) Integer format: int32 time (optional) Long format: int64 timeSecond (optional) Integer format: int32 timestamp (optional) Integer format: int32	Up
Optional«AzureKey» - Optional«AzureKey» present (optional) Boolean	Up
Optional«List«Report»» - Optional«List«Report»» present (optional) Boolean	Up
Optional«Report» - Optional«Report» present (optional) Boolean	Up
Optional«SalesforceKey» - Optional«SalesforceKey» present (optional) Boolean	Up
Page - Page content (optional) array[Object] first (optional)	Up

[Boolean](#)

last (optional)

[Boolean](#)

number (optional)
[Integer](#) format: int32

numberOfElements (optional)
[Integer](#) format: int32

size (optional)
[Integer](#) format: int32

sort (optional)
[Sort](#)

totalElements (optional)
[Long](#) format: int64

totalPages (optional)
[Integer](#) format: int32

Page«AzureKey» - Page«AzureKey»

[Up](#)

content (optional)

[array\[AzureKey\]](#)

first (optional)

[Boolean](#)

last (optional)

[Boolean](#)

number (optional)
[Integer](#) format: int32

numberOfElements (optional)
[Integer](#) format: int32

size (optional)
[Integer](#) format: int32

sort (optional)
[Sort](#)

totalElements (optional)
[Long](#) format: int64

totalPages (optional)
[Integer](#) format: int32

Page«DBObject» - Page«DBObject»

[Up](#)

content (optional)

[array\[DBObject\]](#)

first (optional)

[Boolean](#)

last (optional)

[Boolean](#)

number (optional)
[Integer](#) format: int32

numberOfElements (optional)
[Integer](#) format: int32

size (optional)
[Integer](#) format: int32

sort (optional)
[Sort](#)

totalElements (optional)
[Long](#) format: int64

totalPages (optional)
[Integer](#) format: int32

Page«KeyVaultKeyResponse» - Page«KeyVaultKeyResponse»

[Up](#)

content (optional)

[array\[KeyVaultKeyResponse\]](#)

first (optional)

[Boolean](#)

last (optional)

[Boolean](#)

number (optional)
[Integer](#) format: int32

numberOfElements (optional)
[Integer](#) format: int32

size (optional)
[Integer](#) format: int32

sort (optional)
[Sort](#)

totalElements (optional)
[Long](#) format: int64

totalPages (optional)
[Integer](#) format: int32

Page«SalesforceKey» - Page«SalesforceKey»

[Up](#)

content (optional)

[array\[SalesforceKey\]](#)

first (optional)

[Boolean](#)

last (optional)
[Boolean](#)

number (optional) Integer format: int32 numberOfElements (optional) Integer format: int32 size (optional) Integer format: int32 sort (optional) Sort totalElements (optional) Long format: int64 totalPages (optional) Integer format: int32	
Page«ServiceKey» - Page«ServiceKey»	Up
content (optional) array[ServiceKey] first (optional) Boolean last (optional) Boolean number (optional) Integer format: int32 numberOfElements (optional) Integer format: int32 size (optional) Integer format: int32 sort (optional) Sort totalElements (optional) Long format: int64 totalPages (optional) Integer format: int32	
Permission - Permission	Up
certificates (optional) array[String] keys (optional) array[String] secrets (optional) array[String]	
PolicyPrincipal - PolicyPrincipal	Up
pid (optional) String provider (optional) String	
PolicyStatement - PolicyStatement	Up
actions (optional) array[Action] conditions (optional) array[Condition] effect (optional) String Enum: Allow Deny id (optional) String principals (optional) array[PolicyPrincipal] resources (optional) array[Resource]	
PrincipalAndUser - PrincipalAndUser	Up
aws (optional) array[User] azure (optional) array[User] azureChina (optional) array[User] azureGermany (optional) array[User] azureStack (optional) array[User] azureStackAdmin (optional) array[User] azureUsGov (optional) array[User] cloudServiceUsers (optional) map[String, User] latestLoggedInSession (optional) String	

latestLoggedInUser (optional) User latestLoggedInUserService (optional) String loggedInUsers (optional) Integer format: int32 name (optional) String salesforce (optional) array[User] salesforcesandbox (optional) array[User] userAuthentication (optional) Authentication	
Report - Report	Up
createdAt (optional) Date format: date-time createdBy (optional) String csp (optional) String Enum: aws azure salesforce from (optional) Date format: date-time id (optional) ObjectId modifiedAt (optional) Date format: date-time modifiedBy (optional) String name (optional) String status (optional) String Enum: REQUESTED READY FAILED tenant (optional) Tenant to (optional) Date format: date-time type (optional) String Enum: KEY_360_RECONCILIATION KEY_ACTIVITY KEY_AGING KEY_SERVICE_USAGE USER_ACTIVITY	
Resource - Resource	Up
id (optional) String	
ResourceSupport - ResourceSupport	Up
links (optional) array[Link]	
Resources«AwsKey» - Resources«AwsKey»	Up
Resources of AwsKey	
_embedded EmbeddedCollection~AwsKey~ _links map[String, Link] Link collection	
Resources«AwsSchedule» - Resources«AwsSchedule»	Up
Resources of AwsSchedule	
_embedded EmbeddedCollection~AwsSchedule~ _links map[String, Link] Link collection	
Resources«AzureKey» - Resources«AzureKey»	Up
Resources of AzureKey	
_embedded EmbeddedCollection~AzureKey~ _links map[String, Link] Link collection	
Resources«AzureSchedule» - Resources«AzureSchedule»	Up

Resources of AzureSchedule

[_embedded](#)
[EmbeddedCollection~AzureSchedule»](#)
[_links](#)
[map\[String, Link\]](#) Link collection

Resources«Report» - Resources«Report»

[Up](#)

Resources of Report

[_embedded](#)
[EmbeddedCollection~Report»](#)
[_links](#)
[map\[String, Link\]](#) Link collection

Resources«SalesforceKey» - Resources«SalesforceKey»

[Up](#)

Resources of SalesforceKey

[_embedded](#)
[EmbeddedCollection~SalesforceKey»](#)
[_links](#)
[map\[String, Link\]](#) Link collection

Resources«SalesforceSchedule» - Resources«SalesforceSchedule»

[Up](#)

Resources of SalesforceSchedule

[_embedded](#)
[EmbeddedCollection~SalesforceSchedule»](#)
[_links](#)
[map\[String, Link\]](#) Link collection

Resource«AwsKey» - Resource«AwsKey»

[Up](#)

[aliases](#) (optional)
[array\[AwsKeyAlias\]](#)
[autoRotate](#) (optional)
[Boolean](#)
[autoRotateDisableEncrypt](#) (optional)
[Boolean](#)
[autoRotateKeyProvider](#) (optional)
[String](#)
Enum:
 DSM
 DB
 nShield
[autoRotateOnExpire](#) (optional)
[Boolean](#)
[autoRotateUser](#) (optional)
[String](#)
[availableInKeyVault](#) (optional)
[Boolean](#)
[backup](#) (optional)
[KeyVaultKey](#)
[createdAt](#) (optional)
[Date](#) format: date-time
[createdBy](#) (optional)
[String](#)
[grants](#) (optional)
[array\[GrantListEntry\]](#)
[id](#) (optional)
[Objectid](#)
[keyMetadata](#) (optional)
[KeyMetadata](#)
[keyType](#) (optional)
[String](#)
Enum:
 AES256
 RSA2048
 RSA4096
 RSA3072
 ECP256
 ECP384
 ECP521
 ECSECP256K1
[links](#) (optional)
[array\[Link\]](#)
[modifiedAt](#) (optional)
[Date](#) format: date-time
[modifiedBy](#) (optional)
[String](#)
[name](#) (optional)
[String](#)
[originalKey](#) (optional)
[KeyVaultKey](#)
[policy](#) (optional)
[AwsPolicy](#)
[region](#) (optional)
[String](#)
[regionLabel](#) (optional)
[String](#)
[rotationChainId](#) (optional)
[String](#)

rotationStatus (optional)
[String](#)
Enum:
 ARCHIVED
 ACTIVE

source (optional)
[String](#)
Enum:
 KMaaS
 Salesforce
 Azure

status (optional)
[String](#)
Enum:
 CREATED
 UPLOADED
 EXPORTED
 ALREADY_DESTROYED

syncedAt (optional)
[Date](#) format: date-time

tags (optional)
[array\[Tag\]](#)

tenant (optional)
[Tenant](#)

uploadedAt (optional)
[Date](#) format: date-time

uploadedBy (optional)
[String](#)

uploadedFrom (optional)
[String](#)

Resource«AwsSchedule» - Resource«AwsSchedule»

[Up](#)

accessKeyId (optional)
[String](#)

applyOnExpiration (optional)
[Boolean](#)

createdAt (optional)
[Date](#) format: date-time

createdBy (optional)
[String](#)

cron (optional)
[String](#)

description (optional)
[String](#)

fixedRateSchedule (optional)
[Boolean](#)

humanReadableCron (optional)
[String](#)

id (optional)
[ObjectId](#)

interval (optional)
[Integer](#) format: int32

intervalKeyExpiration (optional)
[Integer](#) format: int32

intervalUnit (optional)
[String](#)
Enum:
 Hour
 Day
 Week
 Month
 Year
 None

intervalUnitKeyExpiration (optional)
[String](#)
Enum:
 Hour
 Day
 Week
 Month
 Year
 None

jobType (optional)
[String](#)
Enum:
 KEY_ROTATION
 KEY_SYNCHRONIZATION
 KEY_EXPIRATION

links (optional)
[array\[Link\]](#)

modifiedAt (optional)
[Date](#) format: date-time

modifiedBy (optional)
[String](#)

name (optional)
[String](#)

pause (optional)
[Boolean](#)

startDate (optional)
[Date](#) format: date-time

tenant (optional)
[Tenant](#)

username (optional)
[String](#)

Resource«AzureKey» - Resource«AzureKey»

[Up](#)

autoRotate (optional)
[Boolean](#)

autoRotateDsm (optional)
[Boolean](#)

autoRotateHsm (optional)
[Boolean](#)

autoRotateKeyProvider (optional)
[String](#)
Enum:
DSM
DB
nShield

autoRotateKeyType (optional)
[String](#)
Enum:
AES256
RSA2048
RSA4096
RSA3072
ECP256
ECP384
ECP521
ECSECP256K1

autoRotateOnExpire (optional)
[Boolean](#)

autoRotateUser (optional)
[String](#)

availableInKeyVault (optional)
[Boolean](#)

backup (optional)
[KeyVaultKey](#)

backupAt (optional)
[Long](#) format: int64

createdAt (optional)
[Date](#) format: date-time

createdBy (optional)
[String](#)

deleted (optional)
[Boolean](#)

description (optional)
[String](#)

id (optional)
[ObjectId](#)

keySoftDeletedInAzure (optional)
[Boolean](#)

keyType (optional)
[String](#)
Enum:
AES256
RSA2048
RSA4096
RSA3072
ECP256
ECP384
ECP521
ECSECP256K1

keyVault (optional)
[String](#)

keyVaultDisplayName (optional)
[String](#)

keyVaultLocation (optional)
[String](#)

keyVaultLocationDisplay (optional)
[String](#)

kid (optional)
[String](#)

links (optional)
[array\[Link\]](#)

modifiedAt (optional)
[Date](#) format: date-time

modifiedBy (optional)
[String](#)

name (optional)
[String](#)

originalKey (optional)
[KeyVaultKey](#)

softDeleteEnabled (optional)
[Boolean](#)

source (optional)
[String](#)
Enum:
KMaaS
Salesforce
Azure

status (optional)
[String](#)
Enum:
CREATED

UPLOADED
EXPORTED
ALREADY_DESTROYED

subscriptionId (optional)

[String](#)

syncedAt (optional)

[Date](#) format: date-time

tenant (optional)

[Tenant](#)

uploadedAt (optional)

[Date](#) format: date-time

uploadedBy (optional)

[String](#)

uploadedFrom (optional)

[String](#)

versions (optional)

[array\[AzureKeyVersion\]](#)

Resource«AzureSchedule» - Resource«AzureSchedule»

[Up](#)

applyOnExpiration (optional)

[Boolean](#)

createdAt (optional)

[Date](#) format: date-time

createdBy (optional)

[String](#)

cron (optional)

[String](#)

description (optional)

[String](#)

fixedRateSchedule (optional)

[Boolean](#)

humanReadableCron (optional)

[String](#)

id (optional)

[ObjectId](#)

interval (optional)

[Integer](#) format: int32

intervalKeyExpiration (optional)

[Integer](#) format: int32

intervalUnit (optional)

[String](#)

Enum:

Hour

Day

Week

Month

Year

None

intervalUnitKeyExpiration (optional)

[String](#)

Enum:

Hour

Day

Week

Month

Year

None

jobType (optional)

[String](#)

Enum:

KEY_ROTATION

KEY_SYNCHRONIZATION

KEY_EXPIRATION

links (optional)

[array\[Link\]](#)

modifiedAt (optional)

[Date](#) format: date-time

modifiedBy (optional)

[String](#)

name (optional)

[String](#)

pause (optional)

[Boolean](#)

startDate (optional)

[Date](#) format: date-time

tenant (optional)

[Tenant](#)

username (optional)

[String](#)

Resource«Optional«AzureKey»» - Resource«Optional«AzureKey»»

[Up](#)

links (optional)

[array\[Link\]](#)

present (optional)

[Boolean](#)

Resource«Optional«List«Report»»» - Resource«Optional«List«Report»»»

[Up](#)

links (optional)

[array\[Link\]](#)

present (optional) Boolean	
Resource«Optional«Report»» - Resource«Optional«Report»»	Up
links (optional) array[Link] present (optional) Boolean	
Resource«Optional«SalesforceKey»» - Resource«Optional«SalesforceKey»»	Up
links (optional) array[Link] present (optional) Boolean	
Resource«Page«AzureKey»» - Resource«Page«AzureKey»»	Up
content (optional) array[AzureKey] first (optional) Boolean last (optional) Boolean links (optional) array[Link] number (optional) Integer format: int32 numberOfElements (optional) Integer format: int32 size (optional) Integer format: int32 sort (optional) Sort totalElements (optional) Long format: int64 totalPages (optional) Integer format: int32	
Resource«Page«SalesforceKey»» - Resource«Page«SalesforceKey»»	Up
content (optional) array[SalesforceKey] first (optional) Boolean last (optional) Boolean links (optional) array[Link] number (optional) Integer format: int32 numberOfElements (optional) Integer format: int32 size (optional) Integer format: int32 sort (optional) Sort totalElements (optional) Long format: int64 totalPages (optional) Integer format: int32	
Resource«Report» - Resource«Report»	Up
createdAt (optional) Date format: date-time createdBy (optional) String csp (optional) String Enum: aws azure salesforce from (optional) Date format: date-time id (optional) ObjectId links (optional) array[Link] modifiedAt (optional) Date format: date-time modifiedBy (optional) String name (optional) String status (optional) String	

Enum:
REQUESTED
READY
FAILED

tenant (optional)
[Tenant](#)

to (optional)
[Date](#) format: date-time

type (optional)
[String](#)
Enum:
KEY_360_RECONCILIATION
KEY_ACTIVITY
KEY_AGING
KEY_SERVICE_USAGE
USER_ACTIVITY

Resource«SalesforceKey» - Resource«SalesforceKey»

[Up](#)

availableInKeyVault (optional)
[Boolean](#)

backup (optional)
[KeyVaultKey](#)

createdAt (optional)
[Date](#) format: date-time

createdBy (optional)
[String](#)

description (optional)
[String](#)

displayOrder (optional)
[Integer](#) format: int32

id (optional)
[Objectid](#)

keyDerivationMode (optional)
[String](#)

keyType (optional)
[String](#)
Enum:
AES256
RSA2048
RSA4096
RSA3072
ECP256
ECP384
ECP521
ECSECP256K1

links (optional)
[array\[Link\]](#)

modifiedAt (optional)
[Date](#) format: date-time

modifiedBy (optional)
[String](#)

name (optional)
[String](#)

originalKey (optional)
[KeyVaultKey](#)

source (optional)
[String](#)
Enum:
KMaaS
Salesforce
Azure

status (optional)
[String](#)
Enum:
CREATED
UPLOADED
EXPORTED
ALREADY_DESTROYED

syncedAt (optional)
[Date](#) format: date-time

tenant (optional)
[Tenant](#)

tenantSecret (optional)
[SalesforceTenantSecret](#)

type (optional)
[String](#)
Enum:
Data
SearchIndex
DeterministicData
EventBus
Analytics
None

uploadedAt (optional)
[Date](#) format: date-time

uploadedBy (optional)
[String](#)

uploadedFrom (optional)
[String](#)

uploadedToSalesforceBy (optional)
[String](#)

Resource«SalesforceSchedule» - Resource«SalesforceSchedule»

[Up](#)

applyOnExpiration (optional)
[Boolean](#)

certificate (optional)
[String](#)

createdAt (optional)
[Date](#) format: date-time

createdBy (optional)
[String](#)

cron (optional)
[String](#)

description (optional)
[String](#)

fixedRateSchedule (optional)
[Boolean](#)

humanReadableCron (optional)
[String](#)

id (optional)
[Objectid](#)

interval (optional)
[Integer](#) format: int32

intervalKeyExpiration (optional)
[Integer](#) format: int32

intervalUnit (optional)
[String](#)
Enum:
Hour
Day
Week
Month
Year
None

intervalUnitKeyExpiration (optional)
[String](#)
Enum:
Hour
Day
Week
Month
Year
None

jobType (optional)
[String](#)
Enum:
KEY_ROTATION
KEY_SYNCHRONIZATION
KEY_EXPIRATION

keyDerivationMode (optional)
[String](#)

links (optional)
[array\[Link\]](#)

modifiedAt (optional)
[Date](#) format: date-time

modifiedBy (optional)
[String](#)

name (optional)
[String](#)

pause (optional)
[Boolean](#)

startDate (optional)
[Date](#) format: date-time

tenant (optional)
[Tenant](#)

types (optional)
[array\[String\]](#)
Enum:

username (optional)
[String](#)

RotateKeyRequestData - RotateKeyRequestData

[Up](#)

awsKeyId (optional)
[String](#)

description (optional)
[String](#)

disableEncrypt (optional)
[Boolean](#)

sourceKeyId (optional)
[String](#)

validTo (optional)
[Long](#) format: int64

SalesforceKey - SalesforceKey

[Up](#)

availableInKeyVault (optional)
[Boolean](#)

backup (optional)
[KeyVaultKey](#)

createdAt (optional)
[Date](#) format: date-time

createdBy (optional)
[String](#)

description (optional)
[String](#)

displayOrder (optional)
[Integer](#) format: int32

id (optional)
[Objectid](#)

keyDerivationMode (optional)
[String](#)

keyType (optional)
[String](#)
Enum:
AES256
RSA2048
RSA4096
RSA3072
ECP256
ECP384
ECP521
ECSECP256K1

modifiedAt (optional)
[Date](#) format: date-time

modifiedBy (optional)
[String](#)

name (optional)
[String](#)

originalKey (optional)
[KeyVaultKey](#)

source (optional)
[String](#)
Enum:
KMaaS
Salesforce
Azure

status (optional)
[String](#)
Enum:
CREATED
UPLOADED
EXPORTED
ALREADY_DESTROYED

syncedAt (optional)
[Date](#) format: date-time

tenant (optional)
[Tenant](#)

tenantSecret (optional)
[SalesforceTenantSecret](#)

type (optional)
[String](#)
Enum:
Data
SearchIndex
DeterministicData
EventBus
Analytics
None

uploadedAt (optional)
[Date](#) format: date-time

uploadedBy (optional)
[String](#)

uploadedFrom (optional)
[String](#)

uploadedToSalesforceBy (optional)
[String](#)

SalesforceSchedule - SalesforceSchedule

applyOnExpiration (optional)
[Boolean](#)

certificate (optional)
[String](#)

createdAt (optional)
[Date](#) format: date-time

createdBy (optional)
[String](#)

cron (optional)
[String](#)

description (optional)
[String](#)

fixedRateSchedule (optional)
[Boolean](#)

humanReadableCron (optional)
[String](#)

id (optional)
[Objectid](#)

interval (optional)
[Integer](#) format: int32

intervalKeyExpiration (optional)
[Integer](#) format: int32

intervalUnit (optional)
[String](#)
Enum:
Hour
Day

Week
Month
Year
None

intervalUnitKeyExpiration (optional)

[String](#)

Enum:

Hour
Day
Week
Month
Year
None

jobType (optional)

[String](#)

Enum:

KEY_ROTATION
KEY_SYNCHRONIZATION
KEY_EXPIRATION

keyDerivationMode (optional)

[String](#)

modifiedAt (optional)

[Date](#) format: date-time

modifiedBy (optional)

[String](#)

name (optional)

[String](#)

password (optional)

[String](#)

pause (optional)

[Boolean](#)

startDate (optional)

[Date](#) format: date-time

tenant (optional)

[Tenant](#)

types (optional)

[array\[String\]](#)

Enum:

username (optional)

[String](#)

SalesforceScheduleOutput - SalesforceScheduleOutput

[Up](#)

id (optional)

[String](#)

schedule (optional)

[SalesforceSchedule](#)

SalesforceTenantSecret - SalesforceTenantSecret

[Up](#)

CreatedById (optional)

[String](#)

Description (optional)

[String](#)

Id (optional)

[String](#)

IsDeleted (optional)

[Boolean](#)

KeyDerivationMode (optional)

[String](#)

LastModifiedById (optional)

[String](#)

LastModifiedByUsername (optional)

[String](#)

LastModifiedDate (optional)

[Date](#) format: date-time

SecretValueCertificate (optional)

[String](#)

SecretValueHash (optional)

[String](#)

SystemModstamp (optional)

[Date](#) format: date-time

Type (optional)

[String](#)

attributes (optional)

[Attribute](#)

createdByUsername (optional)

[String](#)

createdDate (optional)

[Date](#) format: date-time

keyvaultName (optional)

[String](#)

source (optional)

[String](#)

status (optional)

[String](#)

Enum:

ACTIVE
ARCHIVED
DESTROYED

version (optional)

[Integer](#) format: int32

ServiceKey - ServiceKey

[Up](#)

availableInKeyVault (optional)

[Boolean](#)

backup (optional)

[KeyVaultKey](#)

createdAt (optional)

[Date](#) format: date-time

createdBy (optional)

[String](#)

id (optional)

[ObjectId](#)

keyType (optional)

[String](#)

Enum:

AES256

RSA2048

RSA4096

RSA3072

ECP256

ECP384

ECP521

ECSECP256K1

modifiedAt (optional)

[Date](#) format: date-time

modifiedBy (optional)

[String](#)

name (optional)

[String](#)

originalKey (optional)

[KeyVaultKey](#)

source (optional)

[String](#)

Enum:

KMaaS

Salesforce

Azure

status (optional)

[String](#)

Enum:

CREATED

UPLOADED

EXPORTED

ALREADY_DESTROYED

syncedAt (optional)

[Date](#) format: date-time

tenant (optional)

[Tenant](#)

uploadedAt (optional)

[Date](#) format: date-time

uploadedBy (optional)

[String](#)

SessionData - SessionData

[Up](#)

accessToken (optional)

[OAuth2AccessToken](#)

auth (optional)

[Authentication](#)

cloudService (optional)

[String](#)

id (optional)

[String](#)

tenant (optional)

[Tenant](#)

user (optional)

[User](#)

Sku - Sku

[Up](#)

family (optional)

[String](#)

name (optional)

[String](#)

Sort - Sort

[Up](#)

SystemInfo - SystemInfo

[Up](#)

build (optional)

[String](#)

buildNumber (optional)

[String](#)

major (optional)

[String](#)

minor (optional)

[String](#)

patch (optional)

[String](#)

version (optional) String	
Tag - Tag	Up
tagKey (optional) String	
tagValue (optional) String	
Tenant - Tenant	Up
cloudService (optional) String	
createdAt (optional) Date format: date-time	
createdBy (optional) String	
id (optional) Objectid	
modifiedAt (optional) Date format: date-time	
modifiedBy (optional) String	
name (optional) String	
ToggleAutoRotateData - ToggleAutoRotateData	Up
enable (optional) Boolean	
type (optional) String Enum: Data SearchIndex DeterministicData EventBus Analytics None	
UpdateKeyPolicyRequestData - UpdateKeyPolicyRequestData	Up
keyId (optional) String	
newKeyPolicy (optional) String	
region (optional) String	
UpdateKeyRequestData - UpdateKeyRequestData	Up
addAliases (optional) array[String]	
addTags (optional) map[String, String]	
description (optional) String	
enable (optional) Boolean	
keyId (optional) String	
region (optional) String	
removeAliases (optional) array[String]	
removeTags (optional) map[String, String]	
UploadKeyRequestData - UploadKeyRequestData	Up
addTags (optional) map[String, String]	
aliasName (optional) String	
allowedExternalAccounts (optional) array[String]	
awsKeyId (optional) String	
description (optional) String	
region (optional) String	
sourceKeyId (optional) String	
validTo (optional) Long format: int64	
User - User	Up

cckmld (optional)
[*String*](#)

createdAt (optional)
[*Date*](#) format: date-time

createdBy (optional)
[*String*](#)

id (optional)
[*ObjectId*](#)

is_first_login (optional)
[*Boolean*](#)

modifiedAt (optional)
[*Date*](#) format: date-time

modifiedBy (optional)
[*String*](#)

name (optional)
[*String*](#)

sessionId (optional)
[*String*](#)

tenant (optional)
[*Tenant*](#)

tenantDisplayName (optional)
[*String*](#)

VerifyAliasData - VerifyAliasData

alias (optional)
[*String*](#)

region (optional)
[*String*](#)

CipherTrust Cloud Key Manager Installation & Configuration Guide

Release 1.10.3

Document Version 1

July 15, 2022



CipherTrust Cloud Key Manager Installation & Configuration Guide

1.10.3

Document Version 1

July 15, 2022

All information herein is either public information or is the property of and owned solely by Thales DIS France S.A. and/or its subsidiaries or affiliates who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Thales DIS France S.A. and any of its subsidiaries and affiliates (collectively referred to herein after as "Thales") information.

This document can be used for informational, non-commercial, internal and personal use only provided that:

- The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any network computer or broadcast in any media and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Thales makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Thales reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

Copyright 2009 - 2022 Thales Group. All rights reserved. Thales and the Thales logo are trademarks and service marks of Thales and/or its subsidiaries and affiliates and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the properties of their respective owners.

Contents

Preface	17
Intended audience	17
Document Conventions	17
Typographical Conventions	17
Notes, Tips, Cautions, and Warnings	18
Hardware-Related Warnings	18
Sales and Support	19
 Chapter 1: Overview	 20
CCKM Components	20
Supported Clouds	21
Supported Key Sources	21
Supported Databases	21
Supported Deployment Methods	22
Supported Internet Browsers	22
CCKM Functionality	22
 Chapter 2: VM Specifications for CCKM Components	 24
With Local MongoDB bundled with CCKM Appliance (Recommended for PoCs Only)	24
PoC – Hardware Recommendation	24
Development/Staging – Hardware Recommendation	24
Production – Hardware Recommendation	24
With External MongoDB (Recommended for Production)	24
PoC – Hardware Recommendation	25
Development/Staging – Hardware Recommendation	25
Production – Hardware Recommendation	25
With MongoDB Atlas (Recommended for Production)	26
PoC – Hardware Recommendation	26
Development/Staging – Hardware Recommendation	26
Production – Hardware Recommendation	26
 Chapter 3: About CCKM Admin and User Portals	 27
CCKM Administration Privileges	27
Accessing CCKM Admin Portal (as CCKM Admin)	28
Accessing CCKM User Portal from CCKM Admin Portal	28
Accessing CCKM User Portal (as CCKM User)	31
Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI	32
Accessing the Multi-account Login/Logout Window	32

Switching to Another User Account (Already Logged On)	32
Logging on to Another User Account	33
Logging Out of a User Account	33
Logging Out of All User Accounts from a Specific Cloud Service or All Cloud Services	33
User Handling within CCKM User Portal (User Login Info)	33
User Authentication	33
User Authorization	34
User Audit	34

Chapter 4: How It Works: Installation and Configuration Flow 35

Step 1: Install and Configure MongoDB	35
Step 2: Install and Configure Key Source	35
Install and Configure CipherTrust Manager as your Key Source	35
Install and Configure DSM as your Key Source	36
Install and Configure nShield as your Key Source	36
Step 3: Setup to Connect to Cloud(s)	36
Connect to Azure	36
Connect to AWS	36
Connect to Google Cloud	37
Connect to IBM Cloud	37
Connect to Salesforce	37
Step 4: Install CCKM in One of the Supported Deployment Methods	37
Step 5: Perform Required Steps for the Initial Configurations of CCKM Admin Portal	37

Chapter 5: Set Up MongoDB 38

Scenario 1: Use Built-in MongoDB for Evaluation and PoC Purposes	38
Scenario 2: Configure an Existing (External) MongoDB (Single Node or Replica Set)	38
(Optional) Enable TLS	38
Create a Database (kmaas) and User (kmaas_user)	39
Scenario 3: Configure MongoDB Atlas	39
Whitelist Devices Allowed to Connect to MongoDB Atlas	39
(Optional) Add MongoDB (Custom) Role (kmaas_role)	40
Add MongoDB User (kmaas_user)	40
Create a New Cluster	41
Obtain Connection String for Connection Using a MongoDB Driver	41

Chapter 6: Install and Configure Key Source 42

Install and Configure CipherTrust Manager as Key Source	42
Install and Configure DSM as Key Source	43

Install and Configure nShield Connect as Key Source	43
Chapter 7: Register CCKM App in Azure Cloud	45
Registering CCKM App in Azure or Azure China Cloud	45
Create an Azure Active Directory Application (Multi-tenanted) in Azure Portal	45
Set Required Permissions on CCKM App	46
Assign Azure Contributor Role to CCKM App as Service Principal	47
Create a Key in Azure Portal	47
Chapter 8: Setting Up CCKM App for Access to Google Cloud	48
Set Up OAuth Consent Screen and Request Verification	48
Enable Identity and Access Management API	49
Enable Cloud Resource Manager API	49
Enable Cloud Key Management Service (KMS) API	49
Create OAuth2 Client ID Credentials	50
Set Session Duration for Google Cloud Console	50
Create Service Account	51
Grant IAM Roles to User and Service Accounts	52
Grant Project Browser and Cloud KMS Admin Roles	52
Grant Organization Viewer Role	52
Create Custom Roles	53
Chapter 9: Create a Connected CCKM App in Salesforce	54
Creating a Connected CCKM App in Salesforce	54
Editing CCKM App to Update OAuth Policies	55
Chapter 10: Install CCKM	57
Install on VMware ESXi Server	58
Install on AWS	59
Start the Installation	59
Configure Instance Details	59
Complete the Installation	60
Install on Microsoft Hyper-V	61
Create CCKM VM	61
Power on CCKM VM and Connect to It	61
Launch on Azure Marketplace	61
Use SSH to Change the Default Password on the CCKM VM	61
Synchronize CCKM Local time with Network Time using NTP Server	62
Configure Multiple NICs on CCKM Instance	63

Configure Network Interface	63
Add New Routes to Routing Table	64
Access the CCKM GUI	64
CCKM Cluster Configuration	65
Chapter 11: Initial Configurations within CCKM Admin Portal	66
Summary of Initial Configuration Steps	66
Change Default Admin Portal Password (During First-time Login)	67
Login into Admin Portal	67
Set Up Database	67
Upload License	67
Configure Key Source	68
Add Cloud	68
Chapter 12: About the Admin Portal	69
CCKM Admin Portal	69
Manage Clouds	70
Add Cloud	70
Step 1: Add Cloud	70
Set Access to Cloud	71
Set Access to AWS Cloud	71
Set Access to Azure (Azure, Azure China) Cloud	71
Set Access to Google Cloud	72
Set Access to IBM Cloud	72
Set Access to Salesforce Cloud	72
Set Access to Salesforce Government Cloud	73
Set Access to Salesforce Sandbox Cloud	73
Configure Key Source	73
Configure CipherTrust Manager as Key Source	74
Prerequisites	74
Configuring CipherTrust Manager	74
Configure DSM as Key Source	75
Prerequisites	75
Configuring DSM	75
Configure nShield Connect (for Azure and AWS Clouds)	75
Prerequisites	76
Configuring nShield	76
Review Logs	77
Review Health Monitor	77

Review Licenses	79
Settings	80
Configure Database	80
Configure Local Database	81
Configure Remote Database	81
Configure ReplicaSet Database	82
Configure MongoDB Atlas	82
Set Proxy Settings	83
Set SMTP Settings	83
Set Debug Log Level	84
Configure User Portal Preferences	84
Manage Deletion of Source Keys	84
Manage Masking of Email Addresses	85
Manage Users	85
Add User	85
Disable User	86
Enable User	86
Delete User	86
Reset Password	87
Update Admin Password Security	87
Chapter 13: CCKM for AWS Key Management	89
Overview	89
Prerequisites on AWS	90
Access AWS Key Management	91
Access CCKM for AWS	91
AWS CCKM User Portal	91
Home	93
Keys	93
Create a Key	96
Create a Native Key	96
Upload your Own Key (BYOK)	98
Delete a Key	98
Delete a BYOK	99
Schedule Delete	99
Schedule Delete Key	99
Import a Key	99
Update Key	100
Download Public Key (Asymmetric Key Only)	100
Rotate Key	100

Manually Rotate a Key (Symmetric Key)	101
Enable Autorotation of Key (Symmetric Key)	102
Manually Rotate a Key (Asymmetric Key, Encrypt, and Decrypt)	102
Manually Rotate a Key (Asymmetric Key, Sign, and Verify)	103
Enable Autorotation of Key (Asymmetric Key, Encrypt, and Decrypt)	103
Enable Autorotation of Key (Asymmetric Key, Sign, and Verify)	104
Synchronize Keys	104
Enable Autorotation before Key Expiration	105
Export Keys	105
Reports	106
Configure CloudTrail and CloudWatch in AWS to Use CCKM Reports	106
Overview	106
Configure CloudTrail	107
Configure CloudWatch Log Group	107
Schedules	107
Schedule Key Rotation	107
Set up a Key Rotation Schedule	108
Pause a Key Rotation Schedule	108
Delete a Key Rotation Schedule	108
Schedule Key Synchronization	108
Set up a Key Synchronization Schedule	109
Schedule Key Expiration	110
Set up a Key Expiration Schedule	110
Logs	110
Key Sources	111
Add (Generate) a Key	111
Add (Generate) a CipherTrust Manager Key	111
Add (Generate) a DSM Key	112
Add (Generate) an nShield Key	112
Delete a Key	112
Manage keys in Key Sources List Page	112
Settings	113
Adding Recipients' Email Addresses (for Notifications)	113
Enabling Notifications	113
Enabling Reporting for AWS Keys	113
Enabling Reporting for AWS Keys Using an Existing CloudTrail and CloudWatch Log Group ..	114
Enabling Reporting for AWS Keys Using a New CloudTrail and CloudWatch Log Group	114
Configuring Syslog Server	115
Configure Remote Syslog Server	115
Delete Remote Syslog Server Configuration	116

About	116
Chapter 14: CCKM for Azure Key Management	117
Overview	117
Accessing and Managing Azure Resources	117
Registering Apps in Azure	118
Prerequisites on Azure	118
Access CCKM for Azure	118
Accessing CCKM for Azure Using User Login	119
Accessing CCKM for Azure Using Service Principal	119
Granting Admin Consent First Time Accessing CCKM for Azure Using Service Principal	119
Accessing CCKM for Azure Using Service Principal (After Granting Admin Consent)	120
Azure CCKM User Portal	120
Home	122
Azure Directory Context Switcher (Support for Guest User for AAD B2B Collaboration)	122
Accessing the CCKM Directory Context Switcher Window	123
Logging Out of User Directory	123
Switching to Another User Directory	123
Logging Out of All User Directories	123
Keys	124
Create a Key	126
Upload your Own Key (BYOK)	126
Create a Native Key	128
Delete a Key	129
Restore a Key	129
Delete a “Soft-deleted” Key	129
Recover a “Soft-delete” Key (to Azure)	130
Purge a “Soft-delete” Key	130
Restore a “Soft-delete” Key (to Azure)	131
Rotate a Key	131
Manually Rotate a Key (Add a New Version of a Key)	132
Enable Autorotation of Key	133
Synchronize Keys	134
Enable Autorotation before Key Expiration	134
Export Keys	135
Backup Key	135
Delete a Backup Key	136
Reports	136
Schedules	137
Schedule Key Rotation	137

Set up a Key Rotation Schedule	137
Pause a Key Rotation Schedule	138
Delete a Key Rotation Schedule	138
Schedule Key Synchronization	138
Set up a Key Synchronization Schedule	139
Schedule Key Expiration	139
Set up a Key Expiration Schedule	140
Logs	140
Key Sources	141
Add (Generate) a Key	141
Add (Generate) a CipherTrust Manager Key	141
Add (Generate) a DSM Key	142
Add (Generate) an nShield Key	142
Delete a Key	143
Manage Keys in Key Sources List Page	143
Settings	143
Adding Recipients' Email Addresses (for Notifications)	143
Enabling Notifications	143
Configuring Syslog Server	144
Configure Remote Syslog Server	144
Delete Remote Syslog Server Configuration	145
Configuring Tenant Password and Revoking Admin Consent (for Service Principal)	145
Change Tenant Password	146
Revoke Admin Consent	146
About	146
Chapter 15: CCKM for Google Key Management	147
Overview	147
Google Cloud IAM	147
Member	147
Roles	148
Resource	149
About Google Cloud Keys	150
Prerequisites on Google Cloud	151
Access CCKM for Google Key Management	152
Google CCKM User Portal	152
Home	154
Keys	154
Create a Key	157
Create a Native Key	158

Upload your Own Key (BYOK)	159
Update Key	162
Update Key (Asymmetric Key)	162
Update Key (Symmetric Key)	163
Rotate a Key	163
Manually Rotate a Key (Symmetric Key)	164
Enable Autorotation of Key (Symmetric Key)	164
Manually Rotate a Key (Add New Key Version) (Asymmetric Key)	165
Enable Autorotation of Key (Add New Key Version) (Asymmetric Key)	166
View All Key Versions	167
Synchronize Keys	167
Disable or Enable Key Version	168
Disable Key Version	168
Enable Key Version	169
Schedule (or Cancel) Destruction of Key Version or All Key Versions	169
Schedule Destruction of Key Version	170
Schedule Destruction of All Key Versions	170
Cancel Schedule Destruction of Key Version	171
Cancel Schedule Destruction of All Key Versions	171
Download Public Key (Asymmetric Key Only)	171
Export Keys	171
Reports	172
Generate a Report	173
Delete a Report	173
Schedules	173
Schedule Key Rotation	174
Set up a Key Rotation Schedule	174
Pause a Key Rotation Schedule	175
Delete a Key Rotation Schedule	175
Schedule Key Synchronization	175
Set up a Key Synchronization Schedule	175
Logs	175
Key Sources	176
Add (Generate) a Key	176
Add (Generate) a CipherTrust Manager Key	176
Add (Generate) a DSM Key	177
Add (Generate) an nShield Key	177
Delete a Key	177
Manage Keys in Key Sources List Page	177
Settings	178

Adding Recipients' Email Addresses (for Notifications)	178
Enabling Notifications	178
Configuring Syslog Server	179
Configure Remote Syslog Server	179
Delete Remote Syslog Server Configuration	180
Adding Google Cloud Service Account to CCKM	180
Add Service Account	180
Update Service Account	181
Delete Service Account	181
About	181
 Chapter 16: CCKM for IBM Key Management	182
Overview	182
Types of Encryption Keys Supported within IBM HPCS and Key Protect	182
IBM Cloud Identity and Access Management	183
Action Permissions Mapping for CCKM and IBM HPCS/Key Protect	184
CCKM's Interaction with IBM HPCS and Key Protect	185
Prerequisites on IBM HPCS and Key Protect	185
Create an Instance of IBM HPCS or Key Protect in IBM Portal	186
Initialize your HPCS Service Instance	186
Generate a User API Key in IBM Portal	187
Generate Service ID API Key in IBM Portal	187
Create Access Group	187
Add User(s) to Access Group	188
Add Service ID(s) to Access Group	188
Assign Access to your Access Groups (for Users and Service IDs)	188
Assign Additional Access Policy to Service ID	189
Creating an IBM Credentials File in a JSON Format	190
Access CCKM for IBM	190
IBM CCKM User Portal	190
Home	191
Keys	191
Create a Key	194
Create a Native Key	195
Upload your Own Key (BYOK)	196
Delete a Key	197
Delete a Key	197
Deleting Keys Using Dual Authorization	198
Disable and Enable a Root Key	200
Disable a Root Key	200

Enable a Root Key	201
Restore a Key	201
Restore a Key (IBM HPCS)	202
Restore a Key (IBM Key Protect)	202
Update Key	202
Update Key for Dual Authorization of Key Deletion	202
Update Key to Enable Automatic Key Rotation Policy for Root Key	203
Rotate Key	203
Rotate a Key Manually	204
Enable Autorotation of Key	204
Synchronize Keys	204
Export Keys	205
View All Key Versions	205
Reports	205
IBM Cloud Activity Tracker with LogDNA Service	206
Provisioning IBM Cloud Activity Tracker with LogDNA Service	207
Schedules	207
Schedule Key Rotation	207
Delete a Key Rotation Schedule	208
Pause a Key Rotation Schedule	208
Schedule Key Synchronization	208
Set up a Key Rotation Schedule	209
Set up a Key Synchronization Schedule	209
Logs	210
Key Sources	210
Add (Generate) a Key	211
Add (Generate) a CipherTrust Manager Key	211
Add (Generate) a DSM Key	211
Add (Generate) an nShield Key	211
Delete a Key	212
Manage keys in Key Sources List Page	212
Settings	212
Adding Recipients' Email Addresses (for Notifications)	212
Enabling Notifications	212
Configuring Syslog Server	213
Configure Remote Syslog Server	213
Delete Remote Syslog Server Configuration	214
Enabling Reporting for IBM Keys	214
Enabling Reporting for IBM Key Activities Using IBM Cloud Activity Tracker and LogDNA	214
About	215

Chapter 17: CCKM for Salesforce Key Management	216
Overview	216
Prerequisites on Salesforce	216
Enable and Assign System Permissions for CCKM Login	217
Salesforce Cache-only Keys	218
Configuring Cache-only Keys	218
Configure CCKM with SSL Certificate Signed by Salesforce Root CA Certificate	219
Set up Named Credential in Salesforce	219
Access CCKM for Salesforce	220
Salesforce CCKM User Portal	220
Home	222
Tenant Secrets	222
Create a Tenant Secret/Key	225
Upload your Own Tenant Secret	225
Create a Tenant Secret/Key	226
Create a Cache-only Key	226
Verify Call-Out Connection for Cache-only Key	227
Destroy a Tenant Secret	227
Destroy a Cache-only Key	228
Edit a Cache-only Key	228
Import a Tenant Secret	229
Rotate a Tenant Secret	229
Manually Rotate a Tenant Secret (Add a New Version of a Tenant Secret)	229
Enable a Scheduled Tenant Secret Rotation	229
Synchronize Tenant Secrets	230
Export Tenant Secrets	230
Backup Key	231
Delete a Backup Key	231
Certificates	231
Synchronize Certificates	231
Reports	232
Delete a Report	232
Generate a Report	232
Schedules	232
Schedule Key Rotation	233
Set up a Key Rotation Schedule	233
Pause a Key Rotation Schedule	234
Delete a Key Rotation Schedule	234
Schedule Key Synchronization	234

Set up a Key Synchronization Schedule	234
Logs	235
Key Sources	235
Add (Generate) a Key	236
Add (Generate) a CipherTrust Manager Key	236
Add (Generate) a DSM Key	236
Add (Generate) an nShield Key	237
Delete a Key	237
Manage Keys in Key Sources List Page	237
Settings	237
Adding Recipients' Email Addresses (for Notifications)	237
Enabling Notifications	238
Configuring Syslog Server	238
Configure a Remote Syslog Server	239
Delete Remote Syslog Server Configuration	239
Configure Cache-only Key Endpoint	239
About	240
Chapter 18: Backup and Restore	241
Backup and Restore Recommendations	241
Backup and Restore for DSM	241
Backup and Restore for CipherTrust Manager	241
Backup and Restore for MongoDB	241
Chapter 19: Troubleshooting	242
Troubleshooting nShield Configurations	242
Troubleshooting Logs Files for Thales Support Team	243
Restrictions	243
Appendix A: CCKM CLI Commands	244
CCKM CLI Navigation	244
Maintenance Commands	245
Network Commands	245
System Commands	246
User Commands	246
Aplog Commands	247
Appendix B: How to Export Your DSM Server Certificate from DSM	248
Exporting your DSM Server Certificate	248

Appendix C: Configure SSL/TLS Certificate Chain of Trust for CCKM	249
Create and Configure SSL/TLS Certificate Chain of Trust	249
Appendix D: Sample Python Scripts for Authenticate and Call CCKM REST API	251
cckm-azure.py	252
cckm-aws.py	254
cckm-google.py	255
cckm-salesforce.py	256
Appendix E: Authenticate your REST API Client to CCKM REST APIs for Google Cloud Keys	258
Appendix F: General Guidelines for CCKM Appliance	259

Preface

This guide describes how to deploy the CipherTrust Cloud Key Manager (CCKM), and how to configure it to connect with various data sources and clouds. It also describes how to use the CCKM web interface.

Intended audience

This guide is for the system administrator who will install and configure the CCKM. This role is the "CCKM Admin". This guide describes how to use the CCKM GUI for managing AWS, Azure, Azure China, Google Cloud, IBM, Salesforce, Salesforce Government Cloud, and Salesforce Sandbox keys. These users, who must have access credentials to their organizations' cloud accounts, are referred to as "Cloud administrators."

Note

In this document, Salesforce Government Cloud Plus is also referred to as Salesforce Government and Salesforce Government Cloud.

Document Conventions

The document conventions describe common typographical conventions and important notice and warning formats used in Thales technical publications.

Typographical Conventions

This section lists the common typographical conventions for Thales technical publications.

Table 3-1: Typographical Conventions

Convention	Usage	Example
bold regular font	GUI labels and options	Click the System tab and select General Preferences .
<i>bold italic monospaced font</i>	Variables or text to be replaced	https://< <i>Token Server name</i> >/admin/ Enter password: < <i>Password</i> >
<i>regular monospaced font</i>	- Commands and code examples - XML examples	<code>session start iptarget=192.168.253.102</code>
<i>italic regular font</i>	GUI dialog box titles	The <i>General Preferences</i> window opens.
	File names, paths, and directories	<i>/usr/bin/</i>
	Emphasis	<i>Do not</i> resize the page.
	New terminology	<i>Key Management Interoperability Protocol (KMIP)</i>
	Document titles	See <i>CipherTrust Cloud Key Manager Installation & Configuration Guide</i> for information about .
quotes	- File extensions - Attribute values - Terms used in special senses	<code>.js</code> , <code>.ext</code> <code>"true"</code> <code>"false"</code> , <code>"0"</code> <code>"1+1"</code> hot standby failover

Notes, Tips, Cautions, and Warnings

Notes, tips, cautions, and warning statements may be used in this document.

A Note provides guidance or a recommendation, emphasizes important information, or provides a reference to related information. For example:

Note

It is recommended to keep tokenization keys separate from the other encryption/decryption keys.

A tip is used to highlight information that helps you complete a task more efficiently, such as a best practice or an alternate method of performing the task.

Tip

You can also use Ctrl+C to copy and Ctrl+P to paste.

Caution statements are used to alert you to important information that may help prevent unexpected results or data loss. For example:



CAUTION

Make a note of this passphrase. If you lose it, the card will be unusable.

A warning statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data. For example:



WARNING

Do not delete keys without first backing them up. All data that has been encrypted with deleted keys cannot be restored or accessed once the keys are gone.

Hardware-Related Warnings

The following warning statement is used to indicate the risk of electrostatic discharge of equipment:



ELECTROSTATIC DISCHARGE

If this warning label is affixed to any part of the equipment, it indicates the risk of electrostatic damage to the module. To prevent equipment damage, follow suitable grounding techniques.

The following warning statement is used to indicate the risk of hazardous voltages of equipment:



HAZARDOUS VOLTAGES

The warnings in this section indicate voltages that could cause serious danger to personnel.

Sales and Support

If you encounter a problem while installing, registering, or operating this product, please refer to the documentation before contacting support. If you cannot resolve the issue, contact your supplier or Thales Customer Support.

Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

For support and troubleshooting issues:

- <https://supportportal.thalesgroup.com>
- (800) 545-6608

For Thales Sales:

- <https://cpl.thalesgroup.com/encryption/contact-us>
- CPL_Sales_AMS_TG@thalesgroup.com
- (888) 267-3732

Chapter 1: Overview

CCKM Components	20
CCKM Functionality	22

The CipherTrust Cloud Key Manager (CCKM) centralizes the management of the life cycle of cloud keys in multi-cloud, hybrid, and enterprise environments. CCKM addresses the needs of these environments for encrypting data in the cloud while retaining custodianship of encryption keys to comply with data security mandates in cloud storage environments. For the clouds (or cloud services) CCKM supports, CCKM enables enterprises who use these services to bring their own keys to those clouds, back up keys on premise, destroy cloud keys in specific situations, and manage the life cycle of cloud keys. CCKM provides a web GUI for all functions.

Note

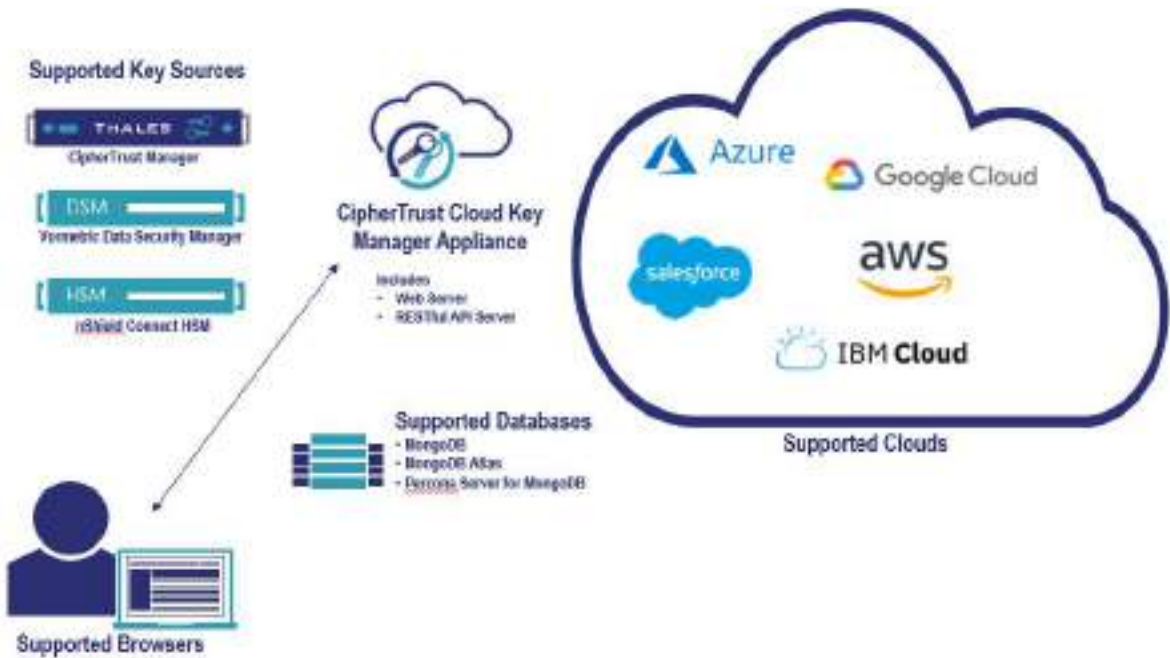
In this document, Salesforce Government Cloud Plus is also referred to as Salesforce Government and Salesforce Government Cloud.

CCKM Components

The CCKM solution comprises of the following components (see [Figure 1-1: "CCKM Solution: CCKM and Components" below](#)):

- CCKM with Web UI pages including an administrator portal (for CCKM administrators) and a user portal (for cloud administrators or cloud key administrative users)
- At least one of the supported clouds (you can use all clouds)
- A supported trusted key source
- MongoDB, MongoDB Atlas, or Percona Server for MongoDB as a database
- A supported Internet browser

Figure 1-1: CCKM Solution: CCKM and Components



The product is delivered as a virtual appliance that can be installed on any one of the supported deployment methods.

Supported Clouds

CCKM supports the following clouds:

- Amazon Web Services (AWS)
- Google Cloud
- IBM
- Microsoft Azure (Azure and Azure China)
- Salesforce (Salesforce, Salesforce Government Cloud, and Salesforce Sandbox)

Supported Key Sources

CCKM uses trusted key sources for the encryption keys employed within the supported clouds. Certain trusted key sources also store the clouds' backup keys. Currently, the supported key sources include:

- Vormetric Data Security Manager (DSM): The DSM supports all of the clouds that CCKM supports. The DSM keys and the backup keys from the supported clouds are stored in DSM.
- CipherTrust Manager: CipherTrust Manager supports all clouds that CCKM supports. The CipherTrust Manager keys and the backup keys from the supported clouds are stored in CipherTrust Manager.

Note

SafeNet Next Generation KeySecure has been rebranded as CipherTrust Manager. Throughout this guide, CipherTrust Manager is employed to reflect the new name of this product.

- nShield Connect hardware security module (HSM): nShield Connect supports the AWS and Azure clouds. The nShield Connect keys and the backup keys from the clouds are stored in MongoDB, the supported CCKM database.

Note

Concurrent use of the CipherTrust Manager, DSM, and nShield as key sources is not supported. Configure only **one** of these supported key sources as a key source.

Supported Databases

CCKM supports MongoDB, MongoDB Atlas, and Percona Server for MongoDB, a global cloud database service, to store key metadata, and CCKM system and user configuration information. Encrypted communication using TLS between CCKM and MongoDB (MongoDB, MongoDB Atlas, or Percona Server for MongoDB) is available. Furthermore, CCKM supports MongoDB Atlas on Google Cloud, as well as on Microsoft Azure.

Notes

- Any general references made to “MongoDB” in this guide applies to any one of the supported MongoDBs (MongoDB, MongoDB Atlas, or Percona Server for MongoDB) unless otherwise noted.
- Configure only **one** of the supported MongoDB databases.
- For the supported versions of MongoDB, MongoDB Atlas, and Percona Server for MongoDB, see [Chapter 4: "How It Works: Installation and Configuration Flow" on page 35](#).
- The MongoDB installation file includes a community (free) edition of MongoDB that can be used for evaluation and proof-of-concept (PoC) purposes. For a production setup, an external MongoDB in replica set configuration or a MongoDB Atlas configuration is recommended.

Supported Deployment Methods

CCKM is delivered as a virtual appliance that can be installed in one of the following methods:

- On-premises by deploying an .ova file
- In Amazon Web Services by deploying an Amazon Machine Image (AMI)
- By instantiating CCKM from the Azure Market place by deploying a Virtual Hard Disk (VHD) image
- In a Microsoft Hyper-V environment by deploying a VHD image

Supported Internet Browsers

The CCKM supports the following Internet browsers:

- Chrome 51.0.2704 (64-bit) or later
- Firefox 45.0 or later
- Internet Explorer 11 or later
- Microsoft Edge 89.0.774.45 or later

CCKM Functionality

CCKM offers similar functionality for each of the supported clouds within the supported deployment scenarios:

- **Life cycle management** of keys, key versions, and attributes:
 - View Keys
 - Update Keys
 - Upload Keys
 - Rotate Keys
 - Delete Keys
- **Disaster recovery** of keys
 - Backup Keys
 - Restore Keys
- **Hybrid key management**
 - On-premise Keys storage
 - Management of both keys originating from trusted key sources and cloud-provider-sourced keys
 - Key synchronization

- **Compliance Management**

- Key storage within on premise FIPS 140-2 Level 3 certification
- Key storage in public or private clouds, inaccessible to clouds with FIPS 140-2 Level 1

- **Certificate handling (Salesforce)**

- View certificates
- Synchronize certificates
- Use certificates to wrap keys for secure key upload

- **Key visibility reporting**

- Combined Key Activity Reconciliation Report
Track and reconcile key activities between CCKM and a cloud
- Key Activity Report
Insight into who, what and when keys were accessed, created, deleted across PaaS
- Key Aging Report
Insight into who owns the keys and when keys expire
- Key Service Usage Report
Insight into applications consuming the keys across IaaS/PaaS/SaaS

Note: This report is not available for Salesforce, Salesforce Government, and Salesforce Sandbox.

- Cloud Key Manager User Action Report
Insight into aggregated user activities, such as delete or create keys
- Connection to a remote syslog server for forwarding of CCKM activity logging. Syslog messages may also be directed to a SIEM system that accepts syslog messages. Syslog messages may also be directed to a Splunk server.

Chapter 2: VM Specifications for CCKM Components

- With Local MongoDB bundled with CCKM Appliance (Recommended for PoCs Only) 24
- With External MongoDB (Recommended for Production) 24
- With MongoDB Atlas (Recommended for Production) 26

This chapter provides the minimum requirements for the CCKM and MongoDB virtual machines (VMs) when CCKM is running bundled with a local MongoDB, with external MongoDB node(s), or MongoDB Atlas.

With Local MongoDB bundled with CCKM Appliance (Recommended for PoCs Only)

PoC – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory
- 100 GB Hard disk (Thin)

Development/Staging – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory
- 100 GB Hard disk (Thin)

Production – Hardware Recommendation

CCKM VM minimum requirements:

- 4 CPUs
- 8 GB Memory (Use 16 GB if API driven high frequency workflows are needed)
- 250 GB Hard disk (Thin)

With External MongoDB (Recommended for Production)

Note

If you are using Percona Server for MongoDB (single node or replica set), this section applies to you.

PoC – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory (Use 8 GB if API driven high frequency workflows are needed to be tested)
- 100 GB Hard disk (Thin)

MongoDB VM minimum requirements:

- 2 CPUs
- 4 GB Memory (Use 8 GB if API driven high frequency workflows are needed to be tested)
- 250 GB Hard disk (Thin)

Development/Staging – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory (Use 8 GB if API driven high frequency workflows are needed to be tested)
- 100 GB Hard disk (Thin)

MongoDB VM minimum requirements:

- 2 CPUs
- 4 GB Memory (Use 8 GB if API driven high frequency workflows are needed to be tested)
- 250 GB Hard disk (Thin)

Production – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 8 GB Memory (Use 16 GB, if API driven high frequency workflows are needed)
- 100 GB Hard disk (Thin)

External MongoDB VM minimum requirements for each node:

- 4 CPUs
- 8 GB Memory (Use 16 GB, if API driven high frequency workflows are needed)
- 250 GB Hard disk (Thin)

With MongoDB Atlas (Recommended for Production)

PoC – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory
- 100 GB Hard disk (Thin)

Development/Staging – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory
- 100 GB Hard disk (Thin)

Production – Hardware Recommendation

CCKM VM minimum requirements:

- 4 CPUs
- 8 GB Memory (Use 16 GB if API driven high frequency workflows are needed)
- 250 GB Hard disk (Thin)

Chapter 3: About CCKM Admin and User Portals

- CCKM Administration Privileges 27
- Accessing CCKM Admin Portal (as CCKM Admin) 28
- Accessing CCKM User Portal from CCKM Admin Portal 28
- Accessing CCKM User Portal (as CCKM User) 31
- Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI 32
- User Handling within CCKM User Portal (User Login Info) 33

CCKM provides Web UI pages for a CCKM Admin Portal for CCKM administrators and a CCKM User Portal for CCKM users (also known as cloud administrators). The chapter provides information about these portals, administration privileges, support for multi-accounts per cloud, and the handling of user (cloud administrator) login.

For information about the available menu options used to configure and manage the CCKM from the CCKM Admin Portal and, see [Chapter 12: "About the Admin Portal" on page 69](#).

Note
In this document, Salesforce Government Cloud Plus is also referred to as Salesforce Government and Salesforce Government Cloud.

CCKM Administration Privileges

From the CCKM Admin Portal, a CCKM administrator can add and configure clouds, monitor CCKM logs to review user activities in CCKM, upload your CCKM license(s) and view all of the licenses for the configured clouds, and manage and configure CCKM settings. Unless the CCKM administrator also has user (cloud administrator) login credentials for Azure, AWS Key Management, Google Cloud, IBM or Salesforce, he or she will not be able to access the CCKM User Portal pages.

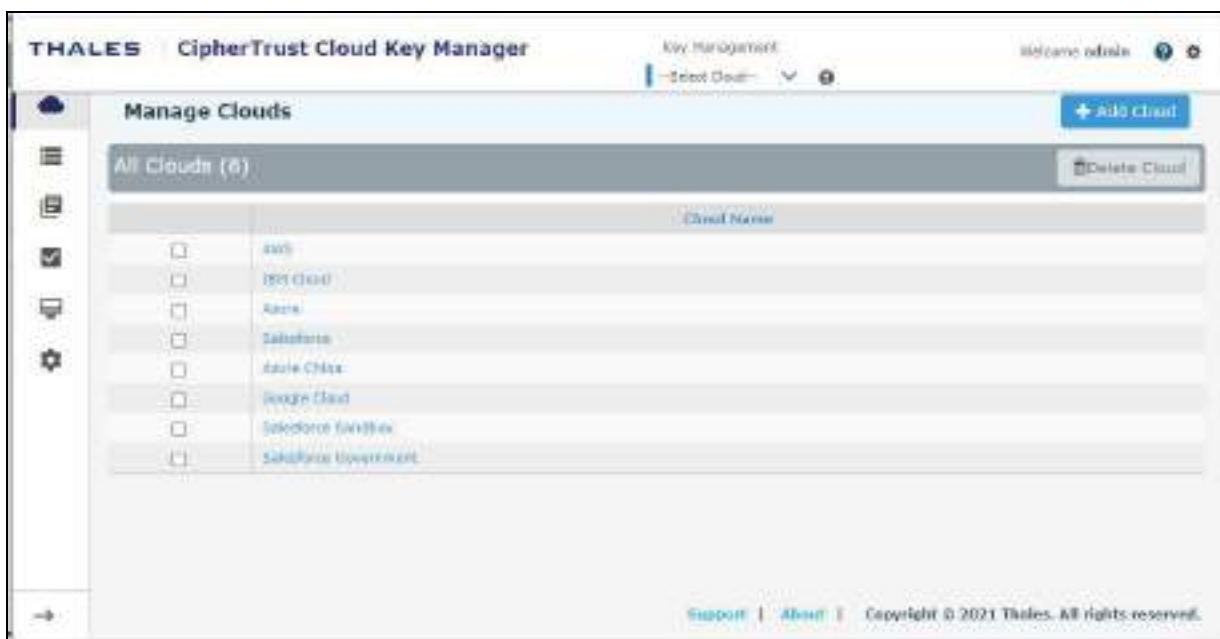
To access the CCKM User Portal, a CCKM user must have valid login credentials for the organization's designated clouds. No specific "CCKM administrator" is created. Administrators have access from the CCKM User Portal to all clouds for which they have *valid* credentials.

Accessing CCKM Admin Portal (as CCKM Admin)

1. Enter “https://<IP address of CCKM instance>” in a browser. The CCKM Admin Login page is displayed:



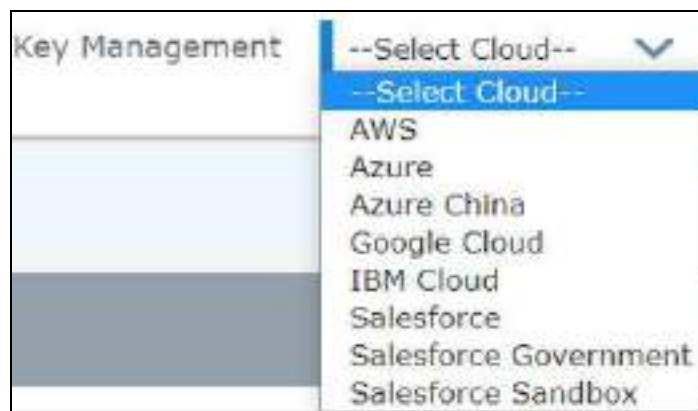
2. Log in as a CCKM administrator using your administrator credentials. The CCKM Admin Portal is displayed:



Accessing CCKM User Portal from CCKM Admin Portal

From the CCKM Admin Portal, you can access the CCKM User Portal and CCKM Home page. However, you are required to enter the user login credentials for the given Key Management service to access the CCKM User Portal.

From the CCKM Admin Portal, select a *configured* cloud from the Key Management drop-down list at the top of the page.



Note

Only the cloud(s) you have configured display in the **Key Management** drop-down list.

The login page for the selected Key Management service is displayed.

Azure

The Microsoft login page opens at <https://login.microsoftonline.com/>

Enter your Microsoft user login credentials or click on the saved user name. The Microsoft Login page is displayed. Enter your Microsoft password and click **Log In**.

AWS

If you select **AWS** as the Key Management service, then the CCKM user login page is displayed. From this page, select the AWS key management service and the login page for AWS displays. Click **Log into AWS**. The AWS Password Login page is displayed. Enter your AWS key credentials in the **AWS Access Key ID** and **AWS Secret Access Key** boxes and click **Log In**.

Google

If you select **Google** as the Key Management service, then the CCKM user login page is displayed. From this page, select the Google key management service. You are redirected to <https://accounts.google.com/>. Enter your Google user login credentials (or click on the saved user name) and then enter your Google password. Click **Log In**. At the consent page, click **Allow** to proceed with the login.

IBM

If you select **IBM** as the Key Management service, then the CCKM user login page is displayed. From this page, select the IBM key management service and the login page for IBM displays. Click **Log into IBM**. The IBM Password Login page is displayed. Under **Login Type**, select **User Name** or **Service ID** depending on which type of login you wish to use. Under **Network Endpoint Mode**, select **Public** or **Private** depending on whether you wish to use a public or a private endpoint to access the IBM cloud services (HPCS or Key Protect) from CCKM. Enter your IBM key credentials in the **IBM Cloud User Name** and **IBM Cloud API Key** boxes and click **Log In**. If you selected **User Name** as the login type, enter your IBM username and IBM Cloud API key for your IBM key credentials. If you selected **Service ID** as the login type, enter your IBM service ID and IBM service ID API Key. You can also upload your IBM key credentials file under the **Load Credentials from IBM credentials file** by choosing the file and then clicking **Log In**.

Salesforce

The Salesforce login page opens at <https://login.salesforce.com/>.

The Salesforce Government login page URL is unique. This login page opens at:

https://{Salesforce_Government_My_Domain_URL} where "Salesforce_Government_My_Domain_URL" is your unique URL.

The Salesforce Sandbox login page opens at <https://test.salesforce.com/>

Enter your Salesforce credentials or click on the saved user name. The Salesforce login page is displayed. Enter your Salesforce password and click **Log In**.

Note

The first time you access the CCKM, the keys are synchronized and a progress page is displayed. Once synchronization is complete, click **Continue**. The **CCKM Home** page is displayed within the CCKM User Portal.



Accessing CCKM User Portal (as CCKM User)

To access the CCKM User Portal and CCKM Home page as a CCKM user:

1. Enter “https://<CCKM IP address>:8443/kmaas” in a browser. The CCKM user login page is displayed.



2. Select the installed Key Management service you wish to use. The login page for the selected Key Management service is displayed.
3. Enter your user login credentials for the given Key Management service to log into CCKM. The **CCKM Home** page is displayed within the CCKM User Portal.



Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI

CCKM supports login and key management for multiple accounts or multiple users per cloud through a single CCKM UI (Multi-account Login/Logout) and browser window.

Note

CCKM does not currently support login and key management for multiple accounts or multiple users per cloud for the Salesforce Government cloud.

The following **Multi-account Login/Logout** window reflects the Azure account (varies depending on the cloud account into which you are currently logged):

User name	Switch to	User Logout
[redacted]@azurevormetric.onmicrosoft.com	☐	☐
[redacted]@azurevormetric.onmicrosoft.com	☐	☐

Show 10 entries « 1 » Showing 1 to 2 of 2

Cancel Add Tenant Logout of Azure Logout All Services

You can perform the following using the **Multi-account Login/Logout** window.

- Logon to multiple accounts for a specific cloud (AWS/Azure/Google/IBM/Salesforce)
- Logout of specific or all accounts within one cloud or all clouds
- Perform key management for multiple accounts from your current CCKM browser window

Accessing the Multi-account Login/Logout Window

To access the **Multi-account Login/Logout** window for a specific cloud, you must have at least one user already logged on to an account. Once you are logged onto an account, from the service **Dashboard**, click on the cloud icon (Context Switcher) at the top of right of the page. The **Multi-account Login/Logout** window displays. The following sections provide instructions on how to use this window.

Switching to Another User Account (Already Logged On)

(Applicable for a given cloud on to which you are currently logged) To access another user account, which is already logged on, select the **Switch** to button next to the user name (account) for which to log on. You are then switched to that account with the user name displayed at the top of the **CCKM Home** page.

Logging on to Another User Account

(Applicable for a given cloud on to which you are currently logged) To log on to another user account, click one of following buttons depending on the service you are currently logged and then enter the user credentials in the login dialog boxes of the cloud:

- **Add Tenant** (applicable to Azure, and Azure China)
- **Add Organization** (applicable to Google Cloud)
- **Add OrgID** (applicable to Salesforce and Salesforce Sandbox)
- **Add Account** (applicable to AWS and IBM)

Once the user authentication is completed, you are logged on to the given user account and the username is displayed at the top of the CCKM Home page.

Logging Out of a User Account

(Applicable for a given cloud on to which you are currently logged) To logout of a specific user account, select the **User Logout** button next to the user name (account) for which to log out. The account is then logged out.

Logging Out of All User Accounts from a Specific Cloud Service or All Cloud Services

To logout of all of the user accounts for a specific cloud, click one of the following depending on the service to which you are currently logged:

- **Logout of Azure (or Logout of Azure China)**
- **Logout of AWS**
- **Logout of Google**
- **Logout of IBM**
- **Logout of Salesforce (Logout of Salesforce Sandbox)**

To logout of all clouds, click the **Logout All Services** button. All services are then logged out.

User Handling within CCKM User Portal (User Login Info)

This section describes how CCKM handles user authentication, user authorization, and user audits within the CCKM User Portal for each the clouds.

User Authentication

For AWS users, the user provides their AWS long-term credentials (for example, access key ID and secret access key) to CCKM. CCKM uses these credentials to obtain a one-time, temporary credential for the user session. *CCKM does not save a user's long-term credentials.*

For Azure, Google Cloud, and Salesforce users, the CCKM itself does not maintain user credentials. It uses OAuth2 to authenticate users.

During login, users are redirected to the Azure, Google Cloud, or Salesforce login page (depending on the cloud you accessing) to enter their respective username/password combinations. Once authenticated, the CCKM User Portal

obtains an OAuth2 access token from the cloud and then maintains a user session between CCKM and the user's browser. While the user session is active, CCKM uses the access token to call the given cloud's REST APIs, and will allow the logged-in user to access and manage keys within the given cloud per the granted permission.

For IBM users, the user provides their IBM long-term credentials (for example, IBM Username and API Key) to CCKM. CCKM uses these credentials to obtain a one-time, temporary credential for the user session. CCKM does not save a user's long-term credentials.

User Authorization

In AWS, user authorization defines what a user can do with AWS KMS. For a user to manage AWS CMKs, he or she must have proper KMS permissions. In addition, to use CCKM, the user should have IAM GetUser permission.

In Azure, user authorization defines what a user can do on Azure keys. For example, whether they can view, upload, update, delete, or restore keys in a key vault. This is managed within Azure Key Vault's Access Control (IAM) and Access Policies. To allow a user to manage keys in a key vault, the key vault owner must give the user "Key Vault Contributor" role in Access Control, and appropriate permissions as needed on keys in **Access Policies**. If CCKM is used as a service principal, then select Azure role-based access control permission model.

In Google Cloud, user authorization defines what a user can do on Google Cloud keys. For example, whether they can view, upload, update, rotate, or schedule destroy keys in Google Cloud KMS. This is managed within Google Cloud's Identity and Access Management (IAM). You will need to have at least the Viewer role on the Key Rings that you wish to manage and the appropriate permissions as needed on the keys and key versions. For more information, see [Chapter 15: "CCKM for Google Key Management" on page 147](#).

In IBM, user authorization defines what a user can do within IBM Cloud HPCS and IBM Key Protect. For a user to manage an IBM HPCS or Key Protect instance, he or she must have proper permissions for these two supported IBM key management services. In addition, to use CCKM, the user should have IAM GetUser permission and keys level permission.

In Salesforce, user authorization defines whether the user has "Manage Encryption Keys" permission or not. Only users who have **Manage Encryption Keys** permission can manage Tenant Secrets.

User Audit

User operations (within the CCKM User Portal), including login and logout (successful or failed) are captured in CCKM audit logs. Currently, there are no access policies on CCKM audit logs and so any logged-in user for a specific organization and service can view all audit logs. These audit logs include operations made by other users in the past, within the scope of that organization.

Chapter 4: How It Works: Installation and Configuration Flow

Step 1: Install and Configure MongoDB	35
Step 2: Install and Configure Key Source	35
Step 3: Setup to Connect to Cloud(s)	36
Step 4: Install CCKM in One of the Supported Deployment Methods	37
Step 5: Perform Required Steps for the Initial Configurations of CCKM Admin Portal	37

This chapter provides a summary of the steps to take to install and configure the CCKM solution (CCKM and its components). Use the steps in this section to guide you in installing and configuring the CCKM solution. Refer to the related sections for details about a given step.

Note

In this document, Salesforce Government Cloud Plus is also referred to as Salesforce Government and Salesforce Government Cloud.

Step 1: Install and Configure MongoDB

Choose **one** of the following and perform the pre-setup steps within MongoDB:

- **Embedded (free) MongoDB** No pre-setup required
- **External MongoDB (single or replica)**
Supported versions are 4.4.6 and later.

Note: This version of CCKM was tested with versions 4.4.6 and 5.0.3 of MongoDB.

- **MongoDB Atlas Supported versions are 4.4.12 and 5.0.6.**
- **Percona Server for MongoDB (single or replica) Supported version is 4.2.8-8**

Note

CCKM communicates with MongoDB through port 27017.

For more information, see [Chapter 5: "Set Up MongoDB" on page 38](#).

Step 2: Install and Configure Key Source

Perform **one** of the following steps depending on which key source you are using.

Install and Configure CipherTrust Manager as your Key Source

If you plan to use the CipherTrust Manager as your key source, you must launch and configure it prior to configuring CCKM to integrate with the CipherTrust Manager. Use version 2.3.0 or later.

Note

By default, CCKM communicates with the CipherTrust Manager through port 443. You can change this setting for the port number of the CipherTrust Manager web interface. For more information, see ["Configure CipherTrust Manager as Key Source" on page 74](#).

Note

When using version **2.6.0** of CipherTrust Manager as a key source for CCKM version **1.10.0** and earlier versions, you are required to either upgrade CCKM to version **1.10.1** or change the CipherTrust Manager property of `ALLOW_UNKNOWN_FIELDS` to `true`. By default, this property is set to `false`.

For more information, see ["Install and Configure CipherTrust Manager as Key Source" on page 42](#).

Install and Configure DSM as your Key Source

If you plan to use the DSM as your key source, then you must install and configure it prior to configuring the CCKM to integrate with the DSM. Install version **6.4.3.17030** or later.

Note

CCKM version **1.10.0** supports IBM cloud key management with DSM **6.4.4** and above.

Note

CCKM communicates with the DSM through ports 443 and 8445.

For more information, see ["Install and Configure DSM as Key Source" on page 43](#).

Install and Configure nShield as your Key Source

If you plan to use nShield Connect as your key source, then you must install and configure it prior to configuring the CCKM to integrate with the nShield Connect.

nShield Connect supports the AWS and Azure (Azure and Azure China) clouds. Install version **12.40.2** or later of nShield Connect.

With the use of nShield Connect, the keys are stored as blobs within the MongoDB.

Note

CCKM communicates with nShield Connect through port 9004.

For more information, see ["Install and Configure nShield Connect as Key Source " on page 43](#).

Step 3: Setup to Connect to Cloud(s)

Perform one or more of the following steps depending on which cloud(s) you are using.

Connect to Azure

If using Azure, create an app as a doorway between CCKM and cloud(s).

For more information, see [Chapter 7: "Register CCKM App in Azure Cloud" on page 45](#).

Connect to AWS

If using AWS, there is no requirement to create an app in AWS to connect to CCKM. AWS users should have the following:

- Access key ID and secret access key
- IAM GetUser permission for self (you as a user)
- Proper KMS permissions

Connect to Google Cloud

If using Google Cloud, perform all of the prerequisites to set up to connect to CCKM.

For more information, see ["Prerequisites on Google Cloud" on page 151](#).

Connect to IBM Cloud

If using IBM Cloud, there is no requirement to create an app in IBM to connect to CCKM.

IBM Cloud users should have the following:

- IBM Username and API Key
- IAM GetUser permission for self (you as a user)
- Proper IBM Cloud Hyper Protect Crypto Services (HPCS) permissions, if you are using this IBM key management service.
- Proper IBM Key Protect permissions, if you are using this IBM key management service
- Proper IBM Cloud Activity Tracker and LogDNA permissions

Note

The port that CCKM uses to communicate with HPCS varies. To view the port number for a given HPCS instance, view the endpoint URL, which includes the port number, for the instance. Within the Navigation Menu of the IBM portal, select **Resource List**. Under **Resource list > Services and software**, select the HPCS instance for which to view the port number and select **Overview**. Under **Key management endpoint URL**, find the port number at the end of the URL.

Note

CCKM communicates with Key Protect through port 443.

For more information, see ["Prerequisites on IBM HPCS and Key Protect" on page 185](#).

Connect to Salesforce

If using Salesforce, Salesforce Government, or Salesforce Sandbox, create a connected app in Salesforce.

For more information, see [Chapter 9: "Create a Connected CCKM App in Salesforce" on page 54](#).

Step 4: Install CCKM in One of the Supported Deployment Methods

See [Chapter 10: "Install CCKM" on page 57](#) for more information.

Step 5: Perform Required Steps for the Initial Configurations of CCKM Admin Portal

See [Chapter 11: "Initial Configurations within CCKM Admin Portal " on page 66](#) for more information.

Chapter 5: Set Up MongoDB

Scenario 1: Use Built-in MongoDB for Evaluation and PoC Purposes	38
Scenario 2: Configure an Existing (External) MongoDB (Single Node or Replica Set)	38
Scenario 3: Configure MongoDB Atlas	39

The CCKM uses MongoDB to store key metadata, and CCKM system and user configuration information. The supported MongoDB configurations are described in this chapter.

Notes

- CCKM supports MongoDB, MongoDB Atlas, and Percona Server for MongoDB.
- Configure only **one** of the supported MongoDB databases.
- Be sure to back up your MongoDB database(s) on a regular basis. For more information on how to back up MongoDB, see ["Backup and Restore for MongoDB" on page 241](#).
- Although the CCKM includes a version of MongoDB, the software is supported by MongoDB. Contact the MongoDB Support team with any inquiries or issues.

Scenario 1: Use Built-in MongoDB for Evaluation and PoC Purposes

No pre-setup of MongoDB is required for the use of the built-in MongoDB within CCKM. If you are using the built-in MongoDB database server within CCKM (for evaluation and PoC purposes only), select to use the local database option from the **Database** tab within the **Settings** page of the CCKM Admin Portal.

Scenario 2: Configure an Existing (External) MongoDB (Single Node or Replica Set)

Note

If you are using Percona Server for MongoDB (single node or replica set), this section applies to you.

To configure an existing external MongoDB instance (single node or replica set):

- (Optional) Enable TLS for encrypted communication between MongoDB and CCKM
- (Required) Create one MongoDB database named *kmaas* and the user named *kmaas_user* along with the user passwords and database roles.

Each of these configurations is described in this section.

(Optional) Enable TLS

If you choose to enable TLS in the MongoDB server, add TLS-related parameters to the `mongod.conf` file. Here is a sample of the TLS-related parameters to add to the file:

```
tls:
  mode: requireTLS
  certificateKeyFile: /path to MongoDB .pem file
  FIPSMode: true
  disabledProtocols: TLS1_0,TLS1_1
```

```
security:
  keyFile: /path to keyFile for internal membership authentication
  authorization: enabled
```

For the MongoDB replica nodes, add these TLS-related parameters to each of the associated `mongod.conf` files.

Create a Database (*kmaas*) and User (*kmaas_user*)

Note

If the MongoDB server is TLS-enabled, then ensure that the MongoDB client has the root certificate authority file of MongoDB server as this file is required to perform the TLS handshake between the MongoDB server and CCKM.

In this section, create a database named *kmaas* and the MongoDB user of *kmaas_user* (along with the associated password and roles). CCKM uses the username and password to access the database.

To create a database and user:

1. Access the MongoDB server using the MongoDB client.
2. Create the *kmaas* database and a new user (*kmaas_user*) for this database by running the following commands (replacing `<mypassword>` with the password you choose):

```
use kmaas
db.createUser( { user: "kmaas_user", pwd: "<mypassword>", roles: [ "dbOwner" ] } )
```

Scenario 3: Configure MongoDB Atlas

Before configuring MongoDB Atlas within CCKM, configure MongoDB Atlas. Perform these configurations as described in this section:

- Whitelist Devices Allowed to Connect to MongoDB Atlas
- (Optional) Add MongoDB Role (*kmaas_role*)
- Add MongoDB User (*kmaas_user*)
- Create a New Cluster

If using a MongoDB Atlas cluster, you will also be required to obtain a connection string from MongoDB Atlas that is used to connect CCKM to your MongoDB Atlas cluster using a MongoDB driver.

The configuration steps in this section highlight the *optional* and *required* configurations for MongoDB Atlas to communicate with CCKM. It is assumed you are familiar with configuring MongoDB Atlas. For detailed information on how to configure these steps in MongoDB Atlas, refer to the MongoDB Atlas documentation.

Whitelist Devices Allowed to Connect to MongoDB Atlas

To whitelist the IP addresses of the devices allowed to connect to MongoDB Atlas:

1. In **SECURITY** in the left navigation bar, select **Network Access**.
2. Within **Network Access**, click **ADD IP ADDRESS** to add an IP address to the whitelist. The **Add Whitelist Entry** dialog box displays. In the **Whitelist Entry** box, enter the IP address.
3. Click **Confirm**.
4. Repeat steps 2 through 3 for each device to the whitelist.

(Optional) Add MongoDB (Custom) Role (kmaas_role)

Adding the customized MongoDB user role of `kmaas_role` to MongoDB Atlas is optional. When adding your MongoDB users to MongoDB Atlas, you can associate your MongoDB user to these custom roles. However, you can also associate the users with the pre-defined role of Read and write to any database. This role allows a user associated with this role to read and write to any database including the CCKM databases. If CCKM is the only application you are using with MongoDB Atlas, then assigning the Read and write to any database role to your users is sufficient. However, if you are using more than one application with MongoDB Atlas, then associating customized roles to your users is recommended.

To add the MongoDB user role of `kmaas_role` to MongoDB Atlas and associate actions and roles with a database:

1. In **SECURITY** in the left navigation bar, select **Database Access**.
2. In **MongoDB Roles**, click **ADD NEW CUSTOM ROLE**. The **Add Custom Role** dialog box displays.
3. In the **Custom Role Name** box, enter `kmaas_role`.
4. In the **Action** or **Role** box, select **Database Actions and Roles**.
5. In the **Database** box, enter `kmaas` as the name of the database to which to associate this role.
6. Click **Add Custom Role**.

Add MongoDB User (kmaas_user)

Perform one or the other of these configurations depending on which role you are assigning to your users.

To add the MongoDB user of `kmaas_user` to MongoDB Atlas and associate this user with the pre-defined role of **Read and write to any database**:

1. In **SECURITY** in the left navigation bar, select **Database Access**.
2. In **MongoDB Users**, click **ADD NEW USER**. The **Add New User** dialog box displays.
3. In the **Username** box, enter `kmaas_user`.
4. In the **Password** box, enter the password for the `kmaas_user` user.

Note: Ensure the password does not contain the escape character of “\”.

5. In **User Privileges**, click **Read and write to any database** to associate this user with this role.
6. Click **Add User**.

To add the MongoDB user of `kmaas_user` to MongoDB Atlas and associate this user with the customized role of `kmaas_role`:

1. In **SECURITY** in the left navigation bar, select **Database Access**.
2. In **MongoDB Users**, click **ADD NEW USER**. The **Add New User** dialog box displays.
3. In the **Username** box, enter `kmaas_user`.
4. In the **Password** box, enter the password for the `kmaas_user` user.
5. In **User Privileges**, click **Select Custom Role** and select `kmaas_role` to associate this user with this role.
6. Click **Add User**.

Create a New Cluster

In this step, you will create a new cluster in MongoDB Atlas for the CCKM application. Configuring a new cluster includes setting global cluster configuration, cloud provider and region, cluster tier, and cluster name.

After you create a new cluster for CCKM, one database set will be added to MongoDB Atlas. Thereafter, you are required to initialize the kmaas database by adding the (admin) user document to this database.

To create a new cluster in MongoDB Atlas for CCKM:

1. Under **ATLAS** in the left navigation bar, select **Clusters**.
2. In **Clusters**, click **Build a New Cluster**. The **Create a New Cluster** page displays.
3. In **Cloud Provider & Region**, select the provider and region you wish to use for the cluster.
4. In **Cluster Tier**, select the tier you wish to use for the cluster.
5. In **Additional Settings > for MongoDB 4.0, Backup > Select a Version**, select MongoDB 4.0.
6. (Optional. The display of this setting depends on your account settings.) In the **Turn on Backup (M2 and up)**, set this setting to **YES** and select **Continuous**.
7. In **More Configuration Options > Set Minimum TLS Protocol Version**, select TLS 1.2 and above.
8. Within **Cluster Name** box, enter the name of the new cluster.
9. Click **Create Cluster**.

Obtain Connection String for Connection Using a MongoDB Driver

In this step, you will obtain a connection string from MongoDB Atlas that is used to connect CCKM to your MongoDB Atlas cluster using a MongoDB driver. The connection string is made available to you based on the selected driver type and version. Copy this string and save it to a directory within your laptop or desktop. You will enter this connection string when configuring MongoDB Atlas settings in the Settings page > MongoDB tab of CCKM Admin Portal.

To obtain a collection string using a MongoDB driver:

1. In MongoDB Atlas, under **ATLAS** in the left navigation bar, select **Clusters**.
2. Find the cluster you created and click **Connect**.
3. In the **Choose a connection method**, select **Connect your application**.
4. In the **Choose your driver version**, select your driver and version from the dropdown menus.
 - a. Under **DRIVER**, select **Java**.
 - b. Under **VERSION**, select **4.3** or *later*.
5. In the **Add your connection string into your application code**, the **Connection String Only** tab displays the connection string to use to connect to your MongoDB Atlas cluster from CCKM. Click **Copy** to copy the connection string.
6. Save the connection string to a directory within your laptop or desktop.

Chapter 6: Install and Configure Key Source

Install and Configure CipherTrust Manager as Key Source	42
Install and Configure DSM as Key Source	43
Install and Configure nShield Connect as Key Source	43

This chapter provides information at a high-level about the installation and configuration of the key source you plan to use with the CCKM *prior* to installing and configuring CCKM. As part of these steps, you will *gather* information that is required to be entered in CCKM as part of the key source configuration. The following are the supported key sources:

- CipherTrust Manager
- DSM
- nShield Connect HSM

Note

Concurrent use of the CipherTrust Manager, DSM, and nShield as key sources is not supported. Configure only **one** of these supported key sources as a key source.

SafeNet Next Generation KeySecure has been rebranded as CipherTrust Manager. Throughout this guide, CipherTrust Manager is employed to reflect the new name of this product.

Install and Configure CipherTrust Manager as Key Source

Note

This documentation assumes that you are familiar with using the NextGen CipherTrust Manager. For information on how to launch and configure CipherTrust Manager, refer to the CipherTrust Manager Deployment Guide and CipherTrust Manager Administrator Guide (version 2.0.0 or later).

If you plan to use CipherTrust Manager as your key source, install and configure it *prior* to installing and configuring CCKM.

Note

You will enter the CipherTrust Manager account information you gather during the installation and configuration of CipherTrust Manager in CCKM (described in this section) when configuring this key source in the **CipherTrust Manager** tab of the **Key Sources** page of the CCKM Admin Portal as part of the initial CCKM configuration. (This initial configuration of CCKM takes place after you install CCKM and login into the CCKM Admin Portal for the first time.)

For any CipherTrust Manager you plan to use as key source for CCKM, the CipherTrust Manager administrator must perform the following:

1. Launch and configure CipherTrust Manager.
2. In CipherTrust Manager, create a CipherTrust Manager user or a LDAP user that has domain and system administrator privileges. Note the user name and password for this user.
3. Note the hostname or the IP address of the CipherTrust Manager during the configuration of CipherTrust Manager. This information is required when you configure the CipherTrust Manager settings in the **CipherTrust Manager** tab of the **Key Sources** page

4. Note the Connection String that will be used to connect CCKM to CipherTrust Manager. By default, the Connection String is "local_account". If an LDAP user is selected, the connection string is the LDAP domain name.
5. Download the server web certificate (in a Base64 format) of the CipherTrust Manager from the CipherTrust Manager server or browser.

Note: The common name of the server web certificate of CipherTrust Manager and its hostname must match. Otherwise, when you proceed to test the connection between the CCKM and CipherTrust Manager (when configuring CipherTrust Manager in the **CipherTrust Manager** tab within **Key Sources** page in the CCKM Admin Portal), the connection fails, and an error message displays.

Install and Configure DSM as Key Source

If you plan to use DSM as your key source, install and configure it prior to installing and configuring CCKM.

Note

You will enter the DSM account information you gather during the installation and configuration of DSM in CCKM (described in this section) when configuring this key source in the **DSM** tab of the **Key Sources** page of the CCKM Admin Portal as part of the initial CCKM configuration. (This initial configuration of CCKM takes place after you install CCKM and login into the CCKM Admin Portal for the first time.)

For any DSM you plan to use as key source for CCKM, a **DSM administrator** must perform the following:

1. Install and configure DSM. Refer to the latest version of the DSM Administration Guide.
2. When configuring DSM, create a DSM administrator account of type "All Admin" for CCKM.
3. Note the following information (as you will enter this information when configuring the DSM key source in the CCKM Admin Portal:
 - DSM fully-qualified host name
 - DSM CCKM administrator name and password for the "All Admin" account.
4. Download the DSM server certificate. For information on how to do this, see [Appendix B: "How to Export Your DSM Server Certificate from DSM" on page 248](#).

Install and Configure nShield Connect as Key Source

Note

This documentation assumes that you are familiar with using the nShield Connect HSM. This documentation also assumes that you have installed nShield Connect, and the associated RFS and Security World. For information on how to install an nShield Connect, and the associated RFS and Security World, refer to the nShield® Connect Installation Guide.

If you plan to use nShield Connect as your key source, install and configure it *prior* to installing and configuring CCKM.

Note

You will enter the nShield Connect account information you gather during the installation and configuration of nShield Connect in CCKM (described in this section) when configuring this key source in the **nShield** tab of the **Key Sources** page of the CCKM Admin Portal as part of the initial CCKM configuration. (This initial configuration of CCKM takes place after you install CCKM and login into the CCKM Admin Portal for the first time.)

For any nShield Connect you plan to use as key source for CCKM, the nShield Connect administrator must perform the following:

Note

To complete the configuration of nShield Connect, you are required to install CCKM and note its IP address or already know the IP address of CCKM. After obtaining the IP address of the CCKM, you can return to completing the nShield Connect configuration by adding the IP address of CCKM as an nShield client (last step).

1. Configure nShield Connect and note its IP address. You will add this IP address when you configure CCKM to connect to the nShield Connect.
2. Set up RFS and note its IP address. You will add this IP address when you configure CCKM to connect with nShield Connect.
3. Set up Security World.
4. In the nShield Connect, configure CCKM as an nShield client by specifying the IP address of CCKM. For information on how to configure the CCKM as an nShield client, refer to the nShield Connect User Guide for Unix.

Chapter 7: Register CCKM App in Azure Cloud

Registering CCKM App in Azure or Azure China Cloud 45

Register CCKM app in Azure **before** adding an Azure cloud in CCKM. CCKM provides support for Azure clouds (Azure and Azure China).

Registering CCKM App in Azure or Azure China Cloud

Before adding an Azure cloud in CCKM, you must register the CCKM App and assign required permissions in the Azure portal. If you are using the CCKM App as a service principal, then you must also assign the Azure role of "Contributor" to the service principal to grant the CCKM App access to Azure resources. Then depending on the type of app credential you plan to employ, either create a key (client secret) in the Azure portal or generate a certificate from CCKM, download it, and then upload it to Azure. This entire process generates the connection data needed to configure the Azure cloud.

Also before configuring an Azure cloud in CCKM, refer to the following related sections to review information about how Azure allows external applications, such as CCKM, to access and manage resources within Azure and the two types of app credentials employed when registering apps in Azure:

- ["Accessing and Managing Azure Resources" on page 117](#)
- ["Registering Apps in Azure" on page 118](#)

The steps to create the connection between CCKM and an Azure Key Vault are the same for Azure and Azure China.

Note

The UI within the Azure portal changes periodically. In regards to the documentation relating to Azure App Registrations, this documentation reflects the legacy App Registrations, which displays as App registrations (Legacy) in the Azure portal. If you are using the new Azure AD app registration, refer to the Azure online documentation for information regarding the equivalent steps (documented in this section) that you would perform in the new Azure AD app registration.

Note

In a production environment, Thales recommends that you **DO NOT** change the cloud configuration for Azure or Azure China by changing the setting for the **Use CCKM as Service Principal** check box.

Create an Azure Active Directory Application (Multi-tenanted) in Azure Portal

To create an Azure Active Directory application in the Azure portal:

1. In the Azure Active Directory > App registrations > New application registration, provide the following parameters:
 - **Name:** choose a name for the app that CCKM will use to access Azure.
 - **Application type:** Web app/API
 - **Sign-on URL:**
https://{CCKM Instance Host name}:8443/kmaas/login/azure (for Azure)
https://{hostname}:8443/kmaas/login/azureChina (for Azure China)

- If you are using CCKM as a service principal, then in addition to the above URLs, also add the following URLs.

Note: In all cases, {hostname} must resolve to either the IP address of your CCKM Instance, or, for CCKM clusters, the IP address of the load balancer. For more information about CCKM cluster configuration, see ["CCKM Cluster Configuration" on page 65](#)

For Azure:

https://{CCKM Instance Host name}:8443/kmaas/auth2/azure

https://{CCKM Instance Host name}:8443/kmaas/azureAdminConsent/azure

For Azure China:

https://{hostname}:8443/kmaas/auth2/azureChina

https://{hostname}:8443/kmaas/azureAdminConsent/azureChina

2. Click **Create** to create the app.
3. Access the new app under **App Registrations**.
4. From **App Registrations** > {App Name} > **Settings**: Note the **Application ID**. This is the "Client ID" used when configuring the cloud within CCKM.
5. From **App Registrations** > {App Name} > **Settings** > **Properties**: select **Yes** for **Multi-tenanted**.
6. Click **Save**.

Set Required Permissions on CCKM App

To set the required permissions on the CCKM app in the Azure portal:

1. Access **App Registrations** > {App Name} > **Settings** > **Required Permissions**.
2. Click **+Add** and add the following APIs and their associated permissions:
 - **Azure Key Vault | DELEGATED PERMISSIONS** | ✓ Have full access to Azure Key Vault service
 - **Windows Azure Service Management API | DELEGATED PERMISSIONS** | ✓ Access Azure Service Management as organization users (preview)
 - **Windows Azure Active Directory (Microsoft.Azure.ActiveDirectory) | DELEGATED PERMISSIONS** | ✓ Sign in and read user profile
 - **Microsoft Graph | DELEGATED PERMISSIONS** | ✓ Read directory data

Note: The Microsoft Graph permission is an **optional** permission. The Microsoft Graph permission is required to query the names of the Azure applications or services using an Azure key from Azure Logs Analytics when you generate the **Azure Key Service Usage Report** from CCKM. The generated report includes the names of these applications or services and other related information. ***Without this permission set, the IDs (instead of the names) of the applications or services using an Azure key are included in the report after you run it.***

3. Select **Grant Permissions** and select **Yes**. (You must have Azure administrator permissions to Grant Permissions as required by the Microsoft Graph API.)

Assign Azure Contributor Role to CCKM App as Service Principal

To assign the Azure role of "Contributor" to the CCKM App as a service principal in the Azure portal:

1. Access **Subscriptions** > {*Subscription Name*} > **Access control (IAM)**.
2. Click the **Role assignments** tab to view the role assignments.
3. Click **Add** > **Add role assignment**. The **Add role assignment** pane displays.
4. In the **Role** list, select the **Contributor** role.
5. In the **Assign access to** list, select **User, group, or service principal** to assign access to the service principal.
6. In the **Select** list, select the name for the App Name that CCKM will use to access Azure.
7. Click **Save**.

Create a Key in Azure Portal

If you will use a secret key or client secret in Azure as part of registering the apps in Azure, create this key in Azure. When setting access to an Azure cloud in the CCKM as part of adding a cloud to CCKM, you will provide this secret key.

To create a secret key (or client secret) in the Azure portal:

1. Access **App Registrations** > {*App Name*} > **Settings** > **Keys**.
2. Enter a key name in **Key description** box.
3. Select a duration period in the **Duration** box.
4. Click **Save**. The **Key Value** column displays the new client secret.



WARNING

Copy this value and store securely; it cannot be retrieved after leaving this view. It will display as Hidden within the Value column. (This value is the "client secret" used when configuring the cloud in CCKM.)

Chapter 8: Setting Up CCKM App for Access to Google Cloud

Set Up OAuth Consent Screen and Request Verification	48
Enable Identity and Access Management API	49
Enable Cloud Resource Manager API	49
Enable Cloud Key Management Service (KMS) API	49
Create OAuth2 Client ID Credentials	50
Set Session Duration for Google Cloud Console	50
Create Service Account	51
Grant IAM Roles to User and Service Accounts	52
Create Custom Roles	53

Prior to configuring and accessing the Google Cloud KMS in CCKM, ensure all of the prerequisites are met within the Google Cloud console. See "[Prerequisites on Google Cloud](#)" on page 151 for a list of all of the prerequisites. This chapter provides information on how to perform most of these prerequisites. For information on how to address the other prerequisites not covered in this document, refer to the Google Cloud online documentation.

Note

This documentation assumes that you are familiar with using Google Cloud Platform, Google Cloud KMS, and Google Identity and Access Management (IAM). For details about these Google technologies, refer to the Google Cloud documentation at <https://cloud.google.com/docs>.

Set Up OAuth Consent Screen and Request Verification

Before your CCKM users can authenticate to Google Cloud, a consent screen will be required to display. This screen will allow your users to choose whether they will allow access to their private data, as well as provide them a link to your terms of service and privacy policy. The consent screen is set up at the project level.

Note

As part of the OAuth Consent screen setup, you will be adding a scope that is sensitive. This type of scope requires verification from the Google Cloud security team, which can take approximately four to six weeks to complete.

For development purposes, the consent screen does not need to be submitted for verification. However, OAuth is restricted to 100 sensitive scope logins until the consent screen is published.

After performing the steps in this section, the next step to take is to enable IAM API (see "[Enable Identity and Access Management API](#)" on the next page).

To set up your consent screen for a project and request verification:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. From the left-side menu, select **APIs & Services**. The **APIs & Services** dashboard displays.
4. From the left-side menu, click **OAuth consent screen**. The **OAuth consent screen** displays.

5. From the **Application Type**, select **Public** or **Internal** depending on whether you will allow users to access CCKM outside of your organization. A public application allows access to users *outside* of your organization as well as inside your organization while an internal application allows access to users from your organization.
6. In the **Application name** box, enter the name of the app asking for consent. This is the name of your CCKM app.
7. Under **Scopes for Google APIs**, click **Add scope** and add ".../auth/cloudkms" and ".../auth/cloud-platform" as required scopes.
8. Under **Authorized domains**, enter the domain name of your organization and press **Enter**.
9. Click **Save**.
10. Click **Submit** for verification.

Enable Identity and Access Management API

Note

Enable IAM APIs after you set up your consent screen for a project and request verification (see ["Create OAuth2 Client ID Credentials" on the facing page](#) and after you create a service account for a project (see ["Create Service Account" on page 51](#)).

To enable IAM APIs:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. In the search box, enter "Identity and Access Management".
4. Select **Identity and Access Management (IAM) API**. The **Identity and Access Management (IAM) API** page displays.
5. Select the **ENABLE** button.

Enable Cloud Resource Manager API

To enable Cloud Resource Manager APIs:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. In the search box, enter "Cloud Resource Manager API".
4. Select **Cloud Resource Manager API**. The **Cloud Resource Manager API** page displays.
5. Select the **ENABLE** button.

Enable Cloud Key Management Service (KMS) API

To enable Cloud Key Management Service (KMS) APIs:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. In the search box, enter "Cloud Key Management Service (KMS) API".

4. Select **Cloud Key Management Service (KMS) API**. The **Cloud Key Management Service (KMS) API** page displays.
5. Select the **ENABLE** button.

Create OAuth2 Client ID Credentials

CCKM must have authorization credentials that identify itself to Google's OAuth 2.0 server. The OAuth credentials will be known to both CCKM and Google Cloud. You are required to create credentials for your project. CCKM will use these credentials that you have enabled for your project to access Google Cloud APIs. It is these credentials (Client ID and Client secret) that you will enter in the CCKM Admin portal to add Google Cloud service to CCKM.

Note

After creating the OAuth2 client ID credentials, copy the Client ID and Client secret from the **Web application** page, paste this information into a file, and then save it. When adding a Google Cloud service to CCKM, refer to this file to obtain the Client ID and Client secret.

To create the OAuth 2 Client ID credentials:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. From the left-side menu, select **APIs & Services**. The **APIs & Services** dashboard displays.
4. From the left-side menu, click **Credentials**. The **Credentials** page displays.
5. Click **+ Create Credentials**, then select **OAuth client ID**.
6. In the **Application type** drop-down menu, select **Web application**.
7. In the **Name** box, enter the name of your OAuth client (CCKM app name).
8. Under **Authorized redirect URIs**, click **+ ADD URI** and enter the CCKM redirect URI(s) in the **URIs** box. The OAuth 2.0 server will use the redirect URI(s) (CCKM endpoint(s) to which to send its responses. For example, ***https://cckm.example.com:8443/kmaas/login/googleCloud*** where "cckm.example.com" is the fully qualified domain name of your CCKM.
9. Click **Create**. The **Client ID for Web application** page displays including the name of OAuth client, Client ID, Client secret, and creation date.

Set Session Duration for Google Cloud Console

As organization administrator, you can configure a fixed session duration for specific apps and services that use Google Cloud. With a fixed session duration set, users are required to re-enter their login credentials to continue to access the Google Cloud console after the session expires.

If your Google user account belongs to an organization, your refresh token will be affected by your organization's security policy for Google Cloud session control. For more information, see [Control session length for Google Cloud Console and gcloud CLI](#).

If you are employing key rotation and key synchronization schedules in CCKM, you are required to use the session duration option of **Session never expires** when setting your security policy for Google Cloud session control. Otherwise, your scheduled job will fail when the given schedule begins. This is due to CCKM saving the refresh token (to MongoDB) that it obtains from Google Cloud when a user logs into CCKM for the first time. When CCKM runs the scheduled job, it uses the refresh token to obtain an access token from Google Cloud. Due to the security policy for session control being set, the saved refresh token expires after the session duration (1 to 24 hours

depending on configuration), disallowing CCKM from obtaining the access token to begin the scheduled job. As a result, the scheduled job will fail. To ensure the scheduled job is successful, set the session duration to **Session never expires**.

Note

If you are employing key rotation and key synchronization schedules in CCKM, it is recommended that you first create an organization unit, and then add all the users who will use CCKM to that unit. Thereafter, change the security policy for that unit to **Session never expires**.

As an *alternative* to setting the session duration to **Session never expires** for your security policy for session control to avoid the failure of scheduled jobs, use service accounts. You can create a service account, use the service account to authenticate to the CCKM REST API, and then add schedules using CCKM REST API. These schedules added by the service account do not use refresh tokens and are not affected by the security policy for session control. For more information about creating a service account, see ["Create Service Account" below](#).

To set up a session duration for an organization:

1. Login as admin of your organization at the Google Admin console (<https://admin.google.com>).
2. From the home page, go to **Security** and then Google Cloud session control (Beta).
3. Under **Organization Unit** (on the left), select your organization unit.
4. Under **Session duration**, configure the session duration using *one* of the following ways:
 - a. If you will set up key rotation and key synchronization schedules in CCKM, select **Session never expires** so that the session never expires.
 - b. Select **Set session duration** and from the drop-down list for **session duration**, select from 1 to 24 hours.
5. Click **Save**.

Create Service Account

To employ CCKM REST APIs to manage Google Cloud keys, you will need to use a service account from Google Cloud to authenticate to the CCKM REST APIs.

Note

After successfully creating your service account, take note of the email address that Google Cloud assigned to it. You are required to enter this email address when adding service account to CCKM.

After performing the steps in this section, proceed to enabling IAM API (see ["Enable Identity and Access Management API" on page 49](#)). Thereafter, grant IAM Roles to the service account (see ["Grant IAM Roles to User and Service Accounts" on the facing page](#)).

To create a service account for a project:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. From the left-side menu, select **IAM & Admin > Service Accounts**. The **Service Accounts** dashboard displays.
4. Click **+ CREATE SERVICE ACCOUNT**. The **Create service account** page displays.
5. In the **Service account name** box, enter the name of the service account.
6. (Optional) In the **Service account description** box, enter a description of the service account.
7. Click **Create**.

Grant IAM Roles to User and Service Accounts

For your user account or service account to use CCKM, grant the Project Browser and Cloud KMS Admin roles to the account. If your project belongs to an organization, then it is recommended that you grant the Organization Viewer role to your accounts.

The role of Project Browser allows read access to browse the hierarchy of a project, including the folder, organization, and IAM policy. The role of Cloud KMS Admin gives full access to Cloud KMS resources with the exception of encrypt and decrypt operations. Access to view an organization name is given through the use of the Organization Viewer role. This role allows CCKM to find the organization to which the account belongs and display the organization name within the CCKM User Portal. For more information about Google Cloud roles, see [Predefined roles within Google Cloud online documentation](#).

If you are using a service account, the next step to take after performing the steps in this section is to add a service account in CCKM. See ["Adding Google Cloud Service Account to CCKM" on page 180](#) for more information. Thereafter, you can use the service account to authenticate CCKM REST API. After successfully authenticating to the CCKM REST APIs, you can proceed to making calls to CCKM REST APIs to manage Google Cloud keys. For more information, see [Appendix D: "Sample Python Scripts for Authenticate and Call CCKM REST API" on page 251](#) and [Appendix E: "Authenticate your REST API Client to CCKM REST APIs for Google Cloud Keys" on page 258](#).

If this permission is not granted to the user who logs into CCKM, this user will not be able to see the organization name. Only the organization ID will be displayed for the user on CCKM.

Grant Project Browser and Cloud KMS Admin Roles

To grant the roles of Project Browser and Cloud KMS Admin to your user account or service account:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. From the left-side menu, select **IAM & Admin > IAM**. The **IAM** dashboard displays.
4. Click **+ ADD**. The **Create service account** page displays.
5. In the **New members** box, enter the username of your user account you wish to grant roles (if this is for a user account). Enter the email address of the service account you wish to grant roles (if this is for a service account).
6. In the **Select a role** box, select the role of **Project** (on the left), and **Browser** (on the right).
7. Click **ADD ANOTHER ROLE**.
8. In the **Select a role** box, select the role of **Cloud KMS** (on the left), and **Admin** (on the right).
9. Click **Save**.

Grant Organization Viewer Role

If your user or service account belongs to a project that is a member of an organization, and you wish to view the organization name in CCKM, then grant the role of Organization Viewer to these accounts at the organization level in addition to granting the roles of Project Browser and Cloud KMS Admin at the project level. Granting the role of Organization Viewer is *optional*. If you do not grant the Organization Viewer role to your accounts, the organization name does not display in CCKM. Instead, only the organization ID displays.

Note

To grant the role of Organization Viewer to your user or service account, the role of Organization Administrator is *required*. If you do not have this role, make the request of your Organization Administrator to grant this role to your user or service account.

To grant the Organization Viewer role to your user account or service account:

1. (As an Organization Administrator) Go to the Google Cloud Platform Console.
2. From the Organization list, select the organization in which the CCKM app belongs.
3. From the left-side menu, select **IAM & Admin > IAM**. The **IAM** dashboard displays.
4. Click **+ ADD**. The **Create service account** page displays.
5. In the **New members** box, enter the username of the user account you wish to grant the role (if this is for a user account). Enter the email address of the service account you wish to grant the role (if this is for a service account).
6. In the **Select a role** box, select the role of **Resource Manager** (on the left), and **Organization Viewer** (on the right).
7. Click **Save**.

Create Custom Roles

If you wish to grant specific permissions to users or groups to meet your specific requirements, then create custom roles. Group one or more of the supported permissions into a custom role and grant this role to users or groups. See ["Create Custom Roles" above](#) table for the permissions that are required for each of the supported CCKM actions.

To create custom roles to use for CCKM:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. From the left-side menu, select **IAM & Admin > Roles**. The **Create Role** dashboard displays.
4. In the **Title** box, enter the name of the custom role.
5. Click **+ ADD PERMISSIONS**. The **Add permissions** page displays.
6. From the list of permissions, select the permission(s) you wish to use for the custom role.
7. Click **ADD**.

Chapter 9: Create a Connected CCKM App in Salesforce

Creating a Connected CCKM App in Salesforce	54
Editing CCKM App to Update OAuth Policies	55

Before adding a Salesforce, Salesforce Government, or Salesforce Sandbox cloud in CCKM, you must create a connected CCKM app and assign the required permissions to it in the given Salesforce portal (Salesforce, Salesforce Government, or Salesforce Sandbox).

If you are employing key rotation and key synchronization schedules in CCKM, you are required to set the **refresh token for the app is valid until revoked** within **Refresh Token Policy** of OAuth policies **after** successfully creating a connected CCKM app in Salesforce.

Note
In this document, Salesforce Government Cloud Plus is also referred to as Salesforce Government and Salesforce Government Cloud.

Note
The information regarding Salesforce in this document is applicable to Salesforce, Salesforce Government, and Salesforce Sandbox unless otherwise noted. The CCKM UI for Salesforce, Salesforce Government, and Salesforce Sandbox is identical with the exception of the display of the names of the clouds. Your login credentials for Salesforce, Salesforce Government, and Salesforce Sandbox are unique to each cloud.

Note
Salesforce provides the following profile interfaces: original profile user interface and enhanced user interface. The procedures provided in this chapter reflect the enhanced user interface. For more information about the Salesforce profile interfaces, refer to the appropriate Salesforce online documentation.

Creating a Connected CCKM App in Salesforce

Note
Prior to creating a connected CCKM app in Salesforce, ensure you have the required user permissions to create connected apps. For information about these user permissions, refer to the appropriate Salesforce online documentation.

1. In the Salesforce app creation interface, provide basic information of the app. For example, app name, and description.
2. Define how the app communicates with Salesforce using the following **API** parameters:

- Select the **Enable OAuth Settings** check box.

- Within the **Callback URL** box, add the callback URLs for Salesforce, Salesforce Government, or Salesforce Sandbox depending on which cloud you are configuring.

Callback URL for Salesforce: `https://{hostname}:8443/kmaas/login/salesforce`

Callback URL for Salesforce Government: `https://{hostname}:8443/kmaas/login/salesforcegov`

Callback URL for Salesforce Sandbox: `https://{hostname}:8443/kmaas/login/salesforcesandbox`

{hostname} must resolve to the IP address of your CCKM Instance.

Note: If you plan to deploy a CCKM cluster, then the {hostname} must resolve to the IP address of the load balancer. For more information about CCKM cluster configuration, see ["CCKM Cluster Configuration" on page 65](#).

- For **Selected OAuth Scopes**: add the following options:
 - **Access and manage your data (api)**
 - **Access your basic information (id, profile, email, address, phone)**
 - **Perform requests on your behalf at any time (refresh_token, offline_access)**

3. Click **Save**. The **Consumer Key** is created and displayed, and the **Consumer Secret** is created.

Note the consumer key and secret and store them securely; you will use them when adding the Salesforce, Salesforce Government, or Salesforce Sandbox cloud in CCKM.

Editing CCKM App to Update OAuth Policies

If you are employing key rotation and key synchronization schedules in CCKM, you are required to set the **Refresh token is valid until revoked** within **Refresh Token Policy** of Salesforce OAuth policies **after** successfully creating a connected CCKM app in Salesforce. Otherwise, your scheduled job will fail when the schedule begins. This is due to CCKM saving the refresh token (to MongoDB) that it obtains from Salesforce cloud when you log into CCKM for the first time. When CCKM runs the scheduled job, it uses the refresh token to obtain an access token from Salesforce cloud. However, the saved refresh token does expire after the session duration (1 to 24 hours depending on configuration) disallowing CCKM from obtaining the access token to begin the scheduled job. As a result, the scheduled job will fail. However, you can set the Salesforce **Refresh Token Policy**, which determines how long a refresh token is valid, so that the refresh token is valid until revoked. Setting the **Refresh token is valid until revoked** allows the connected CCKM app to use the refresh token indefinitely (unless you or your administrator revoke it). To ensure the scheduled jobs are successful, set the **refresh token for the app is valid until revoked** setting. For more information about Salesforce refresh access tokens, refer to the appropriate Salesforce online documentation.

Note

If you are upgrading from older versions of CCKM to the current version of CCKM and using CCKM for Salesforce or Salesforce Sandbox key management, this section is also applicable to you. After upgrading to the latest version of CCKM, perform the procedure in this section. Otherwise, you will **not** be able to log into the CCKM User Portal for any of the Salesforce clouds including Salesforce Sandbox.

To set the **Refresh token is valid until revoked** setting as part of the Salesforce OAuth policy settings of the connected CCKM app:

1. From **Manage Apps > Connected Apps > {created_CCKM_connected_app}**, click **Action**.
2. Select **Edit** from the drop-down menu.
3. From the **Connect App Edit** page, select **Refresh token is valid until revoked** within **OAuth Policies > Refresh Token Policy**.
4. Click **Save**.

Chapter 10: Install CCKM

Install on VMware ESXi Server	58
Install on AWS	59
Install on Microsoft Hyper-V	61
Launch on Azure Marketplace	61
Use SSH to Change the Default Password on the CCKM VM	61
Synchronize CCKM Local time with Network Time using NTP Server	62
Configure Multiple NICs on CCKM Instance	63
Access the CCKM GUI	64
CCKM Cluster Configuration	65

Install CCKM on VMware, AWS, or Microsoft Hyper-V environments. For the Azure Marketplace, launch and configure CCKM from the marketplace.

Note

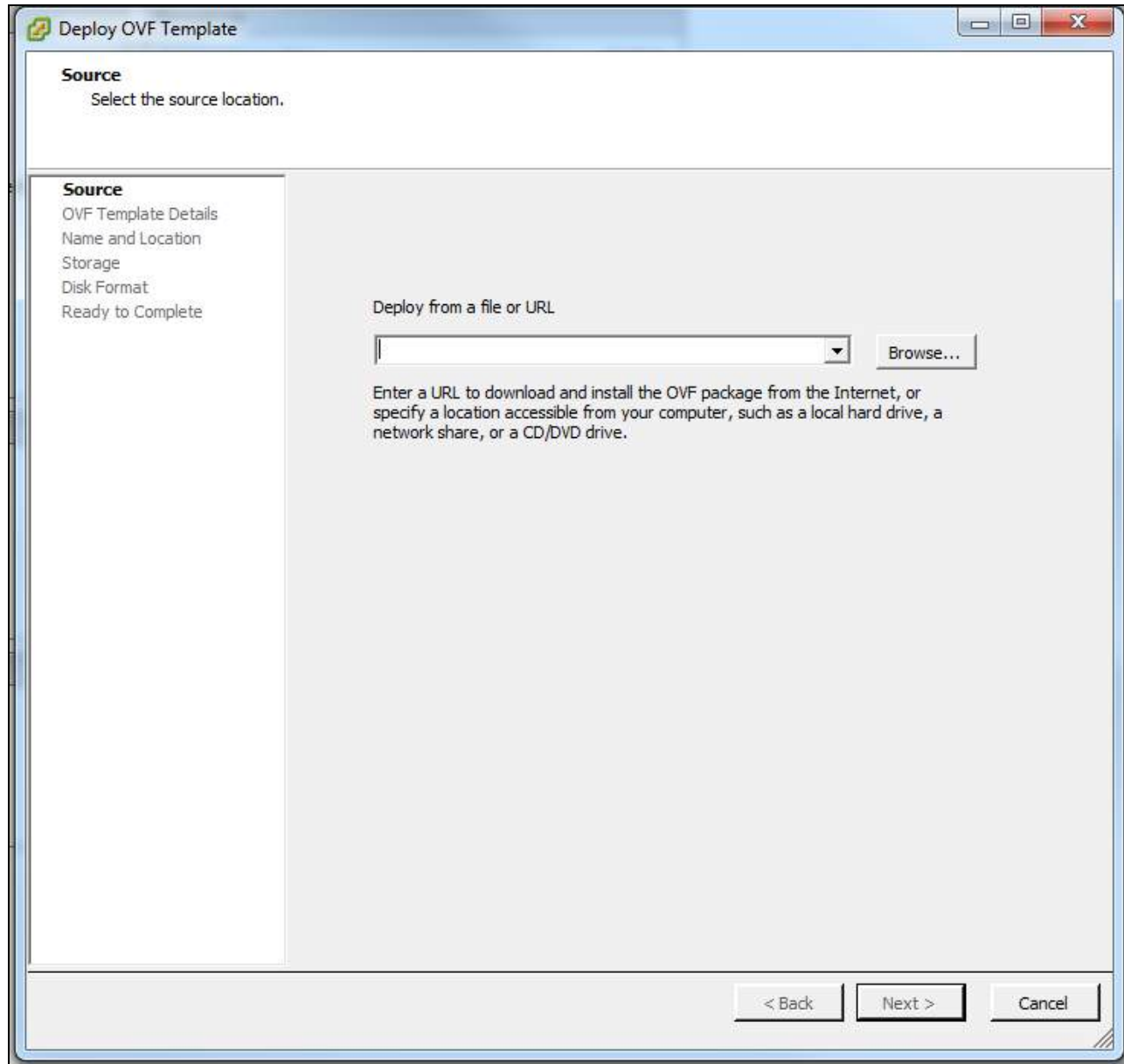
CCKM is supported on VMware ESXi 6.0 and later versions.

After installing CCKM and changing the default password, the **first** step to take is to synchronize the local time with network time using your preferred Network Time Protocol (NTP) server(s). Ensure at least one NTP server is configured. For more information, see ["Synchronize CCKM Local time with Network Time using NTP Server" on page 62](#).

(Optional) If you wish to use multiple NICs, configure the NIC(s) **after** adding an NTP server to CCKM and synchronizing with it. Then proceed to with accessing the CCKM GUI **after** completing the NIC configuration.

Install on VMware ESXi Server

To create the CCKM Server on a VMware ESXi server, load the CCKM server's .ova template file in the VMware ESXi server and create the VM. Give the CCKM VM a name and keep the default resource settings.



Take note of the IP address of the VM. This will be the IP Address of CCKM.

Install on AWS

The administrator should be familiar with AWS in general and AMI installations in particular.

Start the Installation

Contact Thales Support to obtain the CCKM Amazon Machine Image (**AMI**).

1. Log in to your AWS account.
2. Find the AMI in the AWS Dashboard at **EC2 > IMAGES > AMIs > Private images**.
3. Select the correct version of the CCKM AMI and click **Action > Launch**.
4. At **Choose an Instance Type**, select:
 - **Family**: General Purpose
 - **Type**: m4.xlarge
 - **vCPUs**: 4
 - **Memory**: 16
 - **Network Performance**: High
5. Click **Next**: Configure Instance Details.

Step 3: Configure Instance Details

Configure the instance details. You can't launch or update instances (or use the console) until you've specified the instance type, which determines the hardware configuration of the instance, such as the number of vCPUs, memory, and network bandwidth.

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	On-Demand Instance	
Network	us-east-1c	Create new VPC
Subnet	us-east-1c	Create new subnet
Auto-assign Public IP	Yes (will be setting later)	
Placement group	Default	Create new Placement group
Capacity reservation	None	Create new Capacity reservation
Launch type	On-demand	Create new On-demand
Shutdown behavior	Stop	
Instance-level metrics	Basic	
Monitoring	Amazon CloudWatch	
Termination protection	Off	
T2 Unlimited	On	

Configure Instance Details

Fill out the *Configure Instance Details* page as follows:

1. In the **Network** box, select a VPC.
2. Select your subnet; it must be in the same subnet as the CipherTrust Manager or DSM depending on which key source you are using.
3. If you want the CCKM available on a public IP address, then set *Auto-assign Public IP* to **Enable**.

This will become the CCKM GUI address.

4. Click **Next: Add Storage**.
5. Continue to "[Complete the Installation](#)" below.

Complete the Installation

On the *Add Storage* page:

1. Use the default settings for adding storage. Click **Next: Tag Instance**.
2. In the *Tag Instance* page, create a name (key) and value to help organize your Amazon resources. In the *Name* column, type "**Name**". In the *Value* column, type the name you want to use for this instance. Note the following about tags (from Amazon's AWS documentation):
Tags don't have any semantic meaning to Amazon EC2 and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. If you delete a resource, any tags for the resource are also deleted.
3. Click **Next: Security Group**.
4. Select **Create a new security group**, enter a group name, and click **Add Rule** to configure the following sets of parameters:

- **Type:** SSH, **Port:** 22, **Source:** specify the SSH source that can reach your instance
- **Type:** HTTPS, **Port:** 443, **Source:** This port is used to access the CCKM Admin UI (CCKM Admin Portal)
- **Type:** HTTPS, **Port:** 8443, **Source:** This port is used to access the CCKM GUI (CCKM user portal)

Note: Rules with source 0.0.0.0/0 allow all IP addresses to access your instance (*not recommended*). Do not leave source IP addresses with the default 0.0.0.0/0.

5. Click **Review** and Launch.
6. Review parameters and click **Launch**.
A dialog box allows you to define a new public key and private key for EC2.
7. Create a new key/value pair and click **Launch Instances**.
8. You can now use the public or private DNS in your browser to access the CCKM GUI.

Install on Microsoft Hyper-V

To create a CCKM server on a Microsoft Hyper-V environment, load the CCKM VHD file and then create a virtual machine (VM) with the Hyper-V Manager. After you create the VM, you can begin to configure CCKM using the CCKM CLI.

Create CCKM VM

To create a CCKM VM with Hyper-V Manager:

1. In Hyper-V Manager, right-click and select **New Virtual Machine** to start the Hyper-V wizard.
2. Review the **Before You Begin** content (if it displays) and click **Next**.
3. At the **Specify Name and Location** page, specify the name for your VM and click **Next**.
4. At the **Specify Generation** page, leave the default of **Generation 1** and click **Next**.
5. At the **Assign Memory** page, enter the minimum of 4096 MB and click **Next**.
6. At the **Configure Networking** page, select the desired network connection and click **Next**.
7. At the **Connect Virtual Hard Disk** page, select **Use an existing virtual hard disk**, browse to select your CCKM VHD file, and click **Next**.
8. At the **Summary** page, click **Finish** to complete the wizard.

Power on CCKM VM and Connect to It

To power on the CCKM VM, connect to it, and then log into the CCKM CLI:

1. Right click on your new VM and select **Start** to power it on.
2. Right click on the VM again and select **Connect** to open a console for your VM.
3. Wait until the "cckm login" prompt displays (this may take a few minutes) and then log into CCKM CLI to configure your CCKM VM.

Launch on Azure Marketplace

Launch and configure CCKM from Azure Marketplace. Refer to the Azure online documentation for information.

Use SSH to Change the Default Password on the CCKM VM

After launching CCKM on a VM, it is important to change the default password as a security measure. To do so:

Note

After changing the default password, the **next** step is to add an NTP server to CCKM and synchronize the local time with the network time. For more information, see ["Synchronize CCKM Local time with Network Time using NTP Server" on the next page](#).

1. SSH to the IP address of the new VM (the IP address of CCKM).
2. Enter the username and password as prompted.
Username: cliadmin
Password: cliadmin123

3. Follow the prompt: Enter the old password, and then a new password twice. **Store securely.**
ATTENTION: This username/password combination is used to access the VM via SSH. There is no way to recover a forgotten password—in this case, the VM would have to be discarded and a new CCKM launched.
4. The SSH session automatically ends. To validate, launch a new session and log in with the new password.

Synchronize CCKM Local time with Network Time using NTP Server

In this release of CCKM, the local time on CCKM server is no longer synchronized with the network time. **After installing CCKM** and changing the default password, the **first** step to take is to synchronize the local time with the network time using your preferred Network Time Protocol (NTP) server(s). For example purposes, the “pool.ntp.org” is used as the NTP server. However, you can use any NTP server you wish to use.

Note

At least one NTP server must be configured **before** you configure to add the NTP server to CCKM and synchronize with it.

After adding an NTP server to CCKM and synchronizing the local time with the network time, the **next** step is to access the CCKM GUI. For more information, see ["Access the CCKM GUI" on page 64](#). However, if you are using multiple NICs, then configure the NIC(s) **after** configuring to add the NTP server to CCKM and synchronize with it. Then proceed to access the CCKM GUI **after** completing the NIC configuration. For more information, see ["Configure Multiple NICs on CCKM Instance" on the facing page](#).

To add an NTP server to CCKM and synchronize the CCKM local time to the network time:

1. Access your CCKM server through SSH (such as Putty):

```
ssh cliadmin@<cckmhost>
```
2. Enter your user password when prompted:

```
cliadmin@>cckmhost>'s password:  
CCKM CLI Main Menu  
cckm>
```
3. Access the CCKM Network command category:

```
cckm> network
```
4. Access the NTP service settings and add the named NTP server to contact for time synchronization:

```
network> ntpservice --add pool.ntp.org  
NTP Servers:  
pool.ntp.org
```
5. Perform immediate clock synchronization with the configured NTP server:

```
network> ntpservice --sync  
Synchronizing time with NTP server(s):  
pool.ntp.org  
19 Mar 23:23:25 ntpdate[18641]: step time server 217.91.44.17 offset  
29286.555701 sec
```

6. Verify that the status of the NTP service is enabled:

```
network>
network> ntpservice --status
enabled
Active: active (running) since Thu 2020-03-19 14:58:26 PDT; 8h ago
No association ID's returned
network>
```

Configure Multiple NICs on CCKM Instance

CCKM supports multiple network interface controllers (NICs). This feature is available for ESXi and Azure cloud deployments. It is not available on AWS instances. Using the CLI network commands, you can assign a static IP to each NIC and add new routes to the routing table, **Manually** attach a NIC to a CCKM instance **prior** to performing the configurations in this section.

This section is only applicable to you if you are using multiple NICs.

Configure Network Interface

To configure the interface:

1. Log into the CCKM instance using cliadmin user.
2. View the network device configuration to verify the new NIC is attached to instance:
3. Enter the network command category and configure a static IP address for the specified network interface:

```
network> show --list

network> setup eth1
<dhcp or static>? static
ipv4 address? 10.3.200.201
Subnet mask (e.g., 255.255.255.0)? 255.255.0.0
```

OR

Enable DHCP IP address leasing for the specified network interface:

```
network> setup eth1
<dhcp or static>? dhcp
```

4. Display the network interface setting to view the IP address configuration :

```
network> ip --show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:42:91:42 brd ff:ff:ff:ff:ff:ff
inet 10.0.0.6/24 brd 10.0.0.255 scope global eth0
valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:91:da:8c brd ff:ff:ff:ff:ff:ff
inet 10.0.0.4/24 brd 10.0.0.255 scope global eth1 valid_lft forever preferred_
lft forever
```

Add New Routes to Routing Table

To add a new route to the routing table:

Note

The default route is associated with eth0.

1. From the network commands, add a route to the routing table:

```
route add -net 10.10.65.1/24 gw 10.10.65.254 dev eth1
INFO: Execute 'route --make-persistent' command to make these changes reboot
persistent
```

2. Set the routing table changes to be persistent after a reboot:

```
route -make-persistent
WARNING: This command will make the current routing table persistent after
reboot.
Any invalid configuration in routing table can make the system inaccessible over
the network. Do you wish to continue? <y/n> [n] y
SUCCESS: Routes are now reboot persistent
```

Access the CCKM GUI

Open a browser and point to the CCKM IP address with the https prefix (for example: <https://19.4.5.89>).

Once the CCKM is configured, the CCKM IP address will open to the CCKM GUI Home page.

CCKM Cluster Configuration

For high availability (HA) and improved performance of the CCKM solution, you can set up a CCKM cluster using at least two CCKM servers and one MongoDB database. To set up the CCKM cluster, launch two or more CCKM VMs using the same OVA, AMI, or VHD file for each VM you add to the cluster, configure the VMs to use the same MongoDB database.

When you configure the database information on the second or additional CCKMs (following the steps in section ["Set Up Database" on page 67](#)), enter the same configuration passphrase and database information that you entered (in the **Configuration Passphrase** box) for the first CCKM. This configuration allows the additional CCKM servers to join the first CCKM in the cluster and share the same CCKM configuration in the MongoDB database.

You can access the CCKM Web UI and REST API on any of the CCKM servers.

Optionally you can add a load balancer in front of all CCKM servers. In this case, you can access CCKM Web UI and REST API using your load balancer. When configuring the load balancer, configure the listeners to forward traffic for ports 443 and 8443. For port 8443, enable session stickiness. For example, if you are using an AWS load balancer, enable session stickiness on cookie "JSESSIONID". If you are using an Azure application gateway, enable cookie based affinity. For health check, set the ping target to "https:8443/kmaas/health". Refer to the documentation for the load balancer you are employing in the CCKM cluster for information on how to configure this configuration.

Chapter 11: Initial Configurations within CCKM Admin Portal

Summary of Initial Configuration Steps	66
Change Default Admin Portal Password (During First-time Login)	67
Login into Admin Portal	67
Set Up Database	67
Upload License	67
Configure Key Source	68
Add Cloud	68

This chapter provides the required steps for the initial configuration of the CCKM Admin Portal **after** you install CCKM and login into the CCKM Admin Portal for the **first** time.

After installing CCKM, access the CCKM Admin Portal at https://<CCKM_hostname_or_IP_address> to begin configuring CCKM.

Note

In this document, Salesforce Government Cloud Plus is also referred to as Salesforce Government and Salesforce Government Cloud.

Summary of Initial Configuration Steps

The following is a summary of the required steps to perform during the initial configuration of the CCKM within the CCKM Admin Portal to get you up and running to begin using CCKM. For steps 2 through 6, you are prompted (at the top of the page) with each step to take next. After you *initially* log into CCKM Admin Portal *after* the CCKM installation, you are prompted to change your CCKM Admin Portal password.

1. Change your default Admin Portal password. (See ["Change Default Admin Portal Password \(During First-time Login\)" on the next page.](#))
2. Login into Admin Portal. (See ["Login into Admin Portal" on the next page.](#))
3. Configure your database settings. (See ["Set Up Database" on the next page.](#))
4. Upload your license file. (See ["Upload License" on the next page.](#))
5. Configure your key source (CipherTrust Manager, DSM, or nShield). (See ["Configure Key Source" on page 68.](#))
6. Add cloud(s). (See ["Add Cloud" on page 68.](#))

Change Default Admin Portal Password (During First-time Login)

You are required to change the default admin password for the CCKM Admin Portal **after** your **initial** or first-time login into this portal. As part of the initial login, enter the default admin username of "**admin**" and the default admin password of "**admin123**". When prompted to change the current password, do the following:

1. Enter the current admin password (default admin password).
2. Enter your new password.
3. Enter your new password again to confirm it.
4. Click **Update**.

Login into Admin Portal

After changing the admin password, you are prompted to log into CCKM Admin Portal again using your new password. Enter your admin username and password and click **Login**.

Set Up Database

You are prompted to set up your database within the **Settings** page. In the **Database** settings page, you have the following options to configure MongoDB:

- Local Database
- Remote Database
- ReplicaSet Database
- MongoDB Atlas

The CCKM server requires a MongoDB database (MongoDB, MongoDB Atlas, or Percona Server for MongoDB) to store user setup and cloud usage data.

For information on how to configure MongoDB, see "[Configure Database](#)" on page 80. **After setting up MongoDB settings successfully, CCKM restarts.**

Upload License

After the CCKM service is restarted, you are prompted to go to the **Licenses** page to upload your CCKM license(s) for the cloud(s). You must have a valid license file to upload to the **Licenses** page. Once the license file is uploaded to this page, information about this license displays on the page. To obtain your CCKM license(s) for one or more clouds, contact Thales Sales.

To upload a license file:

1. Click **Upload License** at the top of the page.
2. From the **Upload License** dialog box, select **Choose File** and choose the license.
3. Click **Upload**. The license is displayed on the **Licenses** page.

Configure Key Source

You are required to configure the key source as part of your initial configuration of CCKM. You have three options to configure your key source from the **Key Sources** page of the CCKM Admin Portal using the following tabs from this page:

- CipherTrust Manager
- DSM
- nShield

Note

Concurrent use of the CipherTrust Manager, DSM, and nShield as key sources is not supported. Configure **only** one of these supported key sources as a key source.

For information on how to configure a key source, see ["Configure Key Source" on page 73](#)

Add Cloud

As part of your **initial** configuration of CCKM, you are required to add at least one cloud to manage through CCKM from the **Clouds** page of the CCKM Admin Portal. You can also add more clouds after the initial configuration of CCKM. The following are your options:

- AWS
- Azure
- Azure China
- Google Cloud
- IBM Cloud
- Salesforce
- Salesforce Government
- Salesforce Sandbox

For information on how to add a cloud, see ["Manage Clouds " on page 70](#)

Chapter 12: About the Admin Portal

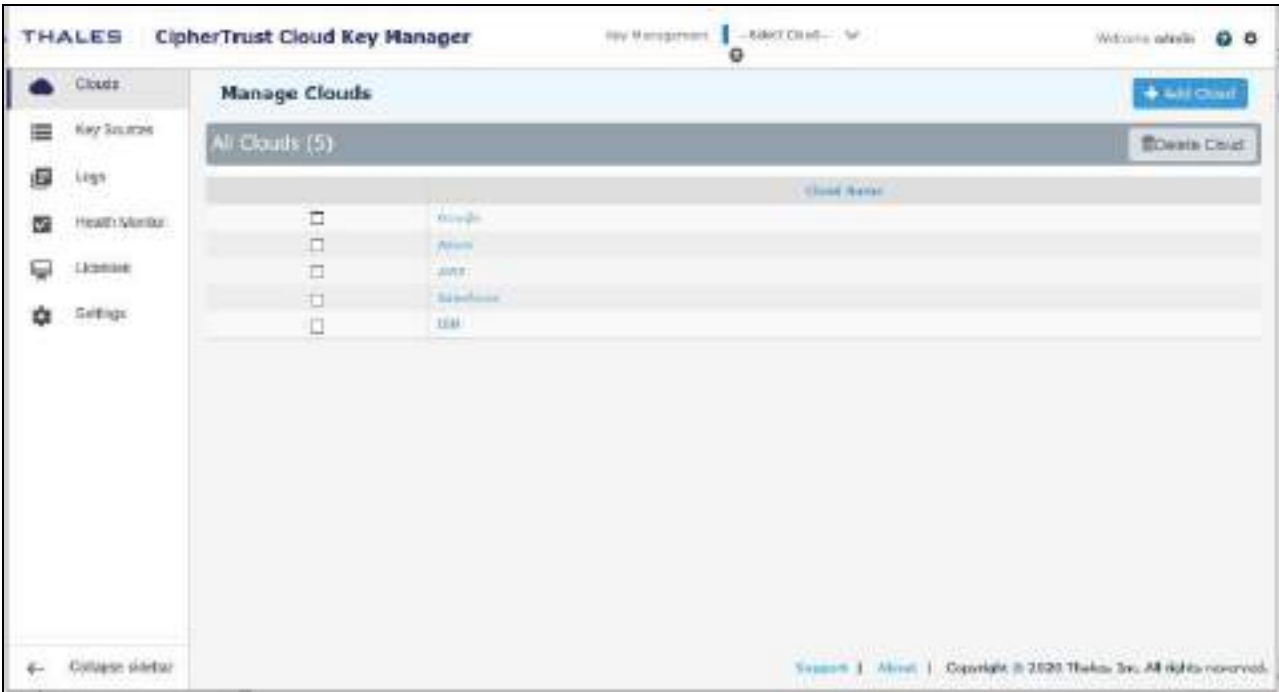
CCKM Admin Portal	69
Manage Clouds	70
Configure Key Source	73
Review Logs	77
Review Health Monitor	77
Review Licenses	79
Settings	80

This chapter describes the Web UI pages of the CCKM Admin Portal and the settings and configurations you can perform using these pages.

Note
In this document, Salesforce Government Cloud Plus is also referred to as Salesforce Government and Salesforce Government Cloud.

CCKM Admin Portal

The CCKM Admin Portal contains the following options in the left-hand navigation bar:



- **Cloud:** From the **Cloud** page, you can create a connection between CCKM and a given cloud to permit CCKM access to any of the cloud-service users with valid user credentials. From this page, you can also delete a cloud.
- **Key Sources:** From the **Key Sources** page, you can configure your key source.
- **Logs:** The **Logs** page lists audit logs related to the actions performed through the CCKM Admin Portal. You can search, sort, and view these audit logs.

- **Health Monitor:** The **Health Check Monitor** page provides a snapshot of the status of all major components in a CCKM solution. In addition, this page provides the option to remove a listing of a CCKM server that has a status of "Bad" from this page.
- **Licenses:** In **Licenses**, you can view all of your available licenses for your configured clouds.
- **Settings:** The **Settings** page allows you to configure database, proxy, SMTP, debug level, key deletion policy, and CCKM admin users.

Manage Clouds

From the **Clouds** page, you create a connection between CCKM and a given cloud to permit CCKM access to any of the cloud-users with valid user credentials. The following are the supported clouds:

- AWS
- Azure
- Azure China
- Google Cloud
- IBM Cloud
- Salesforce
- Salesforce Government Cloud
- Salesforce Sandbox

From the **Clouds** page, you can also delete a cloud.

Add Cloud

Step 1: Add Cloud

1. Select **Clouds** from the left-hand navigation bar, select **Add Cloud**, and then select the cloud you wish to add from the **Cloud** drop-down list. The following are the options:
 - AWS
 - Azure
 - Azure China
 - Google Cloud
 - IBM Cloud
 - Salesforce
 - Salesforce Government
 - Salesforce Sandbox
2. Click **Next**.
3. Proceed to the next section to set access to the cloud.

Set Access to Cloud

Refer to the appropriate section below for information on how to set access to the cloud you are adding.

- If setting access to AWS cloud, see ["Set Access to AWS Cloud" below](#)
- If setting access to Azure cloud, see ["Set Access to Azure \(Azure, Azure China\) Cloud" below](#)
- If setting access to Azure China cloud, see ["Set Access to Azure \(Azure, Azure China\) Cloud" below](#)
- If setting access to Google cloud, see ["Set Access to Google Cloud" on the next page](#)
- If setting access to IBM cloud, see ["Set Access to IBM Cloud" on the next page](#)
- If setting access to Salesforce cloud, see ["Set Access to Salesforce Cloud" on the next page](#)
- If setting access to Salesforce Government cloud, see ["Set Access to Salesforce Cloud" on the next page](#)
- If setting access to Salesforce Sandbox cloud, see ["Set Access to Salesforce Sandbox Cloud" on page 73](#)

Set Access to AWS Cloud

Step 2: Set access to AWS cloud

In this step, set access to AWS cloud:

1. (Optional) Add a description for the cloud.
2. Click **Save**.

Set Access to Azure (Azure, Azure China) Cloud

Step 2: Set Access to Azure (Azure, Azure China) cloud

In this step, set access to Azure or Azure China cloud:

Note

If you will use CCKM as a service principal, then ensure to select the **Use CCKM as Service Principal** box.

1. Enter:

Description: (Optional) the description of the cloud.

Client ID: the 'application id' of the Azure app .

If you are using a client secret, select **Use client secret** and enter the client secret in the **Client Secret for the Azure Application** box.

If you are using a certificate, select **Use certificate** and then click the **Generate and download** button. A message displays indicating a new certificate is generated and ready to download. Download the certificate and save it to a directory within your laptop or desktop. The certificate thumbprint automatically displays in the **Thumbprint** box after you download the certificate.

Note: Upload this certificate to the Azure Portal after completing the configuration of the cloud.

2. If you will use CCKM as a service principal, select the **Use CCKM as Service Principal** check box.

Note: The **Use CCKM as Service Principal** check box acts as a toggle to use CCKM on behalf of users or as a service principal. You can change this setting after completing the configuration of the cloud. To use CCKM on behalf of users, ensure this check box is deselected. In a production environment, Thales recommends that you **DO NOT** change the cloud configuration for Azure or Azure China by changing the setting for the **Use CCKM as Service Principal** check box.

3. Click **Save**.

Set Access to Google Cloud

Step 2: Set Access to Google Cloud

In this step, set access to Google Cloud:

1. (Optional) Add a description for the cloud.
2. Enter the Client credentials:
 - **Client ID:** The "application id" of the Google Cloud app.
 - **Client Secret:** This is the secret known only to the Google Cloud app and CCKM.
3. Click **Save**.

Set Access to IBM Cloud

Step 2: Set Access to IBM Cloud

In this step, set access to IBM Cloud:

1. (Optional) Add a description for the cloud.
2. Click **Save**.

Set Access to Salesforce Cloud

Step 2: Set Access to Salesforce cloud

In this step, set access to Salesforce cloud:

1. (Optional) Add a description for the cloud.
2. Enter the Client credentials:
 - **Client ID:** The value of the 'Consumer Key' of the Salesforce app.
 - **Client Secret:** The value of the 'Consumer Secret' of the Salesforce app.
3. Click **Save**.

Set Access to Salesforce Government Cloud

Step 2: Set Access to Salesforce Government cloud

In this step, set access to Salesforce Government cloud:

1. (Optional) Add a description for the cloud.
2. Enter the Client credentials:
 - **Client ID:** The value of the 'Consumer Key' of the Salesforce Government app.
 - **Client Secret:** The value of the 'Consumer Secret' of the Salesforce Government app.
 - **My Domain URL:** The value of the 'My Domain URL' of the Salesforce Government app.
3. Click **Save**.

Set Access to Salesforce Sandbox Cloud

Step 2: Set Access to Salesforce Sandbox cloud

In this step, set access to Salesforce Sandbox cloud:

1. (Optional) Add a description for the cloud.
2. Enter the Client credentials:
 - **Client ID:** The value of the 'Consumer Key' of the Salesforce Sandbox app.
 - **Client Secret:** The value of the 'Consumer Secret' of the Salesforce Sandbox app.
3. Click **Save**.

Configure Key Source

You are required to configure the key source as part of your initial configuration of CCKM. You have three options to configure your key source from the **Key Sources** page using the following tabs from this page:

- CipherTrust Manager
- DSM
- nShield

Note

Concurrent use of the CipherTrust Manager, DSM, and nShield as key sources is not supported. Configure only one of these supported key sources as a key source.

To configure CipherTrust Manager, select the **CipherTrust Manager** tab. For information on how to configure CipherTrust Manager, see ["Configure CipherTrust Manager as Key Source" on the next page](#).

To configure DSM, select the **DSM** tab. For information on how to configure DSM, see ["Configure DSM as Key Source" on page 75](#).

To configure nShield, select the **nShield** tab. For information on how to configure nShield, see ["Configure nShield Connect \(for Azure and AWS Clouds\)" on page 75](#).

Configure CipherTrust Manager as Key Source

CCKM provides support for the use of CipherTrust Manager as key source to generate, store, and retrieve the encryption keys that CCKM uses and manages. This section covers how to configure the connection between CCKM and CipherTrust Manager within the **Key Sources** page > **CipherTrust Manager** tab of the CCKM Admin Portal.

Note

Each time you add an additional node to your Key Secure instance that is connected to your (running) CCKM instance or CCKM cluster, a restart of the kmaas app service is required. To restart this service, in the CCKM command line prompt, enter the system command category and then enter the following command: `server --kmaas -restart`. For more information about CCKM CLI commands and the system command category and server command, see ["System Commands" on page 246](#).

Prerequisites

Prior to configuring CipherTrust Manager within the **Key Sources** page, launch and configure CipherTrust Manager. For more information about these prerequisite steps, see ["Install and Configure CipherTrust Manager as Key Source" on page 42](#).

Configuring CipherTrust Manager

To configure **CipherTrust Manager** in the **Key Sources** page > **CipherTrust Manager** tab within the CCKM Admin Portal, perform the following steps:

1. In the **Server** box, enter the hostname or the IP address of CipherTrust Manager.
If the hostname of CipherTrust Manager is not reachable during this configuration, CCKM will display the **IP address** box. If the **IP address** box displays, enter the IP address of CipherTrust Manager in this box.
2. In the **Port** box, enter the port number of CipherTrust Manager web interface. The default value is 443.
3. In the **Connection** box, enter the connection string. The default value is "local_account". The value of "local" is supported. If CipherTrust Manager is integrated with an LDAP server, enter the domain name of the LDAP server.
4. In the **Username** box, enter the username for the CipherTrust Manager admin.
5. In the **Password** box, enter the password for the CipherTrust Manager admin.
6. From the **Certificate** box, select the certificate of CipherTrust Manager. Click **Browse** to select the file.
7. In the **Domain Prefix** box, enter a domain prefix for CipherTrust Manager. The domain prefix is the name of the CipherTrust Manager client. In this case, the client is the CCKM. This name must be unique within CipherTrust Manager.
8. Click **Test Connection** to test the connection between CCKM and CipherTrust Manager. If the test fails, verify that you entered the correct information in the previous steps. Reenter the correct information, if required, and then test the connection again. Otherwise, proceed to the next step.
9. Click **Save**.

Configure DSM as Key Source

CCKM provides support for the use of DSM as key source to generate, store, and retrieve the encryption keys that CCKM uses and manages. This section covers how to configure the connection between CCKM and DSM within the **Key Sources** page > **DSM** tab of the CCKM Admin Portal.

Prerequisites

Prior to configuring DSM within the **Key Sources** page, launch and configure DSM. For more information about these prerequisite steps, see ["Install and Configure DSM as Key Source" on page 43](#).

Configuring DSM

To configure DSM in the **DSM** tab of **Key Sources** page of the CCKM Admin Portal, perform the following steps:

1. In the **DSM Hostname** box, the Fully Qualified Hostname of the DSM, which you can find on the dashboard of the DSM Management Console. Example: dsm-a43.vormetric.com.
2. In the **Domain Prefix** box, enter a unique DSM domain name prefix. Ensure this name conforms to DSM domain name restriction.
3. From the **Certificate** box, select the certificate of DSM . Click **Browse** to select the file.
4. In the **Username** box, enter the username for the DSM admin.
5. In the **Password** box, enter the password for the DSM admin.
6. Click **Save**.

Configure nShield Connect (for Azure and AWS Clouds)

If you are using CCKM to manage keys in Azure or AWS, support for the use of nShield Connect as a key source is available to generate source keys. This section covers how to configure the connection from CCKM to nShield Connect HSM(s), a remote file system (RFS), and the Security World for nShield. This configuration is available within the **Key Sources** page > **nShield** tab of CCKM Admin Portal. The prerequisites steps you are required to perform are also provided in this section.

After successfully configuring the nShield Connect and the associated RFS and Security World in the **Key Sources** page, CCKM runs a number of nShield Connect commands to enroll the CCKM as a new nShield client to the nShield Connect HSM(s), and updates the Security World configuration from the RFS computer. The output of these commands (provided in links) display on the right side of the **nShield Info** page. Be sure to click on each link within the **nShield Info** page to verify that each command ran successfully. If an error message displays, be sure to address the error. For common issues that may arise during the configuration of the connection from CCKM to nShield Connect and the associated RFS and Security World, see ["Troubleshooting nShield Configurations " on page 242](#).

Note

This documentation assumes that you are familiar with using nShield Connect. This documentation also assumes that you have installed nShield Connect, and the associated RFS and Security World. For information on how to install an nShield Connect, and the associated RFS and Security World, refer to the *nShield® Connect Installation Guide*. For information on how to configure nShield Connect, and the associated RFS and Security World, refer to the *nShield Connect User Guide for Unix*.

Prerequisites

Prior to configuring the nShield Connect and the associated RFS and Security World within the Settings page, perform the following steps:

- Install and configure nShield Connect, RFS, and Security World.
- Configure CCKM as an nShield Connect client within the nShield Connect HSM by specifying the IP address of CCKM.

For more information regarding these prerequisite steps, see ["Install and Configure nShield Connect as Key Source" on page 43](#)

Configuring nShield

To configure nShield Connect and the associated RFS and Security World in the **Key Sources** page > **nShield** tab, perform the following steps:

1. From **nShield 1 IP Address**, enter the IP address of the nShield Connect.
2. Select **Add More nShield IP** to add and configure more nShield Connect HSMs.
3. From **nShield RFS IP Address**, enter the IP address of the remote file system (RFS).
4. Click **Add to Security World** to enroll CCKM to Security World. A set of links to the configuration command output display. Click on each link to verify that the commands ran successfully. If an error displays, address it before proceeding to the next step.
5. (Only perform this step, if all of the commands from the previous steps ran successfully.) Click **Save**. This step will restart the hardserver (Security World Software server) on which the CCKM is installed and configured.

Note: If an error occurred during any of the previous steps, and you click **Save**, you will not be able to generate an nShield Connect key within the CCKM Source Key page when you attempt to generate it. An error message displays. For common issues that may arise during the configuration of the connection from CCKM to nShield Connect HSM(s) and the associated RFS and Security World, see ["Troubleshooting nShield Configurations" on page 242](#)

Review Logs

Select **Logs** from the left-hand navigation bar within the CCKM Admin Portal to access the **Logs** page.

This page lists audit logs, describing user actions performed on the Admin Portal, set to Info, Warn, or Error alert levels.

The CCKM administrator can use these logs to review admin activities in CCKM Admin Portal. Note that these logs differ from the logs used by Thales Support for deep troubleshooting (see ["Troubleshooting Logs Files for Thales Support Team" on page 243](#)), and CCKM user audit logs.

From the **Logs** page, you can:

- **Search** by log name or category filter
- **Search** for logs by **Date Range**
- Sort by **Date**, **Event Message**, **Severity**, **Category**, **Event name**, **Where** the activity occurred (which CCKM server), and **Severity** of the alert/Fineness of log capture level.
- Check **More** details to find, for example, uploaded file size, user ID, or timestamp.
- Select a log and **Refresh** it.
- Use **Download Debug Logs** to download debug log files.

Review Health Monitor

Select **Health Monitor** from the left-hand navigation bar within the CCKM Admin Portal to obtain a snapshot of the status of all major components in a CCKM solution.

The screenshot displays the 'Health Check Monitor' page in the CipherTrust Cloud Key Manager interface. The page has a top navigation bar with the product name, a user profile, and a 'Refresh' button. A left sidebar contains navigation icons. The main content area features a table with the following columns: Component, IP Address, Last Check, Status, and Action. Two rows of data are shown, both indicating a 'Good' status for various components including Key Management, Certificate Management, and Cloud Key Manager. The 'Action' column for each row contains a 'Refresh' button.

Component	IP Address	Last Check	Status	Action
Key Management	10.10.10.10	2023-10-10 10:10:10	Good	Refresh
Certificate Management	10.10.10.10	2023-10-10 10:10:10	Good	Refresh

On the **Health Check Monitor** page, the following columns display including ones that provide the status (either “Good” or “Bad”) of each of the major components of a CCKM solution:

- **CCKM Hostname**—The host name of the CCKM instance for which the information is displayed on the **Health Check Monitor** page.
- **IP Address**—The IP address of the CCKM server for which the information is displayed on the **Health Check Monitor** page.
- **Last Checked**—This column displays when the health status of this CCKM server was last checked . A status of “Bad” displayed in this column indicates that the last check took place more than 5 minutes ago, and the CCKM is down.
- **Database**—This column indicates the status of the connection between CCKM and the MongoDB server(s). A status of “Bad” indicates the connection between CCKM and the MongoDB server(s) is down or one or more of the MongoDB servers are down.
- **Data Security Manager, Connection**—This column indicates the status of the connection between the CCKM and DSM. A status of “Bad” indicates the network connection between the CCKM and DSM may be down or the DSM may be down. If a status of “Bad” is displayed for this column, check to see whether the DSM is running and reachable from CCKM.
- **Data Security Manager, Password**—This column indicates whether the configured DSM user password in the CCKM is valid. If the password status is “Bad”, then change it in **DSM** tab of **Key Sources** page of the CCKM Admin Portal.

Note: At a scheduled time (a configured setting in DSM), the DSM Administrator is required to change the DSM user password in the DSM based on security regulations. This change would require an update of this password within CCKM. This is a case when the password in both DSM and CCKM can get out of sync.

- **CipherTrust Manager, Connection**—This column indicates the status of the connection between the CCKM and CipherTrust Manager. A status of “Bad” indicates the network connection between the CCKM and CipherTrust Manager may be down or the CipherTrust Manager may be down. If a status of “Bad” is displayed for this column, check to see whether the CipherTrust Manager is running and reachable from CCKM.
CipherTrust Manager, Password—This column indicates whether the configured CipherTrust Manager user password in the CCKM is valid. If the password status is “Bad”, then change it in **CipherTrust Manager** tab of **Key Sources** page of the CCKM Admin Portal.
- **Cloud Key Manager**—This column indicates the status of the CCKM (software) server running the CCKM User Portal. A status of “Bad” indicates the CCKM server is down, which means the CCKM User Portal is also down. To investigate this issue, check to see whether the kmaas (CCKM) process is running. If the process is not running, restart it by using CLI command **system->server --restart --kmaas**. Also download debug logs by clicking on **Download Debug Logs** on the **Logs** page within the CCKM Admin Portal. You can also download debug logs through the CCKM CLI command **applog->upload**.
- **Configuration Changed**—This column indicates the last time the CCKM configuration was changed on this CCKM.
- **Action**—This column provides the option to remove a listing of a CCKM server that has a status of “Bad” from the **Health Check Monitor** page. The status of a CCKM server displays under the **Cloud Key Manager** column. To remove a CCKM server with a status of “Bad”, find the row that holds the CCKM you wish to remove and click **Remove** (from the **Action** column). The given CCKM server is removed from the **Health Check Monitor** page.

Note

If you remove a listing of a CCKM server that has a status of "Good" from the **Health Check Monitor** page, a background service running on CCKM will add the listing back to the page. This service periodically checks for active CCKM servers, which are connected to MongoDB(s). If an active CCKM server is not yet added to the page, this service adds it as an entry in the **Health Check Monitor** page.

Review Licenses

Select **Licenses** from the left-hand navigation bar within the CCKM Admin Portal to access the **Licenses** page where you can view all of the licenses for your configured clouds.

Note

When a license for a previously configured cloud becomes invalid, you can no longer manage the given cloud's keys through CCKM. The following message displays at the top of the CCKM Admin Portal page, after you upload a license to CCKM that contains an invalid license for at least one cloud:

You currently do not have a valid license for <Cloud name>. If you wish to manage your keys from the mentioned cloud(s) through CCKM, renew your license.

To continue to manage these keys through CCKM, you must renew your license. After renewing your license, upload it to CCKM using the **Upload License** button (top right of the **Licenses** page).

Note

When a license for a configured cloud expires, the following message displays at the top of the CCKM Admin Portal page:

License for <Cloud name> has expired. If you wish to continue to manage your keys from the mentioned cloud(s) through CCKM, renew your license.

After renewing your key, upload it to CCKM using the **Upload License** button (top right of the **Licenses** page).

Cloud Accounts and Subscriptions

(Optional) From the **Cloud Accounts and Subscriptions** box, you can enter the ID numbers (one per line) for each of your cloud accounts, subscriptions, organizations, and projects that you wish to permit to use your CCKM. Enter the following information for each of the supported clouds:

- For AWS, enter the account ID(s).
- For Azure, enter the subscription ID(s).
- For Google Cloud, enter the project ID(s).
- For IBM Cloud, enter the account ID(s).
- For Salesforce, enter the organization ID(s).

If you wish to add a comment about an entry, start a separate line by adding the "#" character before your comment.

Note

If you leave the **Cloud Accounts and Subscriptions** box *blank*, users from other accounts can *also* use your CCKM to manage their cloud keys as long as they have valid cloud login credentials and can access your CCKM User Portal.

Settings

Select **Settings** from the left-hand navigation bar within the CCKM Admin Portal to access the **Settings** page. From the **Settings** page, you can:

- Set or modify the MongoDB settings
- Set or modify the proxy server settings
- Set or modify the SMTP settings
- Set or modify the debug log levels
- Configure user management preferences for the CCKM user portal by allowing CCKM administrators the capability to allow or disallow users to delete source keys from CCKM and enable the masking of user email addresses displayed in CCKM audit logs and reports
- Add and manage the administrator users and set or modify the password security

Configure Database

In the **Database** tab within the **Settings** page, you have the following options to configure MongoDB:

- **Local Database**
- **Remote Database**
- **ReplicaSet Database**
- **MongoDB Atlas**

The CCKM server requires a MongoDB database (MongoDB, MongoDB Atlas, or Percona Server for MongoDB) to store user setup and cloud usage data.

Prior to configuring any of the MongoDB options from the Database page, the MongoDB database server(s) must be deployed. The CCKM solution also includes a built-in version of MongoDB, which is to be used for evaluation and PoC purposes. For the prerequisites *prior* to performing the steps in this section, see [Chapter 5: "Set Up MongoDB" on page 38](#).

When configuring your database, you have the option to perform authentication of the MongoDB users through an AD server using LDAP. Perform the following steps *prior* to enabling this feature in this section: In the AD server, create a user "kmaas_user", and add it to AD group "dbOwner".

Also when configuring your database, it is required that you enter a passphrase. This passphrase is used to derive a master key. The master key, in turn, is used to encrypt all passwords, including the key source (CipherTrust Manager or DSM) and MongoDB passwords, and other sensitive user credentials that are entered into CCKM. The passphrase is shared among all CCKMs within a multi-CCKM environment and must be entered as part of the configuration of each additional CCKM server.

Note

After the passphrase is saved to CCKM, it cannot be changed or deleted. It is important that you note this passphrase for future use. If you configure additional CCKMs for a multi-CCKM environment, then you are required to enter the same passphrase when configuring these CCKMs.

After setting up MongoDB settings successfully, CCKM restarts.

Configure Local Database

If you are using the built-in MongoDB database server within CCKM (for evaluation and PoC purposes only), select to use the local database option from the **Database** page:

1. In **Database** page, select **Local Database**. The following boxes within the **Use Local Database** page are automatically filled with the default values:
Database IP:localhost
Port:27017
Database Name: kmaas
Database Login Credentials:
Username: kmaas_user
Password: *****
2. In the **Configuration Passphrase** box, enter a passphrase. The passphrase must be between 12 and 32 characters long (standard ASCII alphabet characters (a-z, A-Z), integers (0-9), and a limited set of special characters (!@#\$%^&*(){}[])). **It is important that you note this passphrase and store it securely.**
3. Click **Test Connection** to verify that the connection between the MongoDB database server and the CCKM works. If the test is successful, the **Save** button is enabled and you are prompted to save the database setting.
4. Click **Save**. Thereafter, the CCKM Admin and User portals restart. **Be sure to wait for Admin Portal to restart before performing other operations.**

Configure Remote Database

If you are using an external or remote mongoDB database server (single instance) including a Percona Server for MongoDB, select to use the remote database option from the **Database** page:

1. In **Database** page, select **Remote Database**.
2. In the **Configuration Passphrase** box, enter a passphrase. The passphrase must be between 12 and 32 characters long (standard ASCII alphabet characters (a-z, A-Z), integers (0-9), and a limited set of special characters (!@#\$%^&*(){}[])). **It is important that you note this passphrase and store it securely.**
3. In the **Database IP** box, enter the IP address of the MongoDB database server.
4. In the **Database Hostname** box, enter the host name of the MongoDB database server.
5. In the **Port** box, enter the port number of the MongoDB database server.
6. Optional. To use TLS or SSL for a secure connection to the MongoDB, select the **Enable TLS/SSL for secure connection to MongoDB** check box. The **Upload SSL Certificate box** displays. If you performed this step, proceed to the next step. Otherwise, skip to step 8.
7. From the **Upload SSL Certificate** box, click **Choose File** to select the MongoDB Root CA certificate file for secure TLS connection to MongoDB.
8. In the **Database Name** box, enter the name of the MongoDB database.
9. In the **Database Login Credentials** boxes, enter your username and password for the MongoDB database.
10. Optional. Select the **Use LDAP for authentication** box to enable authentication of the MongoDB users through an AD server using LDAP.
11. Click **Check Database Connection** to verify that the connection between the MongoDB database server and

the CCKM works. If the test is successful, the **Save** button is enabled.

- Click **Save**. Thereafter, the CCKM Admin and User portals restart. **Be sure to wait for Admin Portal to restart before performing other operations.**

Configure ReplicaSet Database

If you are using external or remote MongoDB database servers (replica set) including Percona Servers for MongoDB, select to use the replica set database option from the **Database** page:

- In **Database** page, select **ReplicaSet Database**.
- In the **Configuration Passphrase** box, enter a passphrase. The passphrase must be between 12 and 32 characters long (standard ASCII alphabet characters (a-z, A-Z), integers (0-9), and a limited set of special characters (!@#\$%^&*(){}[])). **It is important that you note this passphrase and store it securely.**
- In the **Replica Set Name** box, enter the name of the MongoDB replica set.
- For each instance of the MongoDB, enter the IP address, host name and port number of each database servers in the **Database IP**, **Database Hostname**, and **Port** boxes.
- Optional. If you have more MongoDB instances to add, select the **Add More DB** box and enter the required information.
- Optional. To use TLS or SSL for a secure connection to the MongoDB, select the **Enable TLS/SSL for secure connection to MongoDB** check box. The **Upload SSL Certificate box** displays. If you performed this step, proceed to the next step. Otherwise, skip to Step 8.
- From the **Upload SSL Certificate** box, click **Choose File** to select the MongoDB Root CA certificate file for secure TLS connection to MongoDB.
- In the **Database Name** box, enter the name of the MongoDB database.
- In the **Database Login Credentials** boxes, enter your username and password for the MongoDB database.
- Optional. Select the **Use LDAP for authentication** box to enable authentication of the MongoDB users through an AD server using LDAP.
- Click **Test Connection** to verify that the connection between the MongoDB cluster and CCKM works. If the test is successful, the **Save** button is enabled.
- Click **Save**. Thereafter, the CCKM Admin and User portals restart. **Be sure to wait for Admin Portal to restart before performing other operations.**

Configure MongoDB Atlas

If using MongoDB Atlas database servers, select to use the MongoDB Atlas option from the **Database** page:

- In **Database** page, select **MongoDB Atlas**.
- In the **Configuration Passphrase** box, enter a passphrase. The passphrase must be between 12 and 32 characters long (standard ASCII alphabet characters (a-z, A-Z), integers (0-9), and a limited set of special characters (!@#\$%^&*(){}[])). **It is important that you note this passphrase and store it securely.**
- In the **Atlas Connection String** box, enter the connection string of the MongoDB Atlas. For information on how to obtain this connection string, see ["Obtaining Connection String for Connection Using a MongoDB Driver" on page 1](#).
- In the **Database Name** box, enter the name of the MongoDB Atlas database.
- In the **Database Login Credentials** boxes, enter your username and password for the MongoDB database.

6. Click **Test Connection** to verify that the connection between the MongoDB Atlas and CCKM works. If the test is successful, the **Save** button is enabled.
7. Click **Save**. Thereafter, the CCKM Admin and User portals restart. **Be sure to wait for Admin Portal to restart before performing other operations.**

Set Proxy Settings

Select the **Proxy** tab from the **Settings** page to set (or modify) the proxy server settings.

Note

In the case of a multi-instance CCKM, proxy configuration is not propagated to other instances in the CCKM cluster. You are required to manually configure the proxy settings on each instance in the **Proxy** tab within **Settings** page of the Admin Portal. **If you edit the proxy settings, it is recommended that you logout of the CCKM Admin Portal and then login again to ensure the new settings become effective.**

1. In the **Settings** page, select the **Proxy** tab. The **Proxy Settings** page displays.
2. In the **Hostname** box, enter the hostname or IP address of the proxy server.
3. In the **Port** box, enter the port number of the proxy server.
4. In the **Skip Proxy List** box, enter the hostnames (separated by a pipe symbol "|") that are to pass through the proxy server.
5. (Optional) In the **Username** box, enter the username of the proxy port.
6. (Optional) In the **Password** box, enter the password for the proxy server.
7. Click **Test Connection** to verify that the connection between the proxy server and the CCKM works. If the test is successful, the **Save** button is enabled.
8. Click **Save**.

Set SMTP Settings

In CCKM, SMTP is used to send email notifications for user actions performed and scheduled for the supported clouds. Set optional notification information.

If your SMTP server is configured with a certificate, then you are required to add the SMTP certificate to CCKM **prior** to setting the SMTP settings from the **Settings** page. To add an SMTP server certificate to CCKM, run the following `system` command from the CCKM CLI where `ALIAS` is the name of the SMTP server:

```
system> security --addcert <ALIAS>
```

For more information about the `system` category commands, see [Appendix A: "CCKM CLI Commands" on page 244](#). This appendix provides a high-level overview of each of the command categories that are available in the CCKM.

Note

A single configured SMTP host is common across all configured clouds. Therefore, the SMTP-related configurations that you enter in the SMTP page within the Settings page are applicable to all clouds that you configure.

Select the **SMTP** tab from the **Settings** page to set (or modify) SMTP settings.

1. In the **Settings** page, select the **SMTP** tab.
2. In the **Hostname** box, enter the hostname of the SMTP host.
3. In the **Port** box, enter the port number of the proxy server.
4. In the **From** box, enter the sender's email address.
5. (Optional) In the **Username** box, enter the username of the SMTP port.
6. (Optional) In the **Password** box, enter the password for the SMTP server.
7. click **Save**.

Set Debug Log Level

You can set or modify your the debug log levels (applied to logs that are stored in the file) from the **Debug Logs** tab within the **Settings** page. The following are the debug log levels you can set:

- Trace (Recommended for debugging issues in health check, DSM High Availability (HA), and propagation in cluster)
- Debug (Recommended for debugging any other issues)
- Info (Recommended for normal usage)
- Warn
- Error

To set the debug log levels:

1. Select the **Debug Logs** tab from the **Settings** page.
2. Select the Log level.
3. Click **Save**.

Configure User Portal Preferences

Using the **User Portal Preferences** tab from the **Settings** page, you can configure your user management preferences for the CCKM user portal. From this page, you can do the following:

- Allow or disallow CCKM users to delete source keys from CCKM
- Enable the masking of user email addresses when displayed in CCKM audit logs and reports

Manage Deletion of Source Keys

The **User Portal Preferences** tab from the **Settings** page provides you the capability to allow or disallow CCKM users to delete source keys using the **Allow Source Key Deletion** check box. When this check box is deselected, the **Delete** option in the **Actions** column of the **Key Sources** page is unavailable (not displayed as an option), and the calling of the REST API to delete a source key is denied. By default, the **Allow Source Key Deletion** check box is selected, which allows the deletion of source keys using the **Delete** option or the REST API call.

To manage the deletion of source keys:

1. In the **Settings** page, select the **User Portal Preferences** tab. The **User Portal Preferences** page displays. Within **Source Key Deletion**, the **Allow Source Key Deletion** check box is selected by default.
2. To disallow the deletion of source keys, deselect the **Allow Source Key Deletion** check box.
3. Click **Save** to save the new setting.

Manage Masking of Email Addresses

From the **User Portal Preferences** tab, you have the capability to enable the masking of the user email addresses that display in the **Users** column of CCKM audit logs and reports. By selecting **Mask email addresses** option, all email addresses in audit logs and reports will be obfuscated. This feature is applicable if you are employing email addresses as usernames within your cloud(s). By default, the masking of email addresses is not enabled.

To enable the masking of email addresses:

1. In the **Settings** page, select the **User Portal Preferences** tab. The **User Portal Preferences** page displays. Within **Mask Email Addresses**, the **Do not mask email addresses** option is selected by default.
2. To enable masking, select **Mask email addresses**.
3. In the **Mask start position** box, enter the position within the email address to which to begin the masking. The start position must be greater than 0. For example, if you enter 2 as the start position, masking will begin at the second character within the email address.
4. In the **Number of characters to mask** box, enter the number of characters to mask. The mask length must be greater than 0.
5. Click **Save** to save the new setting.

Manage Users

Select the **User Management** tab from the **Settings** page to manage the users and set or modify the password security.

The CCKM user has the ability to create, enable, disable, and delete additional users from the **User Management** tab.

Note

A user with the username of 'admin' cannot be deleted.

Add User

1. Log in to CCKM, select the **Settings** icon, and choose the **User Management** tab.
2. Click **Add User**.
The **User** dialog box is displayed.
3. Enter the required and optional fields:
 - **Username:** required
 - **First Name, Last Name, Email:** optional
 - **Password/Confirm password:** required. Must be 8 to 20 characters, with at least one capitalized letter,

one lowercase, and one digit.

4. Select the **Active** check box (unless this user should be disabled).
5. Click **Submit**.

Disable User

There are two places from which to disable an administrator user:

1. Within the **Actions** column from the **User Management** page, click **Disable** next to the user's name.
2. Within the **Actions** column from the **User Management** page, click **Update** next the user's name.
3. From the **User** dialog box, you can change First Name, Last Name, Email, and clear the **Active** check box.
4. Click **Submit**.

Enable User

There are two places from which to enable an administrator user:

1. Within the **Actions** column from the **User Management** page, click **Enable** next to the user's name.
2. Within the **Actions** column from the **User Management** page, click **Update** next the user's name.
3. From the **User** dialog box, you can change First Name, Last Name, Email, and clear the **Active** check box.
4. Select the **Active** check box.
5. Click **Submit**.

Delete User

Note

The "admin" user cannot be deleted.

1. Within the **Actions** column from the **User Management** page, click **Delete** next to the username of the user you wish to delete. The **Delete User Confirmation** dialog box displays.
2. Click **Confirm** to confirm the deletion of the username.

Reset Password

The password of the "admin" user cannot be reset through the CCKM Admin Portal. You must use the CCKM CLI to reset password (**resetwebadmin**) of this user. For more information, see [Appendix A: "CCKM CLI Commands" on page 244](#).

To reset the password of a user (besides the "admin" user):

Note

By default, the password you enter below will expire soon after entering it. You are prompted to change this password when you log into the Admin Portal *after* resetting the password.

1. Within the **Actions** column from the **User Management** page, click **Reset** next to the username of the user whose password you wish to reset. The **Enter New Password** dialog box displays.
2. Enter the new password and then enter the password again to confirm it.
3. Click **Confirm**. By default, this password will expire after entering it.
4. Log into the Admin Portal using the new password. You are prompted to change the password.

Update Admin Password Security

Any CCKM user can update the Admin password security information.

1. Log in to CCKM, select the **Settings** icon, and choose the **User Management** tab.
2. Click **Password Security**.
The **Update password security** dialog box is displayed.
3. In the **Password duration (days)** box, enter the number of days in which the admin password expires. The password expiration interval is applied globally to each administrator account. If the administrator does not change the password prior to the expiration, the administrator must reset the password immediately the next time the administrator logs in. The expiration interval is an integer between 6 and 365. The default is 90. The password duration must be set to a value greater than 'Password Expiration Notification'.
4. In the **Password history** box, enter the number of times the current password is permitted to be reused. CCKM maintains a password history. You cannot use the same password more than once per the set limit. The default is 4, and the maximum value that can be set is 12. You can set this value to '0' to permit reuse of the current password.
5. In the **Minimum password length** box, set the minimum number of characters, including blank spaces that must be in a password. The minimum password length is an integer between 8 and 256. The default is 8.
6. In the **Password expiration notification (days)** box, enter the number of days prior to the password expiration at which to begin telling the administrator that their password is about to expire. Administrators are notified of the impending expiration. The notification interval is an integer between 6 and 31. The default is 6.
7. Ensure the **Required upper case** check box is selected to enable the requirement of having at least one uppercase alphabet character in the administrator password. This check box is enabled by default. To disable this requirement, ensure this check box is deselected.
8. Ensure the **Required numbers** check box is selected to enable the requirement of having at least one integer in the administrator password. This is enabled by default. This check box is enabled by default. To disable this requirement, ensure this check box is deselected.
9. Ensure the **Require special characters** check box is selected to enable the requirement of having at least one special character (!@#\$%^&*(){}[]) in the administrator password. This check box is enabled by default. To disable this requirement, ensure this check box is deselected.

10. In the **Maximum number of login tries** box, enter the maximum number of unsuccessful login attempts before disabling access for a set interval of time. The login page for the CCKM Admin Portal becomes inoperable and ignores further login attempts by an administrator for the specified interval. The range is between 1 and 10. The default number is 3.
11. In the **User lockout time (minutes)** box, enter the number of minutes to wait before allowing administrators to login. The default is 30 minutes
12. In the **Auth token validity duration (minutes)** box, enter the number of days in which the admin authentication token expires. The authentication token expiration interval is applied globally to each administrator account. If the administrator does not change the token prior to the expiration, the administrator must reset the password immediately the next time the administrator logs in. The expiration interval is an integer between 5 and 60. The default is 60.
13. Click **Submit**.

Chapter 13: CCKM for AWS Key Management

Overview	89
Prerequisites on AWS	90
Access AWS Key Management	91
Access CCKM for AWS	91
AWS CCKM User Portal	91
Home	93
Keys	93
Reports	106
Schedules	107
Logs	110
Key Sources	111
Settings	113

Overview

CCKM for AWS offers risk management and assessment of AWS customer-managed Customer Master Keys (CMKs) across AWS services. It provides operational visibility of AWS CMKs across AWS services, as well as tools to manage the complete life cycle of keys from on premise or from the cloud.

CCKM for AWS Key Management involves a few concepts specific to AWS:

- **Source key:** Source keys are generated from the supported CCKM key sources (CipherTrust Manager, DSM, or nShield Connect) and can be uploaded to AWS.
- **AWS regions:** Source keys can be uploaded to only one AWS region (chosen from a drop-down menu).
- **Customer Master Key (CMK):** There are two types of CMKs—AWS-managed or customer-managed. The CMK origin can be AWS_KMS (generated in KMS), or EXTERNAL (imported by user). CCKM can manage the life cycle of all keys, but owns the key material of only EXTERNAL keys imported by CCKM itself. CMKs are region-specific, meaning CMK operations and APIs work on CMKs in only one region at a time.
- CMKs can be disabled and enabled. CMKs cannot be deleted immediately, but can be scheduled for deletion 7 to 30 days in advance. CMKs can be identified by Key Id, Key ARN, or alias.
- **Alias:** Use a reader-friendly alias when uploading a source key to AWS to make searching for key names more intuitive. Multiple aliases can be assigned to the same CMK.

- **Delete key:** Unlike some other key management cloud services, the AWS CMK APIs do not provide a direct mechanism for creating backup keys. Therefore, the process of key deletion in the CCKM has several phases.
 - A source key that has been generated in a key source (CipherTrust Manager, DSM, or nShield Connect) but not uploaded or used in AWS, can be deleted from the **Key Sources** page by clicking **Delete Key**.
 - Once a source key has been uploaded to AWS, deleting it will only remove it from the **Key Sources** page, but not delete it from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as the key source). CCKM will keep the key in CipherTrust Manager, DSM or MongoDB (depending on the key source you are using) as a backup. For the uploaded AWS key, you can do **“Delete > Delete Key Material Now”** to remove the key material from AWS, and then later do **“Import”** to re-import (restore) the key material in AWS.

After a source key has been uploaded to AWS, to delete it from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as the key source), you must delete both the key uploaded to AWS and the key from the **Key Sources** page. First, schedule a deletion of the uploaded AWS key. After the AWS key is deleted on AWS, synchronize your keys on CCKM. Then delete the source key from the **Key Sources** page.

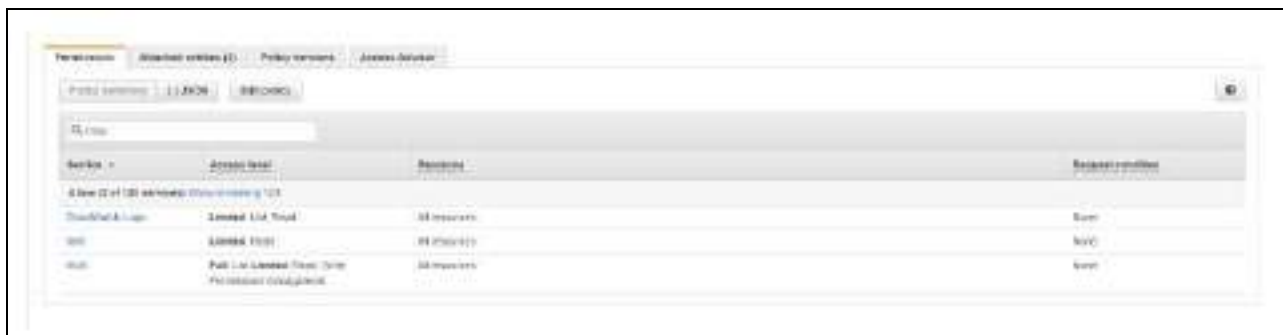
- **CMK key rotation:** The rotation of CMKs is encouraged as part of cryptographic best practices. Note that CCKM supports key rotation for keys that have an alias regardless whether it is a CCKM key or a native key. In CCKM, key rotation involves creating and uploading new external CMK and moving the alias(es) from the old (previous key) to this newly uploaded key (current key). This key rotation does not affect the components of the current CMK, such as the key ID, key ARN, and region.

CCKM allows for a manual or a scheduled rotation of a CMK. Scheduled rotation requires that the key rotation feature be configured in the **Settings** page within CCKM prior to enabling this feature in the **Rotate AWS Key** page.

Prerequisites on AWS

You must have:

- KMSFullAccess
- IAM GetUser permissions
- CloudWatch Logs permissions (Limited: List, Read All Resources)



Access AWS Key Management

You can either enter your AWS secret key ID and access key when prompted, or upload an AWS credentials file on the login page. AWS Credentials file should be .json format (can be written in notepad and saved for upload).

For example:

```
{  
  "aws_access_key_id": "ABCDEFGH1234hiJKLMN",  
  "aws_secret_access_key": "fgreg1234567tg\refg35"  
}
```

Access CCKM for AWS

For details about accessing CCKM as a CCKM user for the CCKM service for AWS Key Management, see [Chapter 3: "About CCKM Admin and User Portals" on page 27](#). This chapter includes information on the ways in which to access the CCKM service for Key Management.

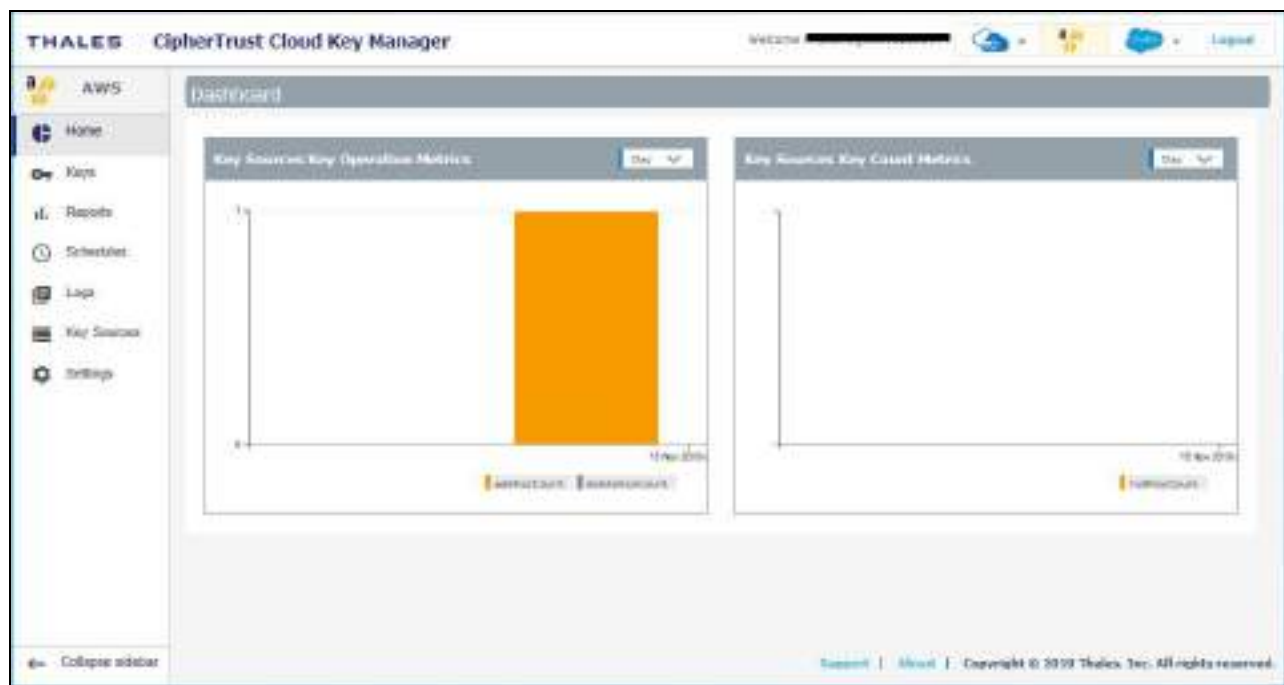
AWS CCKM User Portal

The CCKM User Portal for AWS contains the following menu options in the left-hand navigation bar:

- **Home:** Home page displays key operation and count metrics.
- **Keys:** Keys page displays key management overview and provides the Upload and Synchronize functionality, and the Delete, Restore, and Auto Rotate key operations available in CCKM.
- **Reports:** All Reports overview page links to detailed individual reports.
- **Schedule:** In Schedule, you can schedule automatic key rotation, key rotation based on key expiration, and key synchronization.
- **Logs:** Logs page lists individual logs that can be searched, sorted, and viewed.
- **Key Sources:** In **Key Sources**, you can generate keys in CCKM to upload to AWS. CCKM provides support for the use of CipherTrust Manager and DSM as key sources to generate, store, and retrieve the encryption keys that CCKM uses and manages. CCKM also provides support for the use of nShield Connect HSM as a key source to generate the encryption keys that CCKM uses and manages. These keys are then stored in MongoDB. Also supported in **Key Sources** for all Azure clouds are BYOK and PFX file upload.

Note: CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

- **Settings:** The **Settings** page allows you to add the email addresses of the recipients who are to receive email notifications and configure the settings for email notifications, remote syslog server, and proxy server. You can also view the release number of the CCKM running from this page.



To display the context of which Key Management service you are currently logged into within CCKM, the icon associated with the Key Management service displays at the top of the left-hand navigation bar.

The arrow button at the bottom of the navigation bar allows you to expand the bar to view the names of the menu options and collapse it to view only the icons associated with the menu options. Click the arrow button to either expand or collapse the navigation bar.

On the top-right side of the CCKM portal for AWS, the following options are available:

- **Azure Cloud Service Context Switcher:** This context switcher allows you to login into another Azure cloud service different from the current Azure cloud service into which you are currently logged. If you select the Azure cloud in which you are currently logged, the Multi-account Login/Logout window for the Azure account(s) displays. For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **AWS Login:** If you are not already logged into AWS, clicking on the AWS Login icon brings you to the AWS Password Login page where you can enter your AWS credentials to access the CCKM AWS cloud service (from the current cloud service in which you are currently logged). If you are already logged into AWS, clicking on the AWS Login icon brings you to the Multi-account Login/Logout window for the AWS account(s). For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **Google Cloud Service Context Switcher:** This context switcher allows you to login into another Google Cloud organization different from the current Google Cloud organization into which you are currently logged. If you select the Google Cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Google account(s) displays. For more information about the **Multi-account Login/Logout** window, see ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **Salesforce Cloud Service Context Switcher:** This context switcher allows you to login into another Salesforce cloud service different from the current Salesforce cloud service into which you are currently logged. This is Salesforce, Salesforce Government, or Salesforce Sandbox. If you select the Salesforce cloud in which you are currently logged, the Multi-account Login/Logout window for the Salesforce account(s) displays. For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple](#)

[accounts or Multiple Users per Cloud Service in CCKM UI" on page 32.](#)

- **Logout:** Logout from CCKM.

Home

The **Home** page provides a dashboard to display key operation and count metrics.

Keys

The **AWS Keys** page allows Cloud administrators to view all of the keys and manage them within the boundary of granted permissions. The keys or key vaults that do not have List permission for the respective Cloud administrator are not listed.

At the top right of the **Keys** page, the following buttons are available:

- **Export Keys:** Allows you to export the contents of the Keys list into a report in a CSV format.
- **Alias View:** Allows you to view the Keys list by the key alias names.
- **New Key:** Allows you to upload an external key (BYOK) or create a native key.
- **Synchronize:** Allows you to download any keys that were created in AWS, outside of the CCKM portal, into CCKM.

In this section, how to create, delete, schedule delete, import, update, rotate, synchronize, and export AWS keys are described. Also described in this section is how to download a public key of an AWS asymmetric key.

You can view the **AWS Keys** page by the **Keys** view or the **Alias** view. Here is the **Keys** view with **Key ID** as the first column.

To view the **AWS Keys** page by the **Alias** view, click **Alias View**. Any alias you assigned to the source keys when uploading these keys to AWS is shown in the Alias column, the first column listed within this view.

A **Search** box is available at the top the **AWS Keys** page allowing for a search of a specific key by entering any one of the following parameters:

- **Key ID:** The globally unique identifier for the CMK.
- **Source Key:** The name of the source key. This is the key generated in the key source you are using (CipherTrust Manager, DSM, or nShield Connect).
- **Region:** The AWS Region to which the key is uploaded.
- **Key Type:** The supported CMK key types are Symmetric and Asymmetric. Each key type has its relevant key usage(s) and key spec option(s).
- **Key Usage:** The cryptographic operations for which the CMK is used. The valid values are ENCRYPT_DECRYPT and SIGN_VERIFY. The Symmetric key type uses ENCRYPT_DECRYPT. The Asymmetric key type uses either ENCRYPT_DECRYPT or SIGN_VERIFY.
- **Key Spec:** A CMK property that represents the cryptographic configuration of a given CMK. Supported key specs are SYMMETRIC_DEFAULT, RSA, and elliptic curve (ECC). The Symmetric key type uses SYMMETRIC_DEFAULT. The Asymmetric key type uses either RSA or ECC.
- **ARN:** Amazon Resource Name of the CMK.
- **ARN description:** The description given to the CMK's ARN.

- **Origin:** The source of the key material. The valid values are:
 - **AWS_KMS:** Key material was created by AWS KMS.
 - **EXTERNAL:** Key material was imported from CCKM or another key management system.
- **Key Manager:** The manager of the CMK. The valid values are:
 - **AWS:** AWS-managed CMK.
 - **CUSTOMER:** Customer-managed CMKs that you create, use, and manage.
- **Key State:** The state of the CMK. The valid values are:
 - Enabled
 - Disabled
 - PendingDeletion
 - PendingImport
- **Expiration Model:** Specifies whether the key material of the CMK expires. This value is present only when the source of the key material was imported from CCKM or another key management system. The valid values are:
 - **KEY_MATERIAL_EXPIRES**
 - **KEY_MATERIAL_DOES_NOT_EXPIRE**
- **Tags:** The label assigned to the key, which consists of a customer-defined key and an optional value. For example, department finance.
- **Key Aliases:** The alias or aliases assigned to the key.

The **Advanced** tab allows you to filter your search on the each of the columns displayed on the **AWS Keys** page (with the exception of the **Action** column) and **Tags**.

- Key ID
- Region
- Key Type
- Key Usage
- Key Spec
- Creation Date—Enter START DATE and END DATE.
- Enabled—Available options from the drop-down menu are ENABLED and DISABLED.
- Key State—Available options from the drop-down menu are Enabled, Disabled, PendingImport, PendingDeletion
- Scheduled Deletion Date—Select a START DATE and END DATE.
- Key Material Expiration Date—Select a START DATE and END DATE.
- Key Material Origin
- Rotation Status—Available options from the drop-down menu are ACTIVE and ARCHIVED.
- Tags

Each of the columns are described in more detail below in this section.

To select to filter based on a specific column from the **AWS Keys** page, click **Advanced**. The **Filters** section displays. From the column filter drop-down menu, select the name of the column on which to filter your search. For example, select **Region**. In the **Search** box, enter the value on which to search (for example, **US East (Ohio)** as the region name) and click **Add**. For the **Enabled**, **Key State**, and **Rotation Status** columns, a drop-down menu is available from which to select a value for the search. Click the **down** arrow, select a value from the available options, and then click **Add**. For the **Creation Date**, **Scheduled Deletion Date**, and **Key Material Expiration Date**, select a start and end date for the search and then click **Add**. The specified search filter is then added to **Active Filters**. In the example, **Region: US East (Ohio)** is added to **Active Filters**. The results of the search displays in the **AWS Keys** table. You can narrow your search by selecting more columns to add to the filter. To clear the filtered results, from **Active Filters**, click on the “X” (or the **Delete** button) for each of the named filters.



The following columns display on the AWS Keys page:

- **Key ID:** The globally unique identifier for the CMK.
- **Region:** The AWS Region to which the key is uploaded.
- **Key Type:** The supported CMK key types are Symmetric and Asymmetric. Each key type has its relevant key usage(s) and key spec option(s).
- **Key Usage:** The cryptographic operations for which the CMK is used. The valid values are ENCRYPT_DECRYPT and SIGN_VERIFY. The Symmetric key type uses ENCRYPT_DECRYPT. The Asymmetric key type uses either ENCRYPT_DECRYPT or SIGN_VERIFY.
- **Key Spec:** A CMK property that represents the cryptographic configuration of a given CMK. Supported key specs are SYMMETRIC_DEFAULT, RSA, and elliptic curve (ECC). The Symmetric key type uses SYMMETRIC_DEFAULT. The Asymmetric key type uses either RSA or ECC.
- **Creation Date:** Indicates the date on which the key was created.
- **Enabled:** Indicates whether the key is enabled for use.
- **Key State:** The state of the CMK. The valid values are:
 - Enabled
 - Disabled
 - PendingDeletion
 - PendingImport

- **Scheduled Deletion Date:** Indicates the date on which the key is scheduled to be deleted.
- **Key Material Expiration Date:** Indicates the date on which the key material is set to expire.
- **Key Material Origin:**
 - If the key was uploaded from CCKM, then the value is INTERNAL(<source_key_name>).
 - If the key was not uploaded from CCKM as a BYOK key, then the value is EXTERNAL.
 - If the key material was created by AWS KMS, then the value is KMS.
- **Rotation Status:** Indicates whether the key is in active rotation schedule or if it has been rotated in the past.
 - Auto—if the autorotation is currently enabled.
 - Active—if the key is the most latest in the rotation sequence.
 - Archived—if the key has been rotated out.
- **Actions:** Allows you to select any of the following actions on a given key:
 - Rotate
 - Delete
 - Schedule Delete
 - Cancel Schedule Delete
 - Import
 - Update
 - Download Public Key

Create a Key

You have two options to create a new key in AWS KMS using CCKM:

- ["Create a Native Key" below](#)
- ["Upload your Own Key \(BYOK\)" on page 98](#)

Part of the process of creating a new key in AWS KMS using CCKM with either option is adding it to a AWS region and selecting a key type of symmetric or asymmetric to use. Each key type is associated with a key usage and key spec options.

The following are the supported key usages and key specs in AWS KMS:

- ENCRYPT_DECRYPT: used for Symmetric encryption. The supported key spec is SYMMETRIC_DEFAULT.
- ENCRYPT_DECRYPT: used for Asymmetric encryption. The supported key spec is RSA.
- SIGN_VERIFY: used for Asymmetric signing. The supported key specs are RSA and elliptic curve (ECC).

Create a Native Key

Create a native key in AWS KMS using CCKM. After CCKM creates a native key, it displays the key as **KMS** under the **Key Material Origin** column within the **Keys** page in CCKM.

For more information, see the following sections:

- ["Create a Native Key \(Asymmetric Key\)" on the facing page](#)
- ["Create a Native Key \(Symmetric Key\)" on the facing page](#)

Create a Native Key (Asymmetric Key)

To create a native key that has a key type of Asymmetric used for asymmetric signing or asymmetric encryption:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. From the **Keys** page, select the **New Key** button and then **Create**. The **Create Key** dialog box displays.
3. From the **Region** box, select the AWS Region to which you will add the new key.
4. From **Key Type**, select the key type of **Asymmetric**.
5. From **Key Usage**, select **Encrypt and decrypt** if you will use this key to encrypt and decrypt data. Select **Sign and verify** if you will use this key to sign and verify data.
6. If you selected **Encrypt and decrypt**, then select **RSA_2048**, **RSA_3072**, or **RSA_4096** from the **Key Spec** box for the key spec to apply to the new key.
If you selected **Sign and verify**, then select **RSA_2048**, **RSA_3072**, **RSA_4096**, **ECC_NIST_P256**, **ECC_NIST_P384**, **ECC_NIST_P521**, or **ECC_SECG_P256K1** from the **Key Spec** box for the key spec to apply to the new key.
7. In the **Alias** box, assign an alias for intuitive key name searching and for use in key rotation. Required by AWS Management Console.

Note: In order to rotate keys, you must assign an alias when creating, uploading, or updating a key.

8. (Optional) In the **Description** box, enter a description of the key.
9. (Optional) In the **Tags** boxes, enter the tag name and value of the key and then click the **+** button.
10. (Optional) In the **Allow Access for External Accounts** box, enter the account ID of an external AWS account with which the key is shared and then click the **+** button. This step is applicable only if you are to share this key with an external AWS account.
11. Click **Create**.
The new key is displayed in the list.

Create a Native Key (Symmetric Key)

To create a native key that has a key type of Symmetric used for symmetric encryption/decryption:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. From the **Keys** page, select the **New Key** button and then **Create**. The **Create Key** dialog box displays.
3. From the **Region** box, select the AWS Region to which you will add the new key.
4. From **Key Type**, select the CM key type of **Symmetric**.
The **Key Usage** of **Encrypt and decrypt** and **Key Spec** of **SYMMETRIC_DEFAULT** display and are not available for modification.
5. In the **Alias** box, assign an alias for intuitive key name searching and for use in key rotation. Required by AWS Management Console.

Note: In order to rotate keys, you must assign an alias when creating, uploading, or updating a key.

6. (Optional) In the **Description** box, enter a description of the key.
7. (Optional) In the **Tags** boxes, enter the tag name and value of the key and then click the **+** button.

8. (Optional) In the **Allow Access for External Accounts** box, enter the account ID of an external AWS account with which the key is shared and then click the **+** button. This step is applicable only if you are to share this key with an external AWS account.
9. Click **Create**.
The new key is displayed in the list.

Upload your Own Key (BYOK)

Upload the keys created from the **Key Sources** page of CCKM to the AWS KMS. These are the keys (of algorithm type AES256) you generated from the key source you are using (CipherTrust Manager or DSM). These are your "BYOK" keys.

To upload your BYOK to CCKM:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New** button and then **Upload**. The **Upload a Key** dialog box displays.
3. From the **Region** box, select the AWS Region to which you will upload the key.
4. From the **Key Provider** box, select **CipherTrust Manager**, **DSM**, or **nShield** depending on the key source you are using for the key.
5. From the **Source Key** box, select one of the available source keys. The keys available in the drop-down menu are the keys generated from CipherTrust Manager, DSM, or nShield Connect depending on the key source you are using.
6. In the **Alias** box, assign an alias for intuitive key name searching and for use in key rotation. Required by AWS Management Console.

Note: In order to rotate keys, you must assign an alias, when either uploading or using the Update Key function.

7. (Optional) Enter the expiration date of the key material in the **Key Material Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
8. (Optional) In the **Description** box, enter a description for the key you are uploading.
9. (Optional) In the **Tags** boxes, enter the tag name and value of the key and then click the **+** button.
10. (Optional) In the **Allow Access for External Accounts** box, enter the account ID of an external AWS account with which the key is shared and then click the **+** button. This step is applicable only if you are to share this key with an external AWS account.
11. Click **Upload to AWS**.
The uploaded key is displayed in the list.

Delete a Key

You have the following options to delete a key:

- Delete a BYOK
- Schedule Delete a BYOK or Native key

Delete a BYOK

You can delete a BYOK created from your key source and uploaded to AWS. These keys display as **EXTERNAL** under the **Key Material Origin** column on the **Keys** page. From the **Keys** page, deleting your BYOK deletes it from AWS. AWS requires a minimum wait period of seven days before it fully deletes the entire key from AWS.

To delete a BYOK (external key) from AWS:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, find the key you wish to delete.
3. From the **Actions** column, click **Delete**. The **Delete AWS Key Material** dialog box displays.
A warning message displays indicating that this action will delete AWS CMK "key ID" from AWS Key Management Service. The message also indicates to type the confirmation phrase (I wish to delete the key material from AWS KMS) in the text box if you wish to delete the key.
4. Type or copy/paste the confirmation phrase in the Warning box.
5. Click **Delete**.

Schedule Delete

From the **Keys** page, you have the option to schedule to delete a key from AWS KMS permanently after the designated pending period of seven to 30 days is reached. During key deletion, the key material and metadata will be deleted simultaneously from AWS KMS. **After the key is deleted, it cannot be restored, and the data encrypted with this key cannot be decrypted.** During the pending period, you have the option to cancel the scheduled key deletion using the **Cancel Delete** option from the **Action** column.

Schedule Delete Key

To schedule to delete a key from AWS KMS:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. Find the row that holds the key you wish to schedule to delete.
3. From the **Actions** column, click **Schedule Delete**. The **Schedule delete AWS Key** dialog box displays. A warning message displays in the dialog box indicating that this action will delete AWS CMK "<key ID>" from AWS Key Management Service. A message also indicates to type the confirmation phrase (I wish to delete the key, and I understand it cannot be restored) in the text box, if you still wish to delete the key.
4. In **Schedule to delete key after < > days**, enter a number between **7** and **30**, which is the number of days (designated pending period) after which the key is deleted.
5. Type or copy/paste the confirmation phrase in the dialog box.
6. Click **Delete**.

Import a Key

See ["Delete a Key" on the previous page](#). If the source key originated in CCKM and the key material has been deleted from AWS but the full deletion has not completed, use **Import** to re-import the key from CCKM to AWS.

Update Key

Use **Update** to do the following:

- Create and/or assign an alias to a key.
- Adjust the Enabled status, Description, or Tags.
- Add the account ID of an external AWS account with which the key is shared.
- Add an AWS key policy used to share the key with an external AWS account.

Note

Keys cannot be updated when they are in a PendingImport or PendingDeletion state.

To update a key:

1. On the **Keys** list page (Keys View or Alias View), click **Update** in the **Action** column for a particular key.
2. From the **Key Properties** tab, select an available alias or enter a new alias name in the **Alias** dialog. Click **+**.
3. Adjust the **Enabled** check box, **Description**, or **Tags**.
4. Click **Update**.
5. (Optional) From the **Key Policy** tab, in **Default View**, in **Allow Access for External Accounts**, click **+** and enter the account ID of an external AWS account with which the key is shared. This step is applicable only if you are to share this key with an external AWS account and you have not yet customized the AWS key policy.
You also have the option to enter an AWS key policy that allows you to share this key with an external AWS account. From the **Key Policy** tab, within **Key Policy** in **Policy View**, enter the AWS key policy copied from the associated JSON file.
6. Click **Save Accounts**.

Download Public Key (Asymmetric Key Only)

CCKM provides support for downloading a public key of an AWS asymmetric key for the asymmetric encryption and signing operations. The public key is made available in a Privacy Enhanced Mail (PEM) file format. The default file name is Publickey-<Key ID>.pem where <Key ID> is the key ID of the asymmetric key.

To download the public key of an AWS asymmetric key within CCKM:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. Find the row that holds the asymmetric key for which you wish to download the public key.
3. Select **Download Public Key** from the **Actions** drop-down menu.
The public key file is downloaded to the **Downloads** directory within your laptop or desktop.

Note: Save the file to a secure location after it is downloaded.

Rotate Key

CCKM allows key rotation for keys that have a key alias. When you rotate a key, CCKM uploads a new source key to AWS and moves the alias from this key to the new AWS key. After key rotation, all your applications that are using this key by its alias will use the new key to encrypt future data. To decrypt existing encrypted data, these applications continue to use the old key.

To rotate a key, select the **Rotate Key** action from the **Actions** column of the **All Keys** list. In the **Rotate AWS Key** dialog box, you can choose to **Rotate Now** or **Auto Rotate**.

Rotate Now allows you upload a source key to AWS immediately and move the key alias from this key to the new key. Note that you can only choose a new source key that is **not** in the rotation chain.

Auto Rotate allows you to rotate this key using a schedule of your choice. You must already have created a KEY_ROTATION schedule before you can select an autorotation of a key using **Auto Rotate**. See ["Schedule Key Rotation" on page 107](#) for more information about creating a schedule for the key autorotation. With **Auto Rotate**, CCKM will generate a new key in your key source, upload it to AWS, and then move the key alias from this key to the new key.

Note

Any key that is shared among different IAM users within a single AWS account and is scheduled for autorotation by one IAM user cannot be scheduled for autorotation again by another IAM user. Any key that is scheduled for autorotation displays the status of "AUTO" within the **Rotation Status** column of the **Keys** page. If you attempt to schedule autorotation using **Rotate Key** within the **All Keys** list with a key already scheduled, a message displays indicating that the key is already scheduled for rotation with the scheduled interval, date, and time at which the rotation is to take place. In addition, the name of the user responsible for scheduling the rotation displays.

For more information, see the following sections:

- ["Manually Rotate a Key \(Symmetric Key\)" below](#)
- ["Enable Autorotation of Key \(Symmetric Key\)" on the next page](#)
- ["Manually Rotate a Key \(Asymmetric Key, Encrypt, and Decrypt\)" on the next page](#)
- ["Manually Rotate a Key \(Asymmetric Key, Sign, and Verify\)" on page 103](#)
- ["Enable Autorotation of Key \(Asymmetric Key, Encrypt, and Decrypt\)" on page 103](#)
- ["Enable Autorotation of Key \(Asymmetric Key, Sign, and Verify\)" on page 104](#)

Manually Rotate a Key (Symmetric Key)

To manually rotate a symmetric key:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. From the **Keys** page, choose the **Alias** or **Keys** view.
3. Find the row that holds the key you wish to rotate and select **Rotate Key** from the **Actions** column. The **Rotate AWS Key** dialog box is displayed.
4. Select the **Rotate Now** tab.
5. From **Key Provider** box, select **CipherTrust Manager**, **DSM**, or **nShield** depending on the key source you are using for the key.
6. From the **Source Key** list, select a new source key.
Only source keys that have not yet been uploaded to AWS are listed. The **Aliases** box displays the alias associated with the selected key.
7. Deselect the **Disable Encrypt permission on Current key** check box, if you want to continue using the key that has been rotated out to encrypt new data. By default, this check box is selected. Best practice is not to use the old key that has been rotated out to encrypt new data. Strictly maintain the rotated-out key so that KMS can continue using it to decrypt the previously encrypted data by that key.

8. (Optional) Select the **Key Material Expiration Date** check box to select a date and time on which the key material is set to expire. Select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.
9. (Optional) Enter a description of the new CMK key in the **Description** box.
10. Click **Rotate**.

Enable Autorotation of Key (Symmetric Key)

To enable autorotation of a symmetric key:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. From the **Keys** page, choose **Alias** or **Keys** view.
3. Find the name of the key for which to enable autorotation and select **Rotate Key** from the Actions drop-down menu. The **Rotate AWS Key** dialog box is displayed.
4. Select the **Auto Rotate** tab.

Note: The **Enable autorotation on this key?** toggle and **Schedule** drop-down menu are not applicable to enabling autorotation of this key before it expires.

5. From the **Key Provider** box, select **CipherTrust Manager**, **DSM**, or **nShield** depending on the key source you are using for the key.
6. Deselect the **Disable Encrypt permission on Current key** check box, if you want to continue using the key that has been rotated out to encrypt new data. By default, this check box is selected. Best practice is not to use the old key that has been rotated out to encrypt new data. Strictly maintain the rotated-out key so that KMS can continue using it to decrypt the previously encrypted data by that key.
7. Click **Submit**.

Manually Rotate a Key (Asymmetric Key, Encrypt, and Decrypt)

To manually rotate an asymmetric key used to encrypt and decrypt data:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. From the **Keys** page, choose the **Alias** or **Keys** view.
3. Find the row that holds the key you wish to rotate and select **Rotate Key** from the **Actions** column. The **Rotate AWS Key** dialog box is displayed.
4. Select the **Rotate Now** tab. The **Aliases** box displays the alias associated with the selected key.
5. Deselect the **Disable Encrypt permission on Current key** check box, if you want to continue using the key that has been rotated out to encrypt new data. By default, this check box is selected. Best practice is not to use the old key that has been rotated out to encrypt new data. Strictly maintain the rotated-out key so that AWS KMS can continue using it to decrypt the previously encrypted data by that key.
6. Deselect the **Disable download permission on current public key** check box, if you wish to make available the option to download a public key for the given asymmetric key (by making available the **Download Public Key** option from the **Actions** drop-down menu). By default, this check box is selected. Best practice is to disable the download permission on the current public key.
7. (Optional) Select the **Key Material Expiration Date** check box to select a date and time on which the key material is set to expire. Select a start date and time from a date and time selector by clicking the calendar

button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.

8. (Optional) Enter a description of the new CMK key in the **Description** box.
9. Click **Rotate**.

Manually Rotate a Key (Asymmetric Key, Sign, and Verify)

To manually rotate an asymmetric key used to sign and verify data:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. From the **Keys** page, choose the **Alias** or **Keys** view.
3. Find the row that holds the key you wish to rotate and select **Rotate Key** from the **Actions** column. The **Rotate AWS Key** dialog box is displayed.
4. Select the **Rotate Now** tab. The **Aliases** box displays the alias associated with the selected key.
5. Deselect the **Disable sign permission on current key** check box, if you want to continue using the key that has been rotated out to sign and verify new data. By default, this check box is selected. Best practice is not to use the old key that has been rotated out to sign and verify new data.
6. (Optional) Select the **Key Material Expiration Date** check box to select a date and time on which the key material is set to expire. Select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.
7. (Optional) Enter a description of the new CMK key in the **Description** box.
8. Click **Rotate**.

Enable Autorotation of Key (Asymmetric Key, Encrypt, and Decrypt)

To enable autorotation of an asymmetric key used to encrypt and decrypt data:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. From the **Keys** page, choose **Alias** or **Keys** view.
3. Find the name of the key for which to enable autorotation and select **Rotate Key** from the **Actions** drop-down menu. The **Rotate AWS Key** dialog box is displayed.
4. Select the **Auto Rotate** tab.
5. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.

Note: If the toggle for **Enable autorotation on key expiration?** is displayed, skip it. This option displays only if the key has an expiration date associated with it. This option is not applicable to enabling autorotation of the key.

6. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
7. Deselect the **Disable Encrypt permission on Current key** check box, if you want to continue using the key that has been rotated out to encrypt new data. By default, this check box is selected. Best practice is not to use the old key that has been rotated out to encrypt new data. Strictly maintain the rotated-out key so that KMS can continue using it to decrypt the previously encrypted data by that key.

8. Deselect the **Disable download permission on current public key** check box, if you wish to make available the option to download a public key for the given asymmetric key (by making available the **Download Public Key** option from the **Actions** drop-down menu). By default, this check box is selected. Best practice is to disable the download permission on the current public key.
9. Click **Submit**.

Enable Autorotation of Key (Asymmetric Key, Sign, and Verify)

To enable autorotation of an asymmetric key used to sign and verify data:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. From the **Keys** page, choose **Alias** or **Keys** view.
3. Find the name of the key for which to enable autorotation and select **Rotate Key** from the **Actions** drop-down menu. The **Rotate AWS Key** dialog box is displayed.
4. Select the **Auto Rotate** tab.
5. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.

Note: If the toggle for **Enable autorotation on key expiration?** is displayed, skip it. This option displays only if the key has an expiration date associated with it. This option is not applicable to enabling autorotation of the key.

6. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
7. Deselect the **Disable sign permission on current key** check box, if you want to continue using the key that has been rotated out to sign and verify new data. By default, this check box is selected. Best practice is not to use the old key that has been rotated out to sign and verify new data.
8. Click **Submit**.

Synchronize Keys

The first login into the CCKM AWS automatically synchronizes keys from all AWS regions. If you make a key status change in AWS, you must synchronize to see the latest status in the CCKM GUI. If you schedule a key deletion, then after the designated time period has passed, CCKM will delete the key from AWS. You are then required to synchronize your keys from CCKM to remove this key permanently from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as your key source).

After selecting to synchronize your AWS keys within the selected region(s), the metadata (new or revised) of these keys are downloaded to CCKM.

You have the option to schedule key synchronization. Perform this scheduling in the **Schedules** page.

To synchronize your keys:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. Click **Synchronize**. The **Synchronize Key** dialog box displays.
3. Ensure the regions for which you wish to synchronize keys are included in the **Synchronize keys for regions** box. Remove any region whose keys you wish to **not** synchronize at this time by selecting the "X" next to the region name.
4. Click **Synchronize** to proceed with the synchronization.

Enable Autorotation before Key Expiration

To enable autorotation of a key before it expires, you must first set up a key-expiration schedule in the **Schedules** page. Once the key is enabled for autorotation, the key will be rotated before it expires. After the key is autorotated, the key will have a new expiration date based on the key-expiration schedule.

To enable autorotation of a key before it expires:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. Choose **Alias** or **Keys** view.
3. Find the name of the key for which to enable autorotation before key expiration and select **Rotate Key** from the Actions drop-down menu. The **Rotate AWS Key** dialog box is displayed.
4. Select the **Auto Rotate** tab.
5. From **Enable autorotation on key expiration?**, select **On** to enable the autorotation of this key before it expires. By default, this toggle is initially set to **Off**.

Note: Skip the **Enable autorotation on this key?** as it is not applicable to enabling autorotation of this key before it expires.

6. From the **Schedule** drop-down menu, select a key-expiration schedule to which to apply to this key. Any key-expiration schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
7. From the **Key Provider** box, select **CipherTrust Manager**, **DSM**, or **nShield** depending on the key source you are using for the key.
8. Deselect the **Disable Encrypt permission on Current key** check box, if you want to continue using the key that has been rotated out to encrypt new data. By default, this check box is selected. Best practice is not to use the old key that has been rotated out to encrypt new data. Strictly maintain the rotated-out key so that KMS can continue using it to decrypt the previously encrypted data by that key.
9. Click **Submit**.

Export Keys

Export Keys allows you to export the contents of the **Keys** list into a report in a CSV format.

1. In the left navigation bar, click **Keys**.
2. Click **Export Keys**.

This will download all of the keys in a report in a CSV format.

Reports

It is necessary to set up the CloudTrail and CloudWatch services in AWS before AWS keys will be represented in the CCKM reports. You also have the option to set these services up in CCKM using the **General** tab in the **Settings** page. For more information, see ["Configure CloudTrail and CloudWatch in AWS to Use CCKM Reports" below](#).

Choose Reports in the left-hand navigation to access the following reports:

- **Combined Key Activity Reconciliation Report**

Click **Run Report** to generate a new list of events connected to a key.

- **Key Activity Report**

Illustrates the activity type performed by a specific user and at the stated time.

- **Key Service Usage Report**

Indicates the:

- Key name
- Origin: AWS KMS or External
- Requesting Service: Name or ID of the service using a key
- Region
- Request Timestamp: Time when the application used the key.

- **Key Aging Report**

Provides information about key expiration using the following columns:

- Key Name
- Region
- Scheduled key deletion date
- Key expiration: Date and time at which this key expires.

- **Cloud Key Manager User Action Report**

Lists the users and their actions in creating and deleting keys in the CCKM portal.

Configure CloudTrail and CloudWatch in AWS to Use CCKM Reports

CCKM uses AWS CloudTrail and CloudWatch services to generate reports.

Overview

AWS CloudTrail:

- Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail.
- Actions performed on or by AWS KMS keys are also recorded in CloudTrail.
- CCKM uses these audit records to generate 'Key reconciliation report', 'Key service usage report' and 'Key activity report'.

AWS CloudWatch:

- CloudWatch allows the querying and filtering of CloudTrail logs through log groups.
- CCKM uses CloudWatch log group to fetch KMS audit logs and generate reports.

Configure CloudTrail

1. Log in the AWS Management Console.
2. **Important: Select any region. Note down this region.**
This region will be provided in the CCKM in the last steps.
3. Go to the **CloudTrail** section in **Management Tools**, select **Trail**, and click **Create Trail**. Create trail in CloudTrail with following settings:
 - a. Apply trail to all regions
 - b. 'Read/Write events' in Management events
 - c. Skip data events section.
 - d. Select any Storage location as per your convenience for logs collected by CloudTrail.

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-create-and-update-a-trail.html>

Configure CloudWatch Log Group

Monitor the new trail using a CloudWatch log group.

1. Configure CloudWatch Logs to receive your logs from CloudTrail so that you can monitor for specific log events.
2. Edit trail created in the previous section.
3. Enter a new or existing CloudWatch log group name. **Note down this log group name.**

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/send-cloudtrail-events-to-cloudwatch-logs.html>

Schedules

The **Schedules** page allows Cloud administrators to view and schedule the following job types:

- Key Rotation
- Key Synchronization
- Key Expiration

CCKM provides support for multiple schedules for each of the job types.

At the top right of the **Schedules** page, the **Add Schedule** button is available from which to add a scheduled job type.

Schedule Key Rotation

CCKM provides support for multiple schedules of a rotation of an AWS key. A new version of the key is added to the AWS key during a scheduled key rotation. Scheduled rotation requires that this feature be configured in the **Schedules** page prior to enabling the feature in the **Auto Rotate Key** dialog box (from the **Actions** column of the **All Keys** list). Your AWS key credentials (AWS Access Key ID and AWS Secret Access Key) are required as part of the configuration and are saved to the key source you are using (CipherTrust Manager or DSM) or to MongoDB (if you are using nShield Connect as your key source). If you have both DSM and nShield Connect configured as your key sources, then your AWS key credentials are saved only to DSM where it is stored.

Note

CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

During a scheduled rotation of a key, CCKM uses your associated AWS key credentials to rotate the key. If the AWS Secret Access Key expired or changed, then a new Secret Access Key must be reentered. Otherwise, the scheduled rotation will fail.

You can pause or delete a scheduled key rotation in the **Schedules** page. Note that when a key rotation schedule is deleted from CCKM, then the AWS key credentials are also deleted from the key source you are using (CipherTrust Manager or DSM) or from MongoDB (if you are using nShield Connect as your key source).

Set up a Key Rotation Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_ROTATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Enter your AWS key credentials in the **AWS Access Key ID** and **AWS Secret Access Key** boxes or click **Choose file** from **Upload Credentials** to upload an AWS credentials file that contains both the AWS Access Key ID and AWS Secret Access Key.
9. Click **Save**.

Pause a Key Rotation Schedule

To pause a key rotation schedule, select **Yes** for **Pause** within the **Add Schedule** dialog box and then click **Save**. The setting is saved. To resume the rotation schedule after pausing it, select **No** for **Pause** and then click **Save**. The setting is saved.

Delete a Key Rotation Schedule

To delete a key rotation schedule, select **Delete** from the **Actions** drop-down menu from the **Schedules** page. Note that the AWS key credentials (AWS Access Key ID and AWS Secret Access Key) associated with the given key rotation schedule are also deleted from the key source you are using (CipherTrust Manager or DSM) or from MongoDB (if you are using nShield Connect as your key source).

Schedule Key Synchronization

CCKM allows for multiple schedules of the synchronization of AWS keys. Synchronizing downloads any keys that were created in AWS, outside of the CCKM portal, into your CCKM. Scheduled key synchronization requires that this feature be configured in the **Schedules** page. You can also pause or delete a scheduled key synchronization in

the **Schedules** page. Note that when a key synchronization schedule is deleted from CCKM, then the AWS key credentials (AWS Access Key ID and AWS Secret Access Key) associated with the given key scheduled synchronization are also deleted from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as the key source). If you have both DSM and nShield Connect configured as your key sources, then the AWS key credentials are deleted from DSM where it is stored.

Note

CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Set up a Key Synchronization Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_SYNCHRONIZATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Enter your AWS key credentials in the **AWS Access Key ID** and **AWS Secret Access Key** boxes or click **Choose file** from **Upload Credentials** to upload an AWS credentials file that contains both the AWS Access Key ID and AWS Secret Access Key.
9. Click **Save**.

Schedule Key Expiration

CCKM provides support for multiple schedules of the expiration of AWS keys in the **Schedules** page. You can also pause or delete a scheduled key expiration in the **Schedules** page. Note that when a key expiration schedule is deleted from CCKM, then the AWS key credentials (AWS Access Key ID and AWS Secret Access Key) are also deleted from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as your key source). If you have both DSM and nShield Connect configured as your key sources, then the AWS key credentials are deleted from DSM where it is stored.

Note

CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Set up a Key Expiration Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_EXPIRATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. From **Start Date**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.
8. From **Expire a new key after**, enter a number and select the interval unit of Hour, Day, Week, Month, or Year from the drop-down menu after which the new key expires. The supported ranges are 6 through 24 hours, 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.
9. Enter your AWS key credentials in the **AWS Access Key ID** and **AWS Secret Access Key** boxes or click **Choose file** from **Upload Credentials** to upload an AWS credentials file that contains both the AWS Access Key ID and AWS Secret Access Key.
10. Click **Save**.

Logs

CCKM maintains logs for events, such as login, key generation, synchronizing of keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. For each logged event listed on the **Logs** page, the following information is displayed:

- Event name
- Severity level
- Date and time of the event
- Event message
- User (name) associated with the event

You can search for a specific logged event from the **Logs** page using the **Search** box. Allowable filters on the search are event name, severity level, event message, and user (name).

Key Sources

The **Key Sources** page displays **all** of the keys that were generated from the key source you are using. If you are logged into multiple clouds within CCKM, this page displays all of the keys from these clouds.

Three tabs are available at the top of the **Key Sources** page:

- **CipherTrust Manager Key:** Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.
- **DSM Key:** Use this tab to use DSM as your key source. The keys are generated and stored within DSM.
- **nShield Key:** Use this tab to use nShield Connect HSM as your key source. The keys are generated within nShield Connect and then stored as blobs within MongoDB.

Note

CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Keys listed on the **Key Sources** page are listed by name, cloud ID, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description and actions you can take regarding that key.

A **Search** box is available at the top left of the **Keys Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Cloud ID
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

Add (Generate) a Key

From the **Key Sources** page, you can add (or generate) keys in CCKM using CipherTrust Manager, DSM, or nShield Connect as a key source for *all* of the clouds into which you are currently logged.

Note

The key algorithm currently supported for an AWS key in CCKM is AES256.

Add (Generate) a CipherTrust Manager Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **CipherTrust Manager Key** tab.
3. Select **Add Key > AWS Key**.
4. In the **Add CipherTrust Manager Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **AWS** and (key) **Algorithm** is fixed to **AES256**.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.
(Reset clears the screen before it has been saved.)

Add (Generate) a DSM Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key** tab.
3. Select **Add Key > AWS Key**.
4. In the **Add DSM Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **AWS** and (key) **Algorithm** is fixed to **AES256**.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.
(Reset clears the screen before it has been saved.)

Add (Generate) an nShield Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **nShield Key** tab.
3. Select **Add Key > AWS Key**.
4. In the **Add nShield Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **AWS** and (key) **Algorithm** is fixed to **AES256**.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.
(Reset clears the screen before it has been saved.)

Delete a Key

Deleting a key from the **Key Sources** page deletes the key from CCKM. If you delete a key from **Key Sources** that has already been uploaded to AWS, the key will not be deleted from AWS.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key**, **CipherTrust Manager**, or **nShield Key** tab depending on the key source you used to generate this key.
3. Click **Delete** next to the target key.
4. The **Delete Key** dialog box with a Warning displays asking you to confirm the key deletion by entering the supplied phrase.
5. Enter or copy/paste the confirmation phrase.
6. Click **Delete**.

Manage keys in Key Sources List Page

In addition to adding and deleting keys, the **Key Sources** page allows you to sort the existing keys by Name, Key Type, Algorithm and Created Date.

Settings

To access the **Settings** page in CCKM, select **Settings** from the left-hand navigation bar. From the **Settings** page, you can:

- Enter the email addresses of the recipients who are to receive the email notifications.
- Enable email notifications to be sent out from CCKM when specific types of key actions are performed. For example, when a key is uploaded to the cloud service. You can also enable CCKM to send reminder notifications for when to rotate a key.
- Enable reporting for AWS keys using an existing AWS CloudTrail and CloudWatch log group or a new trail and log group that you create (from the **Settings** page).
- Configure a remote syslog or log server to which to send CCKM log data.
- View information about the software release number of the CCKM you are running including the build and version number.

Adding Recipients' Email Addresses (for Notifications)

Use the **General** tab from the **Settings** page to enter the email addresses of the recipients who are to receive email notifications from CCKM. Enter the recipients' email addresses separated by a comma in the **Notification Recipient** box and then click **Save** to save setting.

Enabling Notifications

Using the **Keys Notification** tab from the **Settings** page, you can configure CCKM to send an email notification when a key is deleted from AWS or restored or uploaded to AWS. You can also use this tab to configure CCKM to send a reminder notification for when to manually rotate a key along with the frequency (in days) at which a notification is sent. Add the email address(es) of the recipients in the **General** tab from the **Settings** page. By default, no notifications are sent.

To enable email notifications:

1. In the **Settings** page, select the **Keys Notification** tab.
2. Under **Notifications**, select **On** for each of the notifications you wish to enable:
 - Rotate AWS Key (reminder notification)
 - Delete AWS Key
 - Import AWS Key
 - Upload AWS Key
3. If you selected to enable **Rotate AWS Key**, in the **Days** box, enter the number of days for the frequency a reminder notification is sent.
4. Click **Save**.

Enabling Reporting for AWS Keys

Use the **General** tab within the **Settings** page to enable CCKM reporting for AWS keys using an existing AWS CloudTrail applicable to an AWS region and a CloudWatch log group to which the trail belongs. You can also use the **General** settings tab to create a new CloudTrail and CloudWatch log group to which the trail belongs. You first select an AWS region in which to create it and then add the name of a new CloudWatch log group. Thereafter, you configure the other required configurations.

Enabling Reporting for AWS Keys Using an Existing CloudTrail and CloudWatch Log Group

1. In the **Settings** page, select the **General** tab. The **General** page displays.
2. From the **Select CloudTrail Region** drop-down menu, select the AWS region to which the trail applies.
3. From the **Select CloudWatch Log Group Name** drop-down menu, select the name of the log group to which trail belongs.
4. Click **Save**.

Enabling Reporting for AWS Keys Using a New CloudTrail and CloudWatch Log Group

1. In the **Settings** page, select the **General** tab. The **General** page displays.
2. From the **Select CloudTrail Region** drop-down menu, select the AWS region in which to create the new trail.
3. From the **Select CloudWatch Log Group Name** drop-down menu, enter the name of the new CloudWatch Log Group to create. This log group receives your logs from CloudTrail so that you can monitor for specific log events. Entering a name of the new log group triggers the display of other configuration boxes associated with creating a new trail and log group.
4. From the **Stack Name** box, enter the name of the new stack to create. The AWS CloudFormation stack automates the creation of all resources required for creating a new trail (bucket, trail, and log group). If a resource fails, the stack rolls back to the previous version. From AWS CloudFormation, you can view information about any failure during the process of the creating the stack using the name of the stack. You can also delete the stack from CloudFormation without any impact to your AWS resources.
5. From the **S3 Bucket Name** box, enter the name of an S3 bucket to designate as the storage of the trail log files. The name must be globally unique. If you do not enter a bucket name, the default bucket name of **cckm-bucket** is used.
6. From the **Cloud Trail Name** box, enter the name of the new trail. If you do not enter a trail name, the default trail name of **cckm-bucket** is used.
7. From the **CCKM IAM Role Name** box, enter the name of the AWS IAM Role used for CCKM. If you enter an existing IAM role name, then CCKM uses this name to apply to the configuration. If you enter a new IAM Role name, CCKM creates this role with the required permissions to create the resources for the trail. If you do not enter an IAM role name for CCKM, the default role name of **AWS-*<region-name>*-cckm-cloudtrail-to-cloudwatch** is used.

Note: If you do not have the permissions to create a new IAM role, contact your AWS administrator to create the IAM role with permissions to create and delete log groups, trails and buckets.

8. Enter your AWS key credentials in the **AWS Access Key ID** and **AWS Secret Access Key** boxes or click **Choose file** from **Upload Credentials** to upload an AWS credentials file that contains both the AWS Access Key ID and AWS Secret Access Key.
9. Click **Save**.

Configuring Syslog Server

Note

In this document, the terms "syslog server" and "log server" are synonymous.

The **Syslog** tab from the **Settings** page allows you to configure the remote syslog or log server to which to send CCKM log data using TCP, TLS, or UDP protocol as the transport protocol. If you use the TLS protocol, then you are required to upload the syslog server certificate to CCKM for the TLS connection.

The **Syslog** tab also allows you to delete your syslog server configuration so that log data are no longer sent to the server.

Prior to configuring the setting for a remote syslog server, ensure that you have installed and configured the syslog server including configuring the setting for the port number the transport protocol will use to connect to CCKM. This port number configuration must match the port configuration you set in the **Syslog** tab of the **Settings** page.

Note

Support for both Syslog and Splunk as a remote syslog server is provided in CCKM. However, the use of a Splunk cloud server as a remote syslog server is currently not supported in CCKM.

Configure Remote Syslog Server

To configure a remote syslog server to which to send CCKM log data:

Note

The facility name of "LOCAL1" for the syslog server displays in the **Facility** box. This default value is not available for modification.

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. In the **Hostname** box, enter the hostname or IP address of the syslog server.
3. In the **Port** box, enter the port number the transport protocol uses to connect to the syslog server. Enter a value between 1 and 65535. The default value is 514.
4. In the **Protocol** drop-down menu, select **TCP**, **TLS**, or **UDP** for the transport protocol to use when sending the log data to the syslog server. The default value is **UDP**. If you select **TCP**, proceed to Step 6. If you select **TLS**, proceed to the next step. Otherwise, proceed to Step 7.
5. In the **Certificate** box, select **Choose File** and then select the syslog server certificate to upload to CCKM.
6. Click **Test Connection** to test the connection between CCKM and the remote syslog server. This button is only available for use when either **TCP** or **TLS** is the selected transport protocol.
7. In the **Log Format** drop-down menu, select **CEF** (Common Event Format), **Plain Text**, or **RFC5424** for the format of the log message to use when sending the log data. The default value is **RFC5424**.
8. (Optional) In the **Description** box, enter a description of the remote syslog server.
9. Click **Save**.

Delete Remote Syslog Server Configuration

To delete the configuration of a remote syslog server:

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. Click **Delete**. The **Delete Syslog** dialog box displays.
A warning message displays indicating that this action will delete the syslog settings from CCKM. The message also indicates to type the confirmation phrase (I wish to delete the syslog settings) in the text box if you still wish to delete the syslog settings.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Delete**.

About

Use the **About** tab to view the release number of the CCKM you are running.

Chapter 14: CCKM for Azure Key Management

Overview	117
Accessing and Managing Azure Resources	117
Registering Apps in Azure	118
Prerequisites on Azure	118
Access CCKM for Azure	118
Azure CCKM User Portal	120
Home	122
Azure Directory Context Switcher (Support for Guest User for AAD B2B Collaboration)	122
Keys	124
Reports	136
Schedules	137
Logs	140
Key Sources	141
Settings	143

Overview

CCKM provides support for Azure clouds (Azure, and Azure China). Note that in this chapter, the information regarding Azure is applicable to all Azure clouds unless otherwise noted. The CCKM UI for Azure, and Azure China is identical with the exception of the display of the names of the clouds.

Accessing and Managing Azure Resources

Note

This section is only applicable to Azure and Azure China.

Azure allows external applications, such as CCKM, to access and manage resources within Azure in the following ways:

- On behalf of a user
- As a service principal

Both ways require that the application be first registered in Azure. For applications on behalf of a user, it is required that the Azure users be created and given the appropriate access controls and policies on the resources. The application obtains an access token through app+user authenticate. What an application can do with the Azure resources depends on what the user can do.

For as-service-principal applications, the application must be given the appropriate access controls and policies on the resources. The application obtains an access token through an *app-only authentication*. What an application can do with Azure resources depends on whether the service principal has proper permissions on the resources.

Registering Apps in Azure

When registering apps in Azure, the following are the two types of app credentials employed:

- Secret (password)—The user will generate a secret key in Azure when registering the app, and then copy the secret key and provide it to the app. When making OAuth authentication calls, the app will send the secret key to Azure.
- Certificate (public key)—the user will create a private key and public key pair locally, create a certificate for the public key, and then provide the certificate to Azure when registering the app. For the private key, the app will create a client assertion and send it to Azure when making OAuth authentication calls.

CCKM currently supports both types of app credentials for Azure and Azure China.

Prerequisites on Azure

The Azure administrator must ensure the following prerequisites are met:

- Each Azure key vault must belong to an Azure Resource Group.
- Key vault owner must give designated users or the CCKM app (if CCKM is used as a service principal) the “Key Vault Contributor” role in Azure Access Control, and appropriate permissions as needed on keys in Access Policies.

Note: If CCKM is used as a service principal within a key vault, you have the option to use the Azure permission model of **Azure role-based access control** for the service principal under **Key vaults > Access policies**. Ensure to also give the service principal the role of “Key Vault Administrator” within the applicable key vault. For more information about Azure role-based access control, refer to the Azure online documentation.

Note: Changing key vault Access Control and/or Access Policies will not affect the active CCKM user sessions. To reflect the change, a user must re-log in to CCKM to establish a new user session.

- CCKM Reports draw on the Azure Log Analytics service, which must be pre-configured if the CCKM Reports functionality is desired (see “[Reports](#)” on page 136). At minimum, you must have:
 - An Azure *Log Analytics* service created and associated with the relevant key vaults.
 - In the *Log Analytics > {Log service name} > Access Control (IAM)* section of the Azure portal, set “**Log Analytics Contributor**” permissions for any user who should be able to generate reports in CCKM. Other users can view, but not create reports.

For more information about how to configure Log Analytics in Azure key vaults, see the latest version of the following Microsoft Azure documentation:

<https://docs.microsoft.com/en-us/azure/key-vault/general/howto-logging?tabs=azure-portal>

Access CCKM for Azure

This section describes the two ways in which to access the CCKM service for Azure Key Management:

- “[Accessing CCKM for Azure Using User Login](#)” on the facing page
- “[Accessing CCKM for Azure Using Service Principal](#)” on the facing page

Accessing CCKM for Azure Using User Login

For details about accessing CCKM as a CCKM user for the CCKM service for Azure Key Management, see [Chapter 3: "About CCKM Admin and User Portals" on page 27](#). This chapter includes information on the ways in which to access the CCKM service for Key Management.

Accessing CCKM for Azure Using Service Principal

Note

This section only applies to Azure and Azure China using CCKM as a service principal.

To access CCKM for Azure using CCKM as a service principal, the first time you log into CCKM as an administrator user, you are required to provide the administrator consent to grant permissions to the CCKM app to access your Azure resources (your organization data). *After* you click on the "Admin Consent" button, you are redirected to the Azure consent page where you can provide your consent (after reviewing the page) by clicking **Accept**.

As an administrator user, you can confirm that CCKM has been granted the permissions as a service principal *after* you have granted your administrator consent by logging into the Azure portal and verifying that the CCKM app is listed in the **Enterprise Applications > All Applications** page.

After you have granted these permissions, you can log into CCKM for Azure using CCKM as a service principal using your tenant credentials.

For more information, see the following sections:

- ["Granting Admin Consent First Time Accessing CCKM for Azure Using Service Principal" below](#)
- ["Accessing CCKM for Azure Using Service Principal \(After Granting Admin Consent\)" on the next page](#)

Granting Admin Consent First Time Accessing CCKM for Azure Using Service Principal

To access the CCKM service for Azure Key Management using CCKM as a service principal for the first time and grant your Admin consent:

1. Log into the CCKM Admin Portal as a CCKM administrator using your administrator credentials. The CCKM Admin Portal displays.
2. From the **Key Management** drop-down list at the top of the CCKM Admin Portal page, select **Azure | Azure China**. You are redirected to the CCKM login dialog box for Azure to use CCKM as a service principal. At the top of this dialog box, the following message displays:
"If this is your first time to use CCKM for managing your Azure keys, please click [here](#) to provide Admin Consent."
3. Select the "admin consent" link (indicated by "here" in blue text within the displayed message). The CCKM **Admin Consent: Azure** dialog box displays.
4. In the **Tenant Name** box, enter the tenant name. To locate your tenant name in the Azure portal (after logging in as an administrator user), go to **Azure Active Directory > properties > Tenant Name**.
5. In the **Tenant ID** box, enter the tenant ID. To locate your tenant ID in the Azure portal (after logging in as an administrator user), go to **Azure Active Directory > properties > Tenant ID**.
6. In the **Tenant Password** box, enter the tenant password. The other users will use this password along with their tenant name to login to the CCKM User Portal. The password must be at least 8 characters long and contain one upper case letter, one lower case letter, one digit, and one special character from a limited set of special characters "!@#%&^*()".

7. In the **Confirm Tenant Password** box, reenter the tenant password.
8. Select **Admin Consent** to provide your consent to Microsoft to grant permissions to the CCKM app. You are redirected to the Microsoft login page to select an account to use to login to your Azure AD. This account must be a user with "Application administrator" role in your Azure AD.
9. Enter your Azure Admin username and password. The Microsoft consent prompt displays.
10. Review the information and then click **Accept** to accept the request to grant permissions to the CCKM app to access your Azure resources as a service principal. You are then redirected to a CCKM page displaying a message indicating the given Azure tenant admin granted permissions to this CCKM application and that you can use your tenant name and password to login to CCKM.
11. Select the "**login to CCKM**" link. You are redirected to the CCKM user login page.
12. Select the Azure key management service you wish to use. You are redirected to the CCKM login dialog box for Azure to use CCKM as a service principal.
13. In the **Tenant Name** box, enter the tenant name.
14. In the **Tenant Password** box, enter the tenant password.
15. Click **Login**. The Azure CCKM dashboard displays.

Accessing CCKM for Azure Using Service Principal (After Granting Admin Consent)

To access the CCKM service for Azure Key Management using CCKM as a service principal *after* you have granted your Admin consent:

1. From the CCKM user login page, select the Azure key management service you wish to use. You are redirected to the CCKM login dialog box for Azure to use CCKM as a service principal.
2. In the **Tenant Name** box, enter the tenant name.
3. In the **Tenant Password** box, enter the tenant password.
4. Click **Login**. The Azure CCKM dashboard displays.

Azure CCKM User Portal

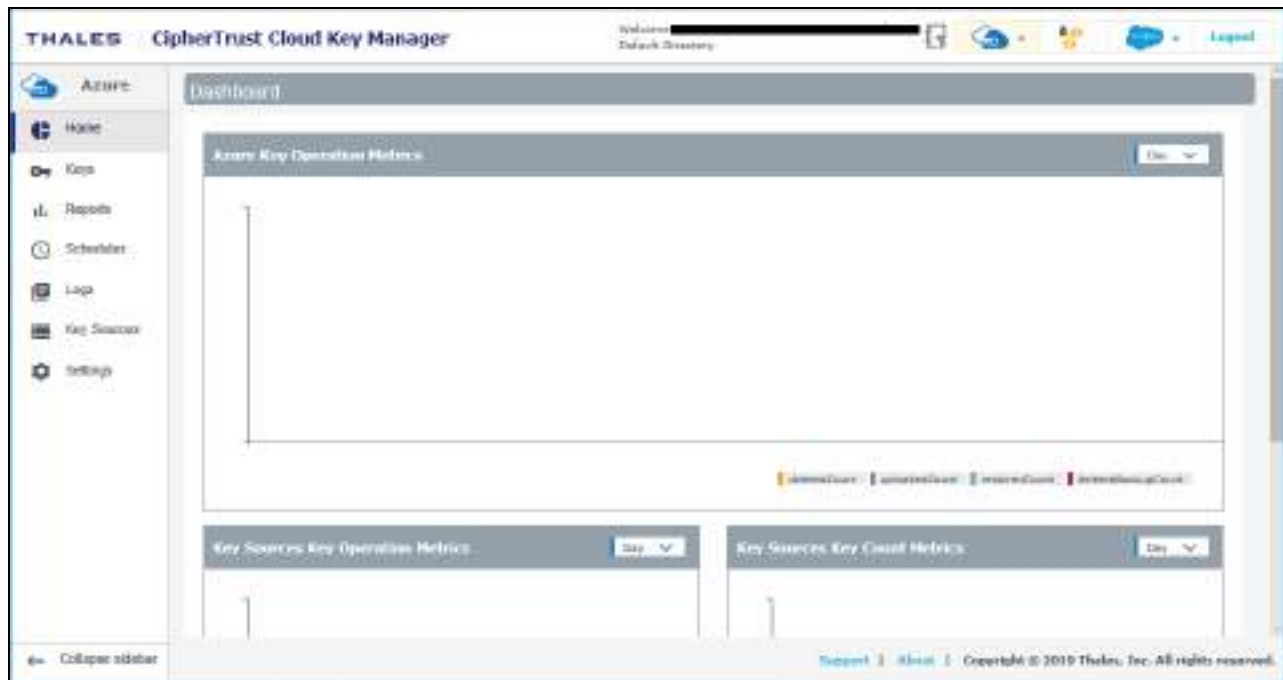
The CCKM User Portal for Azure contains the following menu options in the left-hand navigation bar:

- **Home:** Home page displays key operation and count metrics.
- **Keys:** Keys page displays key management overview and provides the Upload and Synchronize functionality, and the Delete, Restore, and Auto Rotate key operations available in CCKM.
- **Reports:** All Reports overview page links to detailed individual reports.
- **Schedule:** In Schedule, you can schedule automatic key rotation, key rotation based on key expiration, and key synchronization.
- **Logs:** Logs page lists individual logs that can be searched, sorted, and viewed.

- **Key Sources:** In **Key Sources**, you can generate keys in CCKM to upload to Azure. CCKM provides support for the use of CipherTrust Manager and DSM as key sources to generate, store, and retrieve the encryption keys that CCKM uses and manages. CCKM also provides support for the use of nShield Connect HSM as a key source to generate the encryption keys that CCKM uses and manages. These keys are then stored in MongoDB. Also supported in **Key Sources** for all Azure clouds are BYOK and PFX file upload.

Note: CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

- **Settings:** The **Settings** page allows you to add the email addresses of the recipients who are to receive email notifications and configure the settings for email notifications, remote syslog server, and proxy server. You can also view the release number of the CCKM running from this page.



To display the context of which Key Management service you are currently logged into within CCKM, the icon associated with the Key Management service displays at the top of the left-hand navigation bar.

The arrow button at the bottom of the navigation bar allows you to expand the bar to view the names of the menu options and collapse it to view only the icons associated with the menu options. Click the arrow button to either expand or collapse the navigation bar.

On the top-right side of the CCKM portal for Azure, the following options are available:

- **Azure Cloud Service Context Switcher:** This context switcher allows you to login into another Azure cloud service different from the current Azure cloud service into which you are currently logged. If you select the Azure cloud in which you are currently logged, the Multi-account Login/Logout window for the Azure account(s) displays. For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **AWS Login:** If you are not already logged into AWS, clicking on the AWS Login icon brings you to the AWS Password Login page where you can enter your AWS credentials to access the CCKM AWS cloud service (from the current cloud service in which you are currently logged). If you are already logged into AWS, clicking on the AWS Login icon brings you to the Multi-account Login/Logout window for the AWS account(s). For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).

- **Google Cloud Service Context Switcher:** This context switcher allows you to login into another Google Cloud organization different from the current Google Cloud organization into which you are currently logged. If you select the Google Cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Google account(s) displays. For more information about the **Multi-account Login/Logout** window, see ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **Salesforce Cloud Service Context Switcher:** This context switcher allows you to login into another Salesforce cloud service different from the current Salesforce cloud service into which you are currently logged. This is Salesforce, Salesforce Government, or Salesforce Sandbox. If you select the Salesforce cloud in which you are currently logged, the Multi-account Login/Logout window for the Salesforce account(s) displays. For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **Logout:** Logout from CCKM.

Home

The **Home** page provides a dashboard to display key operation and count metrics.

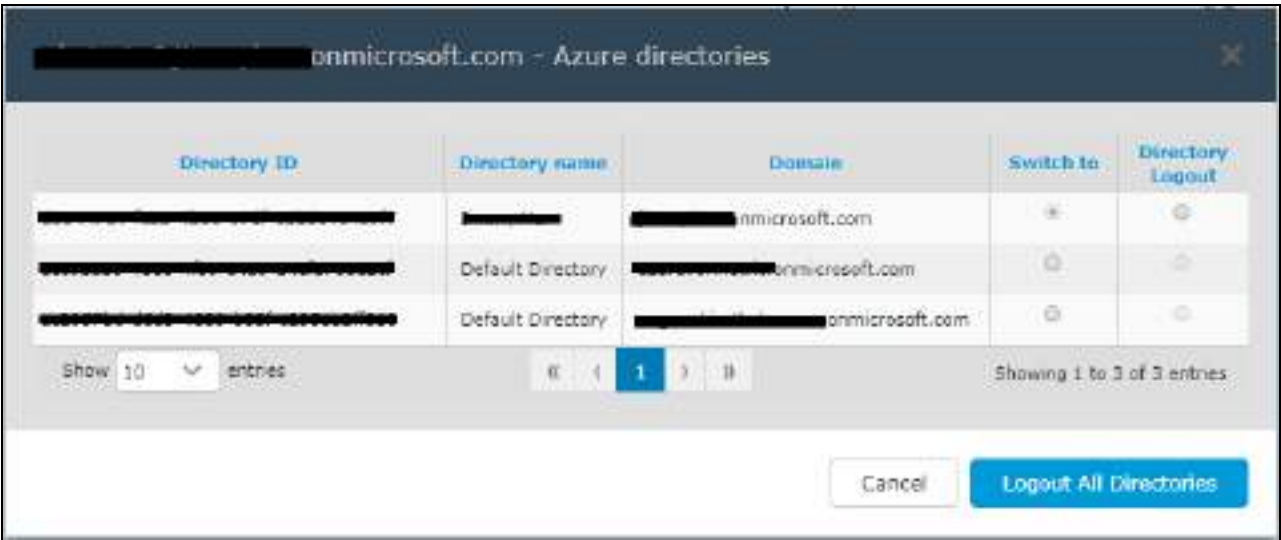
Azure Directory Context Switcher (Support for Guest User for AAD B2B Collaboration)

CCKM supports the ability to allow Azure guest users to take part in B2B collaboration for Azure tenants to share user authentication and applications with partners. Prior to using this feature in CCKM, the guest user account must already be set up in the Azure Active Directory as a guest user with the appropriate access policies applied and the account activated in Azure. Once you are logged into CCKM Azure, from the **CCKM Directory Context Switcher** window, you have the ability to view all of the Azure directories to which you belong, including your default directory and your business partner's directory to which you are set up as a guest user. Using the **CCKM Directory Context Switcher** window, you can switch to any of the other directories to which you belong by logging into the directory from the same window. Once logged into a given directory, you can access and manage the resources and keys that reside in the given directory.

Note

The **Azure Directory Context Switcher** is not applicable to a CCKM that is used as a service principal.

The following **CCKM Directory Context Switcher** window displays the Azure directories in which a given user belongs:



You can perform the following actions using the CCKM Directory Context Switcher window:

- Logon to multiple directories for the Azure cloud
- Logout of a specific directory or all directories
- Perform key management for multiple directories from your current CCKM browser window

Accessing the CCKM Directory Context Switcher Window

To access the **CCKM Directory Context Switcher** window after you are logged into CCKM Azure, click on the **CCKM Directory Context Switcher** icon (🔑) at the top of right of the page. The CCKM Directory Context Switcher window displays. The following sections provide instructions on how to use this window.

Logging Out of User Directory

To logout of a specific directory, select the **Directory Logout** button next to the domain name associated with the directory for which to log out. The **Log out CipherTrust Cloud Key Manager** window displays. Click **Continue with Log Out** to log out. You are logged out of the directory and then the CCKM login page is displayed.

Switching to Another User Directory

To access another directory (you have not already logged into), select the **Switch to** button next to the domain name associated with the directory for which to log on. The Microsoft login page displays. Enter your user credentials for the given directory in the login dialog boxes. Once you successfully log into Microsoft, the Azure CCKM portal is displayed.

After you have logged into a directory, you are not prompted to log in again to switch to that directory.

Logging Out of All User Directories

To logout of all directories for Azure, click **Logout of All Directories**. The **Log out CipherTrust Cloud Key Manager** window displays. Click **Continue with Log Out** to log out. You are logged out of all the directories and then the CCKM login screen is displayed.

Keys

The **Keys** page allows Cloud administrators to view all the keys and manage them within the boundary of granted permissions. Those keys or key vaults that do not have List permission for the respective Cloud administrator will not be listed.

A **Search** box is available at the top left of the **Keys** page allowing for a search of a specific key by entering any one of the following parameters:

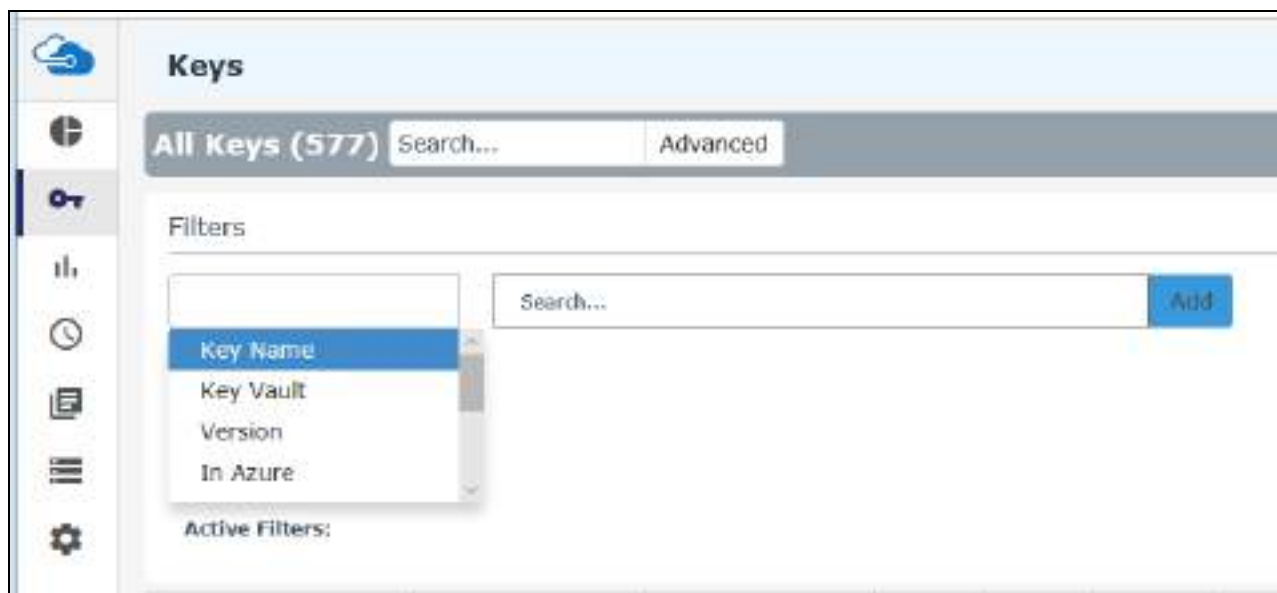
- Key ID
- Version Key ID
- Keyvault name
- Keyvault location
- Source key
- Subscription ID

The **Advanced** tab allows you to filter your search on any or a combination of these columns displayed on the **Keys** page with the exception of the **Count** and **Action** columns:

- Key Name
- Key Vault
- Version
- In Azure—Available options from the drop-down menu are In Azure, Soft Deleted, and Deleted.
- Backup—Available options from the drop-down menu are Backed Up and Not Backed Up.
- Enabled—Available options from the drop-down menu are Enabled and Disabled.
- Location
- Key Material Origin
- Auto Rotate—Available options from the drop-down menu are ON and OFF.

Each of these columns are described in more detail below in this section.

To select to filter based on a column or combination of these columns from the **Keys** page, click **Advanced**. The **Filters** section displays. From the column filter drop-down menu, select the name of the column on which to filter your search. For example, select **Key Name**. In the **Search** box, enter the value on which to search (for example, **MK-createkey** as the key name) and then click **Add**. For the **In Azure**, **Backup**, **Enabled** and **Auto Rotate** columns, a drop-down menu is available from which to select a value for the search. Click the down arrow, select a value from the available options, and then click **Add**. The specified search filter is then added to **Active Filters**. In the example, **Key Name: MK-createkey** is added to **Active Filters**. The results of the search displays in the **Keys** table. You can narrow your search by selecting more columns to add to the filter. To clear the filtered results, from **Active Filters**, click on the “X” (or the **Delete** button) for each of the named filters.



At the top right of the Keys page, the following buttons are available:

- **Export Keys:** Allows you to export the contents of the Keys list into a report in a CSV format.
- **New Key:** Allows you to upload an external BYOK key (CipherTrust Manager, DSM, nShield Connect, HSM .byok or PFX key) or create a native key. For information on how to upload an external BYOK key or create a native key, see ["Create a Key" on the next page](#).
- **Synchronize:** Allows you to download all the keys that were created in Azure (to which you have access) into CCKM. For information on how to synchronize your keys, see ["Synchronize Keys" on page 134](#).

The following columns display on the **Keys** page:

- **Key Name:** The name of the key.
- **Key Vault:** The name of the Azure Key Vault. If the key vault is a premium key vault, then (PREMIUM) will be appended to the key vault name.
- **Version:** The version number of the key.
- **Count:** The version count.
- **In Azure:** Indicates the status of the key in Azure, which can be one of the following:
 - A green check icon indicates the key exists in Azure.
 - An orange exclamation icon indicates the key vault or key was deleted from Azure with the Soft Delete feature enabled.
 - A red exclamation icon indicates the key vault or key was deleted from Azure without the Soft Delete feature enabled.
- **Backup:** Indicates whether a backup of the key exists in CCKM. The status can be one of the following:
 - A green check icon indicates the backup exists in CCKM.
 - A red exclamation icon indicates the backup key does not exist in CCKM.
- **Enabled:** Indicates whether the key is enabled for use.
- **Location:** Indicates the Azure region in which the key is stored. For example, West US or Central West US.

- **Key Material Origin:**
 - If the key was uploaded from CCKM, then the value is Internal(<source_key_name>).
 - If the key was not uploaded from CCKM as a BYOK key, then the value is EXTERNAL.
- **Auto-Rotate:** Indicates whether the key is set to be automatically rotated.
- **Actions:** Allows you to select any of the following actions on a given key:
 - Auto rotate
 - Delete
 - Add Version
 - Recover
 - Purge
 - Delete Backup
 - Restore

Create a Key

You have two options to create a new key:

- ["Upload your Own Key \(BYOK\)" below](#)
- ["Create a Native Key" on page 128](#)

Upload your Own Key (BYOK)

Upload the RSA keys created from the **Key Sources** page of CCKM to the Azure Key Vault. These are your "BYOK" keys.

As part of the upload process, Azure generates a corresponding backup key, which CCKM downloads and stores in the key source you are using (CipherTrust Manager or DSM) or in MongoDB (if you are using nShield Connect as your key source). For more information about the backup key, see ["Backup Key" on page 135](#).

Note

You must have the import permissions for the Azure Key Vault to which you wish to upload the key.

To upload your BYOK key to CCKM:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New Key** button and then **Upload**. The **Upload a Key** dialog box displays.

3. From **Source Key Tier**, ensure the key source you wish to use is selected. The key sources that display within **Source Key Tier** depends on the key source you are using. The following are the options:
 - **DSM**—Generated a key from the DSM and imported it into a Standard or Premium Azure Key Vault.
 - **CipherTrust Manager**—Generated a key from the CipherTrust Manager and imported it into a Standard or Premium Azure Key Vault.
 - **HSM.byok**—Generated a .byok key from an external Thales HSM and imported it into a Premium Azure Key Vault from CCKM. Once imported from CCKM, this key is stored in the DSM securely as a blob.
 - **PFX**—Imported an external password-protected PFX key into a Standard or Premium Azure Key Vault. Once imported from CCKM, this PFX key will be stored in the DSM securely as a blob.
 - **nShield**—Generated a key from the nShield Connect and imported it into a Standard or Premium Azure Key Vault.
4. From **Source Key** box, select the source key. Enter the first three characters of the name of the key to bring it up for selection.
5. From **Key Vault Tier** box, select the key vault tier of **Standard** or **Premium** depending on the subscription service type you have purchased.
6. From the **Key Vault Region** box, select the region where your key vault resides and whether you want to use the Azure Standard or Premium subscription service. Based on the Standard or Premium Key Vault Tier selection, the regions are filtered accordingly displaying only the regions, which are either Standard or Premium.
7. From the **Key Vault** box, select the name of the Azure Key Vault to where you intend to upload the key.
8. Add the name of the key in the **Key Name** box. If you wish to use the key name you entered when you generated this key from the **Key Sources** page, select the **Use source key name** check box. The name will display in the **Key Name** box.
9. From the **Key Destination** box, select either **Software** or **HSM** (Premium subscription only) depending on which destination you want to use.
10. Click **Enabled**, if the key is enabled. Note that if the key is disabled, the key cannot be used for cryptographic operations.
11. Enter the activation date of the key in the **Activation Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
12. Enter the expiration date of the key in the **Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
13. From **Key Operations**, select the cryptographic operations the key will perform. The options are:
 - Encrypt
 - Decrypt
 - Sign
 - Verify
 - Wrap Key
 - UnwrapKey
14. (Optional) From **Tags**, enter the tag name and value of the key.
15. Click **Upload to Azure**.
The uploaded key is displayed on the **All Keys** list page.

Create a Native Key

Create a native key in an Azure key vault and select the key types and sizes natively available in Azure. There is no source key tier as the Azure key vault will be the source of the key material. After CCKM creates a native key, it displays as **EXTERNAL** under the **Key Material Origin** column within the **All Keys** list page.

As part of creating a native key, Azure generates a corresponding backup key, which CCKM downloads and stores in the key source you are using (CipherTrust Manager or DSM), or in MongoDB (if you are using nShield Connect as your key source).

To create a native key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New Key** button and then **Create**. The **Create Key** dialog box displays.
3. From the **Key Vault Tier** box, select the key vault tier of **Standard** or **Premium** depending on the subscription service type you have purchased.
4. From the **Key Vault Region** box, select the region where your key vault resides. Based on the Standard or Premium Key Vault Tier selection, the regions are filtered accordingly displaying only the regions, which are either Standard or Premium.
5. From the **Key Vault** box, select the name of the Azure key vault to where you intend to create the key.
6. Add the name of the key in the **Key Name** box.
7. From the **Key Destination** box, select either **Software** or **HSM** (Premium subscription only) depending on which destination you want to use.
8. From **Key Type**, select **RSA** or **EC** for your key type.
9. If you selected **RSA** for key type, then select **2048**, **3072**, or **4096** from **Key Size** for the key size to apply to the new key. If you selected **EC** for key type, then select **P-256**, **P-384**, **P-521**, or **SECP256K1** from **Elliptical Curve Name** for the name of the elliptical curve to apply to the new version of the key.
10. Click **Enabled**, if the key will be enabled. Note that if the key is disabled, the key cannot be used for cryptographic operations.
11. Enter the activation date of the key in the **Activation Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
12. Enter the expiration date of the key in the **Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
13. From **Key Operations**, select the cryptographic operations the key will perform. The options are:
 - Encrypt
 - Decrypt
 - Sign
 - Verify
 - Wrap Key
 - UnwrapKey
14. (Optional) From **Tags**, enter the tag name and value of the key. Up to 15 tags are supported.
15. Click **Create**.

The new key is displayed on the **All Keys** list page.

Delete a Key

From the **Keys** page, deleting a key deletes it from Azure Key Vault (but a backup key remains in the CCKM). After deletion, two new actions are displayed within the **Action** column: *Restore* and *Delete Backup*.

Note

This Delete action deletes ALL versions of the Azure key.

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. In the resulting **All Keys** list, find the name of the key and click **Delete** from the **Action** column. A warning message is displayed.
3. Enter or copy/paste the confirmation phrase.
4. Click **Delete**.

Restore a Key

In the event that someone deletes an IBM key directly in the Azure portal, it is possible to restore it from the CCKM as follows:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list page, select **Restore** from the **Action** column of a deleted key.
The key and all its versions and metadata are restored to Azure Key Vault. The Success message is displayed, and the key is listed at the top of the **All Keys** page.

Note

If anyone deletes the backup key from CCKM, it will **not** be possible to restore the key to IBM.

Delete a “Soft-deleted” Key

From the **Keys** page, soft deleting a key deletes it from Azure Key Vault removing it from displaying in the CCKM UI. However, the key still exists in the Azure Key Vault and CCKM. After a soft delete, two new actions are displayed within the Action column: *Recover* and *Purge*.

To delete a key in a soft-delete state:

1. In the left navigation pane of the CCKM portal, select **Keys**.
2. From the **All Keys** list page, find the name of the key and click **Soft Delete** from the **Action** column.
A warning displays indicating that this action will delete the key from your Azure key vault and that the operation may take some time to complete. The message also indicates to type the confirmation phrase (I wish to delete the key) in the text box if you wish to delete the key.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Delete**.
The key and all its versions and metadata are deleted from the Azure Key Vault. The Success message is displayed, and the key is no longer listed on the **All Keys** page.

Recover a “Soft-delete” Key (to Azure)

From the **Keys** page, recovering a key that is in a “soft-delete” state allows you to restore the key to the Azure Key Vault. After the key is recovered, the key displays in the Azure Key Vault and is listed within the **All Keys** page.

To recover a key in a soft-delete state:

1. In the left navigation pane of the CCKM portal, select **Keys**.
2. From the **All Keys** list page, find the name of the key and select **Recover** from the **Action** column.
A warning displays indicating that this action will recover your key from your Azure key vault and that the operation may take some time to complete.
3. Click **Recover**.
The key and all its versions and metadata are restored to Azure Key Vault. The Success message is displayed, and the key is listed at the top of the **All Keys** page.

Purge a “Soft-delete” Key

From the **Keys** page, purging a key that is in a “soft-delete” state allows you to delete it permanently from the Azure Key Vault. However, you still can restore this same key to the Azure Key Vault using the **Restore** action. For more information, see ["Restore a “Soft-delete” Key \(to Azure\)" on the facing page](#).

Note

Prior to purging a key that is in a “soft-delete” state, you must first grant the Purge access policy permission within the given key vault in the Azure portal to the CCKM user or application using CCKM (as a service principal) to perform the purge. To grant the Purge access policy permission within the Azure portal, go to key vault **Settings > Access Policies**. For **Permission model**, select **vault access policy**. Under **Current Access Policies**, select **Key Permissions**. Then for the user (under **User**) or the application (under **Application**) (or both) you wish to grant this permission, select **Purge** under **Privileged Key Operations**.

To purge a key in a soft-delete state:

1. In the left navigation pane of the CCKM portal, select **Keys**.
2. From the **All Keys** list page, find the name of the key and click **Purge** from the **Action** column.
A warning displays indicating that this action will purge the key from the Azure key vault. The message also indicates to type the confirmation phrase (I wish to purge the Azure key) in the text box if you wish to purge the key.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Purge**.
The key and all its versions and metadata are deleted from the Azure Key Vault. The Success message is displayed, and the key is no longer listed on the **All Keys** page.

Restore a “Soft-delete” Key (to Azure)

From the **Keys** page, restoring a key that was previously set to a “soft-delete” state and then purged allows you to restore the key from the key source you are using (CipherTrust Manager or DSM) or from MongoDB (if you are using nShield Connect as your key source) and upload it to the Azure Key Vault. With a successful restore of a “soft-delete” key, the key and all its versions and metadata are restored to Azure Key Vault.

To restore a key that was previously set to a “soft-delete” state and then purged:

1. In the left navigation pane of the CCKM portal, select **Keys**.
2. From the **All Keys** list page, find the name of the key and select **Restore Key** from the **Action** column.
A warning displays indicating that this action will restore your key to your Azure key vault and that the operation may take some time to complete.
3. Click **Restore**.
If successful, the Success message is displayed, and the key is listed at the top of the **All Keys** page.

Rotate a Key

CCKM allows for a manual or autorotation of a key. You manually rotate a key by adding a new version from the **Add a version** dialog box. Auto rotating a key is adding a new version of the key at a scheduled time. Key autorotation requires that you first schedule the autorotation in the **Schedules** page prior to enabling this feature in the **Auto-rotate Key** dialog box. Note that the key you select to autorotate must be enabled for it to be used for cryptographic operations after the rotation.

Note

Any key that is shared among different users within the same Azure cloud and is scheduled for a key rotation by one user cannot be scheduled for autorotation again by another user. Any key that is scheduled for autorotation displays the status of “AUTO” with an orange check icon next to it within the **Auto Rotate** column of the **Keys** page. If you attempt to schedule autorotation using **Rotate Key** within the **Keys** page with a key already scheduled, a message displays indicating that the key is already scheduled for rotation with the scheduled interval, date, and time at which the rotation is to take place. In addition, the name of the user responsible for scheduling the rotation displays.

Manually Rotate a Key (Add a New Version of a Key)

To rotate a key manually (add a new version of a key):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. In the **All Keys** list, find the name of the key to which to add a new version.
3. Select **Add Version** from the **Actions** drop-down menu. The **Add a Version** dialog box is displayed.
4. From the **Key Provider** box, select one of the following as the key source to which to add a new version:
 - **DSM**—Key was generated from DSM and imported into a Standard or Premium Azure Key Vault.
 - **CipherTrust Manager**—Key was generated from CipherTrust Manager and imported into a Standard or Premium Azure Key Vault.
 - **HSM.byok**—Your byok key was generated from an external Thales HSM and imported into a Premium Azure Key Vault from CCKM.
 - **PFX**—Key was imported as an external password-protected PFX key into a Standard or Premium Azure Key Vault.
 - **nShield**—Key was generated from nShield Connect and imported into a Standard or Premium Azure Key Vault.
 - **Native**—Key is generated natively in the designated Azure Key Vault using the CCKM UI. Note that this option is equivalent to generating a key in the Azure portal.
5. From the **PFX-File Password** box, enter your password for your PFX file. This box only displays, if you selected PFX as your source key tier.
6. From the **Source Key** box, select the source key to which to add a new version. Based on the selected source key tier and source key, the **Key Vault Tier**, **Key Vault Region**, and **Key Name** boxes are prepopulated with the values associated with your selections.
7. From the **Key Destination** box, select either **Software** or **HSM** (Premium subscription only) depending on which destination you want to use.
8. Click **Enabled**, if the key is enabled. Note that if the key is disabled, the key cannot be used for cryptographic operations. This option may already be selected depending on the selected source key.
9. From **Key Type**, select **RSA** or **EC** for your key type. **Key Type** only displays, if you selected **Native** as your source key tier. If you select **RSA**, then select **2048**, **3072**, or **4096** from **Key Size** for the key size to apply to the new version of the key. If you select **EC**, then select **P-256**, **P-384**, **P-521**, or **SECP256K1** from **Elliptical Curve Name** for the name of the elliptical curve to apply to the new version of the key.
10. Enter the activation date of the new version of the key in the **Activation Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
11. Enter the expiration date of the new version of the key in the **Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
12. From **Key Operations**, click the cryptographic operations the new version of the key will perform. The options are:
 - Encrypt
 - Decrypt
 - Sign
 - Verify
 - Wrap Key
 - UnwrapKey

13. (Optional) From **Tags**, enter the tag name and value of the new version of the key.
14. Click **Upload to Azure**.
The new version of the key is displayed in the list. As part of the process of adding a new key version of a key, Azure generates a corresponding backup key blob, which CCKM downloads and stores in DSM.

Enable Autorotation of Key

To enable autorotation of a key (add a new version of the key at a scheduled time):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. In the **All Keys** list, find the name of the key for which to enable autorotation.
3. Select **Auto Rotate** from the **Actions** drop-down menu. The **Auto-rotate Key** dialog box is displayed. The **Key Algorithm** box is populated with the algorithm used in the selected key to autorotate.
4. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.

Note: If the toggle for **Enable autorotation on key expiration?** is displayed, skip it. This option displays only if the key has an expiration date associated with it. This option is not applicable to enabling autorotation of the key.

5. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
6. Select **Change key algorithm**, if you choose to use a different key algorithm than the one displayed in **Key Algorithm** for the new version of the key. If you choose to use a different key algorithm, **Key Provider**, **Key Destination**, and **Key Type** display for configuration.
7. From the **Key Provider** box, select one of the following as the key source to which to add a new version:
 - **DSM**—Key is to be generated from DSM and imported into a Standard or Premium Azure Key Vault.
 - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager and imported into a Standard or Premium Azure Key Vault.
 - **nShield**—Key is to be generated from nShield Connect and imported into a Standard or Premium Azure Key Vault.
 - **Native**—Key is to be generated natively in the designated Azure Key Vault using the CCKM UI. Note that this option is equivalent to generating a new key in the Azure portal.
8. From the **Key Destination** box, select either **Software** or **HSM** depending on whether you want to use the Azure Standard or Premium subscription service. If you are using the Standard subscription service for the selected key to be autorotated, then Software is selected. If you are using the Premium subscription service for the selected key to be autorotated, then HSM is selected. However, you have the option to select Software to use the Standard subscription service for the new version of the key.

9. From **Key Type**, select **RSA** or **EC** for the key type of the new version of the key.
 - If your source key tier is set to **DSM**, then the key type of **RSA** along with the key sizes of **2048**, **3072**, and **4096** for **Key Size** are your options.
 - If your source key tier is set to **CipherTrust Manager**, then the key type of **RSA** along with the key sizes of **2048**, **3072**, and **4096** for **Key Size** are your options.
 - If your source key tier is set to **nShield**, then the key type of **RSA** along with the key sizes of **2048**, **3072**, and **4096** for **Key Size** are your options.
 - If your source key tier is set to **Native**, then the key types of **RSA** and **EC** are your options. For **RSA**, the key sizes **2048**, **3072**, and **4096** for **Key Size** are your options. For **EC**, the name of the elliptical curves **P-256**, **P-384**, **P-521**, and **SECP256K1** for **Elliptical Curve Name** are your options.
10. Click **Submit**.

Synchronize Keys

Synchronizing allows you to download all the keys that were created in Azure (to which you have access) into CCKM. You have the option to schedule key synchronization. Perform this scheduling in the **Schedules** page.

The story below illustrates a use of synchronization and restoration:

Sally is the Azure Cloud administrator with access to many vaults. She has synchronized keys in CCKM. Now, innocent, ignorant rogue Charlie logs into Azure portal and deletes all the keys in his vault, which also belonged to Sally. Now, if Sally synchronizes keys again from CCKM portal, she will see those keys as deleted in CCKM portal. However, she will also see options to restore the deleted keys back into the relevant Azure Key Vault. This is because CCKM also maintains Azure backup keys as part of basic key management.

To synchronize your keys:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. Click **Synchronize**. A warning box is displayed; click **Synchronize** again to proceed with the synchronization. New keys that are created in Azure are retrieved and displayed in the CCKM UI.

Enable Autorotation before Key Expiration

To enable autorotation of a key before it expires, you must first set up a key expiration schedule in the **Schedules** page. Once the key is enabled for autorotation, the key will be rotated before it expires. After the key is autorotated, the key will have a new expiration date based on the key-expiration schedule.

To enable autorotation of a key before it expires:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. In the **All Keys** list, find the name of the key for which schedule a key rotation.
3. Select **Auto Rotate** from the **Actions** drop-down menu. The **Auto-rotate Key** dialog box is displayed. The **Key Algorithm** box is populated with the algorithm used in the selected key to autorotate.
4. From **Enable autorotation on key expiration?**, select **On** to enable the autorotation of this key before it expires. By default, this toggle is initially set to **Off**.

Note: Skip the **Enable autorotation on this key?** as it is not applicable to enabling autorotation of this key before it expires.

5. From the **Schedule** drop-down menu, select a key-expiration schedule to which to apply to this key. Any key-expiration schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.

6. Select **Change key algorithm**, if you choose to use a different key algorithm than the one displayed in **Key Algorithm** for the new version of the key. If you choose to use a different key algorithm, **Source Key Tier**, **Key Destination**, and **Key Type** display for configuration.
7. From **Source Key Tier**, select one of the following as the key source of the new version of the key:
 - **DSM**—Key is to be generated from DSM and imported into a Standard or Premium Azure Key Vault.
 - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager and imported into a Standard or Premium Azure Key Vault.
 - **nShield**—Key is to be generated from nShield Connect and imported into a Standard or Premium Azure Key Vault.
 - **Native**—Key is to be generated natively in the designated Azure Key Vault using the CCKM UI. Note that this option is equivalent to generating a new key in the Azure portal.
8. From the **Key Destination** box, select either **Software** or **HSM** depending on whether you want to use the Azure Standard or Premium subscription service. If you are using the Standard subscription service for the selected key to be autorotated, then Software is selected. If you are using the Premium subscription service for the selected key to be autorotated, then HSM is selected. However, you have the option to select Software to use the Standard subscription service for the new version of the key.
9. From **Key Type**, select **RSA** or **EC** for the key type of the new version of the key.
 - If your source key tier is set to **DSM**, then the key type of **RSA** along with the key sizes of **2048** and **4096** for **Key Size** are your options.
 - If your source key tier is set to **CipherTrust Manager**, then the key type of **RSA** along with the key sizes of **2048**, **3072**, and **4096** for **Key Size** are your options.
 - If your source key tier is set to **nShield**, then the key type of **RSA** along with the key sizes of **2048**, **3072**, and **4096** for **Key Size** are your options.
 - If your source key tier is set to **Native**, then the key types of **RSA** and **EC** are your options. For **RSA**, the key sizes **2048**, **3072**, and **4096** for **Key Size** are your options. For **EC**, the name of the elliptical curves **P-256**, **P-384**, **P-521**, and **SECP256K1** for **Elliptical Curve Name** are your options.
10. Click **Submit**.

Export Keys

Export Keys allows you to export the contents of the Keys list into a report in a CSV format.

1. In the left navigation bar, click **Keys**.
2. Click **Export Keys**.

This will download all of the keys' versions in a report in a CSV format.

Backup Key

When a key is uploaded to Azure from CCKM, Azure creates a backup key (as an encrypted blob), which contains information about the Azure key and all its versions. The backup key can be used to restore a key that has been deleted. After Azure creates a backup key, CCKM automatically downloads and stores the key in the key source you are using (CipherTrust Manager or DSM), or in MongoDB (if you are using nShield Connect as your key source). In addition, when Azure keys are synchronized, the backup key of any key that was created directly in the Azure portal is also downloaded and stored in the key source you are using (CipherTrust Manager or DSM), or in MongoDB (if you are using nShield Connect as your key source).

Note

CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Delete a Backup Key

Delete a backup key when you definitely do not want the key and have no corresponding data that will ever need decryption.

For a key that has been deleted:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list page, select **Delete Backup** from the action column of a deleted key.
3. The **Delete Key** warning page is displayed. Type or Copy/paste the confirmation phrase (I wish to delete the Azure backup key).
4. Click **Delete**.

The key is removed from the list.

Reports

CCKM includes the following reports:

- **Combined Key Activity Reconciliation Report**

Provides the history of the key in a selected period. Click Run Report to generate a new list of events connected to a key.

- **Azure Key Activity Report**

Illustrates the activity type performed by a specific user and at the stated time.

- **Azure Key Service Usage Report**

Indicates the relationship between the application and the key, including:

- Key name
- Key Description
- Requesting Service: Name or ID of the service using a key
- Service Platform
- Request Timestamp: Time when the application used the key.

- **Azure Key Aging Report**

Provides information about key expiration using the following columns:

- Key Name
- Key description
- Key destination: Indicates services, such as Azure
- Key owner: Person who created the key. In Azure, there is no owner associated with the key.
- Key expiration: Date and time at which this key expires.

- **Cloud Key Manager User Action Report**

Lists the users and their actions in creating and deleting keys in the CCKM portal.

Schedules

The **Schedules** page allows Cloud administrators to view and schedule the following job types:

- Key Rotation
- Key Synchronization
- Key Expiration

CCKM provides support for multiple schedules for each of the job types.

At the top right of the **Schedules** page, the **Add Schedule** button is available from which to add a scheduled job type.

Note

If you are using CCKM as a service principal, you are not required to enter a username and password to schedule key rotation, key synchronization, or key expiration for Azure Key Management. The username and password boxes do not display in the dialog box for each of these job types.

Schedule Key Rotation

CCKM provides support for multiple schedules of the rotation of an Azure key. CCKM allows for a scheduled rotation (autorotation) of an Azure key. A new version of the key is added to the Azure key during a scheduled rotation. Scheduled rotation requires that this feature be configured in the **Schedules** page prior to enabling the feature in the **Auto Rotate Key** dialog box (from the **Actions** column of the **All Keys** list). Your Azure login password is required as part of the configuration and is saved in the key source you are using (CipherTrust Manager or DSM), or in MongoDB (if you are using nShield Connect as your key source). If you have both DSM and nShield Connect configured as your key sources, then the Azure login password is saved only to DSM where it is stored.

Note

CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

During a scheduled rotation of a key, CCKM uses the associated username and password to rotate the key. If the password expired or changed, then the new password must be reentered. Otherwise, the scheduled rotation will fail.

You can pause or delete a scheduled key rotation in the **Schedules** page. Note that when a key rotation schedule is deleted from CCKM, then the Azure user password is also deleted from the key source you are using (CipherTrust Manager or DSM) or from MongoDB (if you are using nShield Connect as your key source).

Set up a Key Rotation Schedule

To set up a key rotation schedule:

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.

Note: If you are using CCKM as a service principal, the **Username** box does not display as you are not required to enter a username when using CCKM as a service principal

3. From the **Job Type** drop-down menu, select **KEY_ROTATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.

6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Enter your Azure user password in the **User Password** box.

Note: If you are using CCKM as a service principal, the **User Password** box does not display as you are not required to enter a password when using CCKM as a service principal.

9. Click **Save**.

Pause a Key Rotation Schedule

To pause a key rotation schedule, select **Yes** for **Pause** within the **Add Schedule** dialog box and then click **Save**. The setting is saved. To resume the rotation schedule after pausing it, select **No** for **Pause** and then click **Save**. The setting is saved.

Delete a Key Rotation Schedule

To delete a key rotation schedule, select **Delete** from the **Actions** drop-down menu from the **Schedules** page. Note that the Azure user password associated with the given key rotation schedule is also deleted from the key source you are using (CipherTrust Manager or DSM) or from MongoDB (if you are using nShield Connect as your key source).

Schedule Key Synchronization

CCKM allows for multiple schedules of the synchronization of Azure keys. Synchronizing downloads any keys that were created in Azure, outside of the CCKM portal, into your CCKM. Scheduled key synchronization requires that this feature be configured in the **Schedules** page. You can also pause or delete a scheduled key synchronization in the **Schedules** page. Note that when a key synchronization schedule is deleted from CCKM, then the Azure user password is also deleted from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as the key source). If you have both DSM and nShield Connect configured as your key sources, then the Azure user password is deleted from DSM where it is stored.

Note

CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Set up a Key Synchronization Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.

Note: If you are using CCKM as a service principal, the **Username** box does not display as you are not required to enter a username when using CCKM as a service principal.

3. From the **Job Type** drop-down menu, select **KEY_SYNCHRONIZATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the key synchronization by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key synchronization by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Enter your Azure user password in the **User Password** box.

If you are using CCKM as a service principal, the **User Password** box does not display as you are not required to enter a password when using CCKM as a service principal.
9. Click **Save**.

Schedule Key Expiration

CCKM provides support for multiple schedules of the expiration of Azure keys in the **Schedules** page. You can also pause or delete a scheduled key expiration in the **Schedules** page. Note that when a key expiration schedule is deleted from CCKM, then the Azure user password is also deleted from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as your key source). If you have both DSM and nShield Connect configured as your key sources, then the Azure user password is deleted from DSM where it is stored.

Note

CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Set up a Key Expiration Schedule

To set up a key expiration schedule:

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.

Note: If you are using CCKM as a service principal, the **Username** box does not display as you are not required to enter a username when using CCKM as a service principal.

3. From the **Job Type** drop-down menu, select **KEY_EXPIRATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. From **Start Date**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.
8. From **Expire a new key after**, enter a number and select the interval unit of Hour, Day, Week, Month, or Year from the drop-down menu after which the new key expires. The supported ranges are 6 through 24 hours, 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.
9. Enter your Azure user password in the **User Password** box.
If you are using CCKM as a service principal, the **User Password** box does not display as you are not required to enter a password when using CCKM as a service principal.
10. Click **Save**.

Logs

CCKM maintains logs for events, such as login, key generation, synchronizing of keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. For each logged event listed on the **Logs** page, the following information is displayed:

- Event name
- Severity level
- Date and time of the event
- Event message
- User (name) associated with the event

You can search for a specific logged event from the **Logs** page using the **Search** box. Allowable filters on the search are event name, severity level, event message, and user (name).

Key Sources

The **Key Sources** page displays **all** of the keys that were generated from the key source you are using. If you are logged into multiple clouds within CCKM, this page displays all of the keys from these clouds.

Three tabs are available at the top of the **Key Sources** page:

- **CipherTrust Manager Key:** Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.
- **DSM Key:** Use this tab to use DSM as your key source. The keys are generated and stored within DSM.
- **nShield Key:** Use this tab to use nShield Connect HSM as your key source. The keys are generated within nShield Connect and then stored as blobs within MongoDB.

Note

CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Keys listed on the **Key Sources** page are listed by name, cloud ID, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description and actions you can take regarding that key.

A **Search** box is available at the top left of the **Keys Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Cloud ID
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

Add (Generate) a Key

From the **Key Sources** page, you can add (or generate) keys in CCKM using CipherTrust Manager, DSM, or nShield Connect as a key source for *all* of the clouds into which you are currently logged.

Note

The key algorithm currently supported for an Azure key in CCKM is RSA.

Add (Generate) a CipherTrust Manager Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **CipherTrust Manager Key** tab.
3. Select **Add Key > Azure Key**.
4. In the **Add CipherTrust Manager Key** dialog box, select the name of the user of the key from the **User** drop-down menu. Service is fixed to **Azure**.
5. From the **Source Key Tier** box, select **CipherTrust Manager** from the drop-down menu.
6. In the **Name** box, enter the name of the new key.
7. Enter a description of the key in the **Description** box.

8. From the **Algorithm** box, select the algorithm (**RSA2048**, **RSA3072**, or **RSA4096**) to apply to the key.
9. Click **Save** to save the new key to CCKM. The new key is added to the keys list. (Reset clears the screen before it has been saved.)

Add (Generate) a DSM Key

From the **Key Sources** page, the **DSM Key** tab allows you to generate a .byok key from an external HSM (such as nShield Connect HSM) and import it into a Premium Azure Key Vault from CCKM. From the **DSM Key** tab, you can also import an external password-protected PFX key into a Standard or Premium Azure Key Vault. Once imported from CCKM, the .byok and PFX key is stored in DSM securely as blobs.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key** tab.
3. Select **Add Key > Azure Key**.
4. In the **Add DSM Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to Azure.
5. In the **Source Key Tier**, select one of the following as a key source of the key:
 - **DSM**—Generate a key from DSM and import it into a Standard or Premium Azure Key Vault.
 - **HSM.byok**—Generate a .byok key from an external Thales HSM and import it into a Premium Azure Key Vault from CCKM. Once imported from CCKM, this .byok key is stored in the DSM securely as a blob. The .byok key is user and region specific. Be sure to select the correct username and region when creating the .byok key.
 - **PFX**—Import an external password-protected PFX key into a Standard or Premium Azure Key Vault. Once imported from CCKM, this PFX key is stored in the DSM securely as a blob.
6. In the **Name** box, enter the name of the new key.
7. Enter a description of the key in the **Description** box.
8. From the **Algorithm** box, select the algorithm (**RSA2048**, **RSA3072**, or **RSA4096**) to apply to the key.
9. Click **Save** to save the new key to CCKM. The new key is added to the keys list. (Reset clears the screen before it has been saved.)

Add (Generate) an nShield Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **nShield Key** tab.
3. Select **Add Key > Azure Key**.
4. In the **Add nShield Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to Azure.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. From the **Algorithm** box, select the algorithm (**RSA 2048**, **RSA 3072**, or **RSA 4096**) to apply to the key.
8. Click **Save** to save the new key to CCKM. The new key is added to the keys list. (Reset clears the screen before it has been saved.)

Delete a Key

Deleting a key from the **Key Sources** page deletes the key from CCKM. When deleting a key from CCKM, the key is deleted from the key source you used to generate the key. However, if you generated the key from nShield Connect, the key is deleted from MongoDB where the key is stored.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key**, **CipherTrust Manager**, or **nShield Key** tab depending on the key source you used to generate this key initially.
3. Click **Delete** next to the target key.
4. The **Delete Key** dialog box with a Warning display asking you to confirm the key deletion by entering the supplied phrase.
5. Enter or copy/paste the confirmation phrase.
6. Click **Delete**.

Manage Keys in Key Sources List Page

In addition to adding and deleting keys, the **Key Sources** page allows you to sort the existing keys by Name, Key Type, Algorithm and Created Date.

Settings

To access the **Settings** page in CCKM, select **Settings** from the left-hand navigation bar. From the **Settings** page, you can:

- Enter the email addresses of the recipients who are to receive the email notifications.
- Enable email notifications to be sent out from CCKM when specific types of key actions are performed. For example, when a key is uploaded to the cloud service. You can also enable CCKM to send reminder notifications for when to rotate a key.
- Configure remote syslog servers to which to send syslog messages.
- (Applicable to CCKM when used as a service principal) Change your tenant password or revoke your administrator consent.
- View information about the software release number of the CCKM you are running including the build and version number.

Note that for Azure, each of the supported clouds (Azure and Azure China) must be set up independently of each other, as these are separate and distinct key management services within Azure.

Adding Recipients' Email Addresses (for Notifications)

Use the **General** tab from the **Settings** page to enter the email addresses of the recipients who are to receive email notifications from CCKM. Enter the recipients' email addresses separated by a comma in the **Notification Recipient** box and then click **Save** to save setting.

Enabling Notifications

Using the **Keys Notification** tab from the **Settings** page, you can configure CCKM to send an email notification when a key is deleted from Azure or restored or uploaded to Azure. You can also use this tab to configure CCKM to send a reminder notification for when to manually rotate a key along with the frequency (in days) at which a

notification is sent. Add the email address(es) of the recipients in the **General** tab from the **Settings** page. By default, no notifications are sent

To enable email notifications:

1. In the **Settings** page, select the **Keys Notification** tab.
2. Under **Notifications**, select **On** for each of the notifications you wish to enable:
 - Rotate Azure Key (reminder notification)
 - Delete Azure Key
 - Restore Azure Key
 - Upload Azure Key
3. If you selected to enable **Rotate Azure Key**, in the **Days** box, enter the number of days for the frequency a reminder notification is sent.
4. Click **Save**.

Configuring Syslog Server

Note

In this document, the terms "syslog server" and "log server" are synonymous.

The **Syslog** tab from the **Settings** page allows you to configure the remote syslog or log server to which to send CCKM log data using TCP, TLS, or UDP protocol as the transport protocol. If you use the TLS protocol, then you are required to upload the syslog server certificate to CCKM for the TLS connection.

The **Syslog** tab also allows you to delete your syslog server configuration so that log data are no longer sent to the server.

Prior to configuring the setting for a remote syslog server, ensure that you have installed and configured the syslog server including configuring the setting for the port number the transport protocol will use to connect to CCKM. This port number configuration must match the port configuration you set in the **Syslog** tab of the **Settings** page.

Note

Support for both Syslog and Splunk as a remote syslog server is provided in CCKM. However, the use of a Splunk cloud server as a remote syslog server is currently not supported in CCKM.

Configure Remote Syslog Server

To configure a remote syslog server to which to send CCKM log data:

Note

The facility name of "LOCAL1" for the syslog server displays in the **Facility** box. This default value is not available for modification.

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. In the **Hostname** box, enter the hostname or IP address of the syslog server.
3. In the **Port** box, enter the port number the transport protocol uses to connect to the syslog server. Enter a value between 1 and 65535. The default value is 514.
4. In the **Protocol** drop-down menu, select **TCP**, **TLS**, or **UDP** for the transport protocol to use when sending the log data to the syslog server. The default value is **UDP**. If you select **TCP**, proceed to Step 6. If you select **TLS**, proceed to the next step. Otherwise, proceed to Step 7.

5. In the **Certificate** box, select **Choose File** and then select the syslog server certificate to upload to CCKM.
6. Click **Test Connection** to test the connection between CCKM and the remote syslog server. This button is only available for use when either **TCP** or **TLS** is the selected transport protocol.
7. In the **Log Format** drop-down menu, select **CEF** (Common Event Format), **Plain Text**, or **RFC5424** for the format of the log message to use when sending the log data. The default value is **RFC5424**.
8. (Optional) In the **Description** box, enter a description of the remote syslog server.
9. Click **Save**.

Delete Remote Syslog Server Configuration

To delete the configuration of a remote syslog server:

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. Click **Delete**. The **Delete Syslog** dialog box displays.
A warning message displays indicating that this action will delete the syslog settings from CCKM. The message also indicates to type the confirmation phrase (I wish to delete the syslog settings) in the text box if you still wish to delete the syslog settings.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Delete**.

Configuring Tenant Password and Revoking Admin Consent (for Service Principal)

Note

The **Tenant** tab is available in the **Settings** page only if you have configured to use CCKM as a service principal.

The **Tenant** tab in the **Settings** page allows you to change your tenant password that is associated with the use of CCKM as a service principal.

Note

The **Tenant** tab is available in the **Settings** page only if you have configured to use CCKM as a service principal.

The **Tenant** tab also allows you to revoke the administrator consent, which you (as the administrator user) had provided during the initial login into CCKM for Azure Key Management (using CCKM as a service principal). See ["Access CCKM for Azure" on page 118](#), for more information about providing your administrator consent. If you revoke the administrator consent and then logout of CCKM, you will receive an error message when you attempt to log back into CCKM. The message indicates that the login failed due to having entered an incorrect tenant name or the given tenant not having granted permissions to CCKM (to be used as a service principal). You must provide the administrator consent again to login into CCKM. Click on the "admin consent" link (indicated by "here" in blue text at the top CCKM login dialog box) after which the CCKM Admin Consent dialog box is displayed.

Change Tenant Password

To change a tenant password that is associated with CCKM used as a service principal:

1. In the **Settings** page, select the **Tenant** tab. The **Tenant** dialog box displays.
2. Enter the new tenant password in the **New Tenant Password** box.
3. Reenter the new tenant password in the **Confirm Tenant Password** box.
4. Click **Save**.

Revoke Admin Consent

To revoke your administrator consent that is associated with CCKM used as a service principal:

1. In the **Settings** page, select the **Tenant** tab. The **Tenant** dialog box displays.
2. Deselect the **Granted Permission to this CCKM** box to revoke your administrator consent. By default, this box is checked.
3. Click **Save**.

About

Use the **About** tab to view the release number of the CCKM you are running.

Chapter 15: CCKM for Google Key Management

Overview	147
About Google Cloud Keys	150
Prerequisites on Google Cloud	151
Access CCKM for Google Key Management	152
Google CCKM User Portal	152
Home	154
Keys	154
Reports	172
Schedules	173
Logs	175
Key Sources	176
Settings	178

Overview

CCKM provides support for managing Google Cloud customer-managed encryption keys (CMEK). CMEK are used to encrypt data at rest within a Google Cloud project. In addition, several Google Cloud services offer the ability to protect data using encryption keys managed by customers within the Cloud Key Management System (KMS). CCKM can control the lifecycle of these keys by automatically or manually rotating CMEK. It also controls Google's ability to decrypt data at rest by disabling the key. Data is protected using a key that meets specific locality requirements using either one of the supported key sources (Data Security Manager (DSM) or CipherTrust Manager) with the CCKM solution. Additionally, CCKM can manage CMEK lifecycle in an automated way using Google Service account for REST API authentication.

Note

This documentation assumes that you are familiar with using Google Cloud Platform, Google Cloud KMS, and Google Identity and Access Management (IAM). For details about these Google technologies, refer to the Google Cloud documentation at <https://cloud.google.com/docs>.

Google Cloud IAM

Using Google Cloud's Identity and Access Management (IAM) allows you to manage access to Google Cloud resources by granting access to these resources. By defining the identity, role, and resource(s) in IAM to which to be given access, you are able to manage access to Google Cloud resources. Access permissions are grouped into roles and an authenticated Google member is granted at least one role to access to the resources.

IAM has three parts as described in the following sections.

Member

A member can be:

- Google Account (for end users)
- Service account (for apps, such as CCKM, and virtual machines)

- Google group (can contain multiple user and service accounts)
- G Suite domain
- Cloud Identity domain

An email address associated with a user, service account, or Google group is a member's identity. The G Suite and Cloud Identity domains each use the associated domain name for its identity.

CCKM supports the use of Google user, group, and service accounts. You will need the user account to login to CCKM UI, and you will need service account to authenticate to CCKM REST API.

Roles

A role is a collection of permissions. There are three kinds of roles in Cloud IAM:

- Primitive roles: Owner, Editor, Viewer.
- Predefined roles: Predefined finer-grained access control roles. For your user account or service account to use CCKM, you need to grant Project Browser and Cloud KMS Admin roles to the account. Note that the Cloud KMS admin role grants all the permissions required to use CCKM. If your project belongs to an organization, then it is recommended you grant Organization Viewer role to your accounts. Granting this role allows you to view the organization name to which your user or service account belongs in CCKM. Otherwise, you are only able to view the organization ID associated with your user or service account.
- Custom roles: User defined roles (*recommended*). If you wish to grant specific permissions to users or groups to meet your specific requirements, then create custom roles. For example, one user or group should only be able to view keys but not perform any key operations. Another user or group should only be able to list keys and perform any operation with the exception of deleting key versions. See [Table 15-1: "\[CCKM Action Mapping to IAM KMS Permissions\]" below](#) for the permissions that are required for each of the supported CCKM actions. See ["Create Custom Roles" on page 53](#) for information on how to create custom roles.

Table 15-1: [CCKM Action Mapping to IAM KMS Permissions]

CCKM Action	Cloud IAM KMS Permissions
Add key version import	• cloudkms.importJobs.create • cloudkms.importJobs.setIamPolicy • cloudkms.importJobs.get • cloudkms.importJobs.useToImport • cloudkms.cryptoKeyVersions.create • cloudkms.cryptoKeys.update
Add key version native	• cloudkms.cryptoKeyVersions.create • cloudkms.cryptoKeys.update
Authenticate	• resourceManager.organizations.get • resourceManager.projects.get
Cancel schedule delete key material	cloudkms.cryptoKeyVersions.restore
Create key	cloudkms.cryptoKeys.create
Rotate key import	• cloudkms.importJobs.create • cloudkms.importJobs.setIamPolicy • cloudkms.importJobs.get • cloudkms.importJobs.useToImport • cloudkms.cryptoKeyVersions.create • cloudkms.cryptoKeys.update

Table 15-1: [CCKM Action Mapping to IAM KMS Permissions] (continued)

CCKM Action	Cloud IAM KMS Permissions
Rotate key native	- cloudkms.cryptoKeyVersions.create - cloudkms.cryptoKeys.update
Schedule delete key material	cloudkms.cryptoKeyVersions.destroy
Synchronize	- cloudkms.keyRings.list - cloudkms.keyRings.getIamPolicy - cloudkms.cryptoKeys.getIamPolicy - cloudkms.cryptoKeys.list - cloudkms.cryptoKeyVersions.list
Update key	cloudkms.cryptoKeys.update
Update key version	- cloudkms.cryptoKeys.update - cloudkms.cryptoKeyVersions.update
Upload key	- cloudkms.cryptoKeys.create - cloudkms.importJobs.create - cloudkms.importJobs.setIamPolicy - cloudkms.importJobs.get - cloudkms.importJobs.useToImport - cloudkms.cryptoKeyVersions.create - cloudkms.cryptoKeys.update

For more information about Google Cloud roles, see [Understanding Roles within Google's IAM](#).

Resource

Google Cloud resources are organized hierarchically:

- Organization: root node of the hierarchy
- Folders: children of the organization
- Projects: children of the organization or of a folder
- Resources: descendants of projects

A Cloud IAM policy grants a member permissions to a resource. Google Cloud KMS permissions can be granted on the following types of resources:

- **project** (*not recommended*): a member that is granted a role or one permission on a project will be given that privilege on all resources contained in the project. Resources contained in a project can be KMS resources or other service resources. KMS resources are key rings and all crypto keys contained in those key rings.
- **key ring**: a member that is granted a role or one permission on a key ring will be given that privilege on all crypto keys contained in the key ring.
- **crypto key**: a member that is granted a role or one permission on a crypto key will be given that privilege on all crypto key versions contained in the key.

About Google Cloud Keys

Google Cloud KMS stores cryptographic keys in a hierarchical structure. Here are levels starting from the top:

- Project
- Location
- Key ring
- Key
- Key version

When you create a key in Google Cloud KMS, you define the purpose and the algorithm of that key (see [Table 15-2: "\[KMS Keys\]" below](#) for each of the available algorithms per key purpose). You have the option to change the algorithm when you create a new key version (depending on the scope of its purpose.). However, you cannot change the key purpose after a key is created.

Google Cloud KMS provides the following key purposes:

- ENCRYPT_DECRYPT (used in Symmetric encryption)
- ASYMMETRIC_SIGN (used in Asymmetric signing)
- ASYMMETRIC_DECRYPT (used in Asymmetric encryption)

The protection level, which you cannot change once the key is created, indicates how the cryptographic operations are performed. Google Cloud KMS provides the following protection levels:

- Software: software is used for cryptographic operations
- HSM: an HSM is used for cryptographic operations
- EXTERNAL (*Not currently supported in CCKM*): An external key manager stores the key used in the cryptographic operations. Only applicable to symmetric encryption.

For each key type (and subtype), the following table provides the key purpose and supported algorithm(s) and protection levels.

Table 15-2: [KMS Keys]

Key Type & Subtype	Key Purpose	Supported Algorithms	Protection Level
Symmetric <i>n/a</i>	ENCRYPT_DECRYPT	GOOGLE_SYMMETRIC_ENCRYPTION	SOFTWARE HSM
Asymmetric <i>signing</i>	ASYMMETRIC_SIGN	• EC_SIGN_P256_SHA256 • EC_SIGN_P384_SHA384 • RSA_SIGN_PSS_2048_SHA256 • RSA_SIGN_PSS_3072_SHA256 • RSA_SIGN_PSS_4096_SHA256 • RSA_SIGN_PSS_4096_SHA512 • RSA_SIGN_PKCS1_2048_SHA256 • RSA_SIGN_PKCS1_3072_SHA256 • RSA_SIGN_PKCS1_4096_SHA256 • RSA_SIGN_PKCS1_2048_SHA512	SOFTWARE HSM

Table 15-2: [KMS Keys] (continued)

Key Type & Subtype	Key Purpose	Supported Algorithms	Protection Level
Asymmetric <i>encryption</i>	ASYMMETRIC_ENCRYPT	• RSA_DECRYPT_OAEP_2048_SHA256 • RSA_DECRYPT_OAEP_3072_SHA256 • RSA_DECRYPT_OAEP_4096_SHA256 • RSA_DECRYPT_OAEP_4096_SHA512	SOFTWARE HSM

Prerequisites on Google Cloud

The Google Cloud administrator must ensure the following prerequisites are met prior to configuring and accessing the Google Cloud KMS in CCKM:

- Create at least one project. For information on how to create a project, see [Creating and managing projects](#) within the Google Cloud online documentation.
- Enable Cloud Resource Manager APIs. See ["Enable Cloud Resource Manager API" on page 49](#) for more information.
- Enable IAM APIs. See ["Enable Identity and Access Management API" on page 49](#) for more information.
- Enable Cloud KMS service and create a key ring in the project. See ["Enable Cloud Key Management Service \(KMS\) API" on page 49](#) for information about enabling the Cloud KMS service.
- Create a user account and a service account. You will need the user account to login to CCKM UI, and the service account to authenticate to CCKM REST API. See ["Create Service Account" on page 51](#) for more information.
- If your Google user account belongs to an organization, ensure your security policy for Google Cloud session control is set to **Session never expires**. See ["Set Session Duration for Google Cloud Console" on page 50](#) for more information.
- Assign the roles of Project Browser and Cloud KMS Admin to the user account and service account. See ["Grant IAM Roles to User and Service Accounts" on page 52](#) for more information.
- If you will use the user account to add a service account to CCKM, assign the Service Account Key Admin role to the user account. Refer to the Google Cloud online documentation for more information.
- (Optional) If your project belongs to an organization, assign the Organization Viewer role to your user account and service account. See ["Grant IAM Roles to User and Service Accounts" on page 52](#) for more information.
- (Optional) If you wish to grant specific permissions to users or groups, then create custom roles. See ["Prerequisites on Google Cloud" above](#) table for the permissions that are required for each of the supported CCKM actions. See ["Create Custom Roles" on page 53](#) for information.
- If you need to generate CCKM reports, assign the following roles to your user account and service account:
 - Logging/Logs Viewer or Project/Viewer
 - Logging/Private Logs Viewer or Project/Owner

CCKM accesses the Admin Activity audit logs and Data Access audit logs in Google Cloud to generate reports. These audit logs must be retained at log buckets in Google Cloud. *Ensure* logs are enabled in Google Cloud.

For information about Google Cloud logs, refer to <https://cloud.google.com/logging/docs/storage>. For information about log retention periods, refer to https://cloud.google.com/logging/quotas#logs_retention_periods.

- Create OAuth2 Client ID credentials for CCKM app. Note the Client ID and secret. You will need these credentials when adding Google Cloud service to CCKM using the CCKM Admin Portal. See "[Create OAuth2 Client ID Credentials](#)" on page 50 for more information.

Access CCKM for Google Key Management

If you will use CCKM UI to manage your Google Cloud keys, then you will need to use your username and password for Google Cloud to login to the CCKM user portal. A Google Cloud service account cannot be used to login to the CCKM user portal. If you will use CCKM REST APIs to manage your Google Cloud keys, then you will need a Google Cloud service account. A Google Cloud user account cannot be used with CCKM REST APIs.

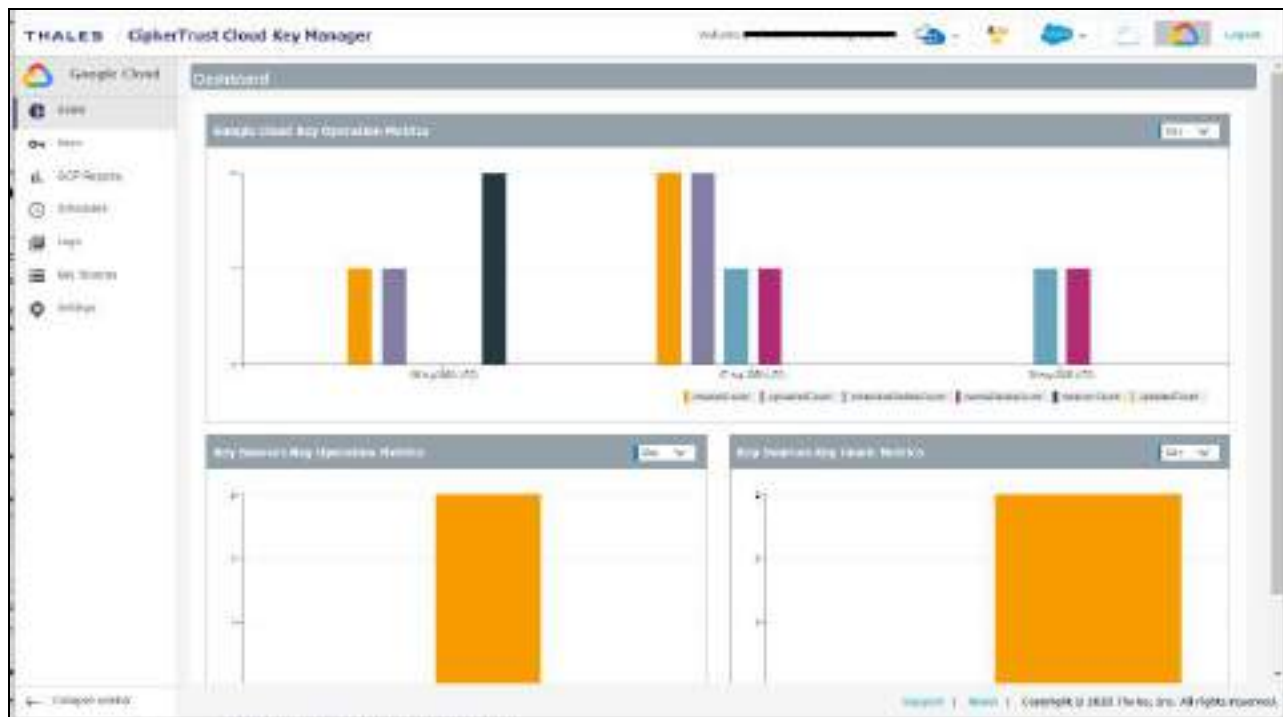
Google CCKM User Portal

The CCKM User Portal for Google Cloud contains the following menu options in the left-hand navigation bar:

- **Home:** Home page displays key operation and count metrics.
- **Keys: Google Cloud Keys** page displays key management overview and provides the functionalities to create a native key in Google Cloud KMS, upload your "BYOK" key, synchronize keys and key rings, and export the contents of the keys list into a report. Also available are the key operations you can perform for both Asymmetric and Symmetric keys and the key versions. For Asymmetric keys, the key operations of Update Key and Add Version are available. For the symmetric keys, the key operations of Update Key and Rotate Key are available. For the key versions of both types of keys, the actions of View all versions, Enable, Disable, Schedule Destroy, and Cancel Destroy are provided. Additionally, for the key versions of the Asymmetric keys, the action of Download (Public Key) is also provided.
- **Reports: All Reports** overview page links to detailed individual reports.
- **Schedule:** In **Schedule**, you can schedule automatic key rotation of Symmetric keys and schedule to add a new key version of Asymmetric keys. Additionally, you have the option to schedule key synchronization.
- **Logs: Logs** page lists individual logs that can be searched, sorted, and viewed.
- **Key Sources:** In **Key Sources**, you can create keys in CCKM to upload to Google Cloud. CCKM provides support for the use of CipherTrust Manager and DSM as key sources to generate, store, and retrieve the encryption keys that CCKM uses and manages.

Note: CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

- **Settings:** The **Settings** page allows you to add the email addresses of the recipients who are to receive email notifications and configure the settings for email notifications and syslog servers. You can also view the release number of the CCKM you are running from this page.



To display the context of which Key Management service you are currently logged into within CCKM, the icon associated with the Key Management service displays at the top of the left-hand navigation bar.

The arrow button at the bottom of the navigation bar allows you to expand the bar to view the names of the menu options and collapse it to view only the icons associated with the menu options. Click the arrow button to either expand or collapse the navigation bar.

On the top-right side of the CCKM portal for IBM CloudAWSAzureGoogle Cloud, the following options are available:

- **Azure Cloud Service Context Switcher:** This context switcher allows you to login into another Azure cloud service different from the current Azure cloud service into which you are currently logged. If you select the Azure cloud in which you are currently logged, the Multi-account Login/Logout window for the Azure account(s) displays. For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **AWS Login:** If you are not already logged into AWS, clicking on the AWS Login icon brings you to the AWS Password Login page where you can enter your AWS credentials to access the CCKM AWS cloud service (from the current cloud service in which you are currently logged). If you are already logged into AWS, clicking on the AWS Login icon brings you to the Multi-account Login/Logout window for the AWS account(s). For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **Google Cloud Service Context Switcher:** This context switcher allows you to login into another Google Cloud organization different from the current Google Cloud organization into which you are currently logged. If you select the Google Cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Google account(s) displays. For more information about the **Multi-account Login/Logout** window, see ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).

- **Salesforce Cloud Service Context Switcher:** This context switcher allows you to login into another Salesforce cloud service different from the current Salesforce cloud service into which you are currently logged. This is Salesforce, Salesforce Government, or Salesforce Sandbox. If you select the Salesforce cloud in which you are currently logged, the Multi-account Login/Logout window for the Salesforce account(s) displays. For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **Logout:** Logout from CCKM.

Home

The **Home** page provides a dashboard to display key operation and count metrics.

Keys

The **Keys** page allows Cloud administrators to view all the keys and manage them within the boundary of granted permissions. Those keys that do not have List permission for the respective Cloud administrator will not be listed.

A **Search** box is available at the top left of the **Keys** page allowing for a search of a specific key by entering any one of the following parameters:

- Key Name
- Version Name
- Key Purpose
- Project Name
- Protection Level
- Template algorithm
- Label
- Version state
- Version Algorithm
- Key Material Origin

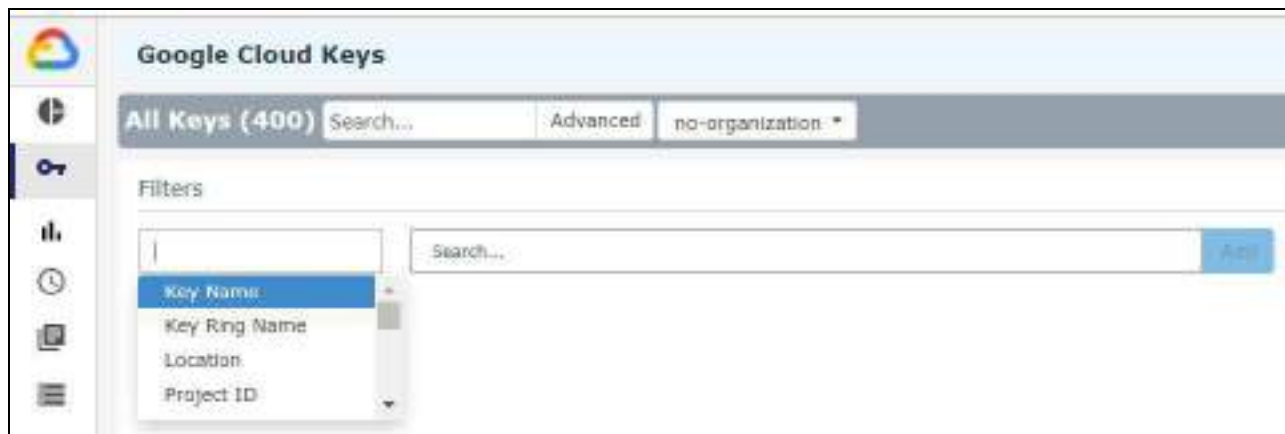
The **Advanced** tab allows you to filter your search on any or a combination of these columns displayed on the **Keys** page with the exception of the **Count** and **Action** columns:

- Key Name
- Key Ring Name
- Location
- Project ID
- Project Name
- Key Purpose
- Protection Level
- Template algorithm
- Key Material Origin
- Version State
- Version Algorithm

- Rotation Status
- Primary Version
- Label

Each of these columns are described in more detail below in this section.

To select to filter based on a column or combination of these columns from the **Keys** page, click **Advanced**. The **Filters** section displays. From the column filter drop-down menu, select the name of the column on which to filter your search. For example, select **Key Name**. In the **Search** box, enter the value on which to search (for example, **MK-createkey** as the key name) and then click **Add**. For the **Key Purpose**, **Template Protection Level**, **Template algorithm**, **Version State**, **Version Protection Level**, **Version Algorithm**, **Auto Rotate**, and **Primary Version** columns, a drop-down menu is available from which to select a value for the search. Click the down arrow, select a value from the available options, and then click **Add**. The specified search filter is then added to **Active Filters**. In the example, **Key Name: MK-createkey** is added to **Active Filters**. The results of the search displays in the **Keys** table. You can narrow your search by selecting more columns to add to the filter. To clear the filtered results, from **Active Filters**, click on the “X” (or the **Delete** button) for each of the named filters.



Available next to the **Advanced** tab is an organization drop-down menu with a list of organizations to which you have access through your granted permissions. From this menu, you can select a different organization context in which to switch. When you select an organization, the key rings that reside within the given organization display within the **Keys** page. If you have key rings that do not belong to any organization, then the option **organizations/no organization** is available from the drop-down menu. Selecting this option displays the key rings that do not belong to any organization within the **Keys** page.

At the top right of the **Keys** page, the following buttons are available:

- **Export Keys:** Allows you to export the contents of the Keys list into a report in a CSV format.
- **New Key:** Allows you to upload an external BYOK (CipherTrust Manager or DSM key) or create a native key. For information on how to upload an external BYOK or create a native key, see ["Create a Key" on page 157](#).
- **Synchronize:** Allows you to download the key rings and key metadata of all the keys that were created in Google Cloud KMS (to which you have access) into CCKM. If you have External Key Manager (EKM) keys saved in Google Cloud KMS, the key rings and key metadata of EKM keys are also downloaded to CCKM during key synchronization. For information on how to synchronize your keys, see ["Synchronize Keys" on page 167](#).

The **Show EKM** check box is also located at the top right of the **Keys** page. Selecting this check box allows you to display information about EKM keys (stored in Google Cloud KMS and downloaded to CCKM during key synchronization) on the CCKM **Keys** page. By default, this box is deselected, and EKM key information is not displayed on the **Keys** page.

Also located at the top right of the **Keys** page next to the **Synchronize** button is the **settings** (or gear) icon. Select this icon to filter which columns (with the exception of the **Actions** column) to display or hide on the **Keys** page. By default, all of the columns display with all of the check boxes for each of the available columns preselected. To hide a column, select the settings icon and then deselect the associated column check box. To display a column, ensure the associated check box is selected.

The following columns display on the **Keys** page:

Key ID: The ID of the key.

Key Ring: The ID of the key ring to which the key belongs.

Location: The name of the geographical region where your Cloud KMS key is stored and can be accessed.

Project ID: The ID of the project to which the key belongs.

Key Purpose: Defines a key's cryptographic capabilities and the algorithms that are supported for the key's version.

Protection Level: Indicates whether the key and key version are stored in Software or HSM.

Version: The version number of the key.

Version State: Indicates the status of the key version in Google Cloud KMS, which can be one of the following:

- **PENDING_GENERATION:** Key version is in the process of being generated. This state is only applicable to asymmetric keys. Once created and ready for use, the state of key version moves to the ENABLED state.
- **ENABLED:** Ready for use.
- **DISABLED:** Cannot be used. However, the key material is still available for use. You can set the state of the key version back to ENABLED to be able to use this key version again.
- **DESTROY_SCHEDULED:** Scheduled for destruction. This key version can be placed back into the DISABLED state.
- **DESTROYED:** Key version is destroyed and no longer available in the Google Cloud KMS. Included in this destruction is the key material. This key version cannot be restored.

Key Material Origin:

- If the key was uploaded from CCKM as a BYOK, then the value is INTERNAL(<source_key_name>).
- If the key was not uploaded from current CCKM as a BYOK, then the value is EXTERNAL.
- If the key material was created by Google Cloud KMS, then the value is KeyRing.

Rotation Status: Indicates whether the key is scheduled for autorotation in CCKM. AUTO is displayed in this column, if the key is scheduled for autorotation. This status *only* applies to scheduled autorotation in CCKM.

Actions: Allows you to select an action to apply to a given key or key version. To view the available actions for a key version, find the name of the key whose key version you wish to view and then expand the row to show the key versions underneath the key name. In the **Actions** column, click **Select**. The available actions are displayed. You may have to scroll through the list if you have more than two actions available for selection.

Actions available for Asymmetric keys:

- **Update Key**
- **Add Version**

Actions available for Asymmetric key versions

- **View all versions**
- **Enable**
- **Disable**
- **Schedule Destruction**
- **Schedule Destruction All Versions**
- **Cancel Destruction**
- **Cancel Destruction All Versions**
- **Download** (Public Key)

Actions available for Symmetric keys:

- **Update Key**
- **Rotate Key**

Actions available for Symmetric key versions

- **View all versions**
- **Enable**
- **Disable**
- **Schedule Destruction**
- **Schedule Destruction All Versions**
- **Cancel Destruction**
- **Cancel Destruction All Versions**

Create a Key

You have two options when you create a new key in Google Cloud KMS:

- ["Create a Native Key" on the next page](#)
- ["Upload your Own Key \(BYOK\)" on page 159](#)

Part of the process of creating a new key with either option is adding it to a key ring in a given Google Cloud location and also selecting the key's purpose. The purpose of a key defines its cryptographic capabilities and the algorithms that are supported for the key's versions. The following are the supported key purposes in Google Cloud KMS:

- **ENCRYPT_DECRYPT:** used for Symmetric encryption. This key purpose is referred to as **Symmetric encrypt/decrypt** in CCKM.
- **ASYMMETRIC_SIGN:** used for Asymmetric signing. This key purpose is referred to as **Asymmetric sign** in CCKM.
- **ASYMMETRIC_DECRYPT:** used for Asymmetric encryption. This key purpose is referred to as **Asymmetric decrypt** in CCKM.

Create a Native Key

Create a native key in a Google Cloud KMS. In the process of creating a native key in Google Cloud KMS, key material is generated by Google Cloud KMS along with key metadata from CCKM are used. After CCKM creates a native key, it displays it as **KeyRing** under the **Key Material Origin** column within the **Google Cloud Keys** page.

For more information, see the following sections:

- ["Create a Native Key \(Asymmetric Key\)" below](#)
- ["Create a Native Key \(Symmetric Key\)" on the facing page](#)

Create a Native Key (Asymmetric Key)

To create a native key that has a key purpose of **Asymmetric sign** (used for asymmetric signing) or **Asymmetric decrypt** (used for asymmetric encryption):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, select the **New Key** button and then **Create**. The **Create Native Key** dialog box displays.
3. From the **Key Purpose** box, select one of the following options depending on whether you will use the key for asymmetric signing or asymmetric encryption:
 - **Asymmetric sign**
 - **Asymmetric decrypt**
4. From the **Version template algorithm** box, select the algorithm to be used in the key material to create the new version of the key. If you selected **Asymmetric sign** as the key purpose, then the following are your options:
 - **2048 bit RSA - PSS Padding - SHA256 Digest**
 - **3072 bit RSA - PSS Padding - SHA256 Digest**
 - **4096 bit RSA - PSS Padding - SHA256 Digest**
 - **4096 bit RSA - PSS Padding - SHA512 Digest**
 - **2048 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
 - **3072 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
 - **4096 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
 - **4096 bit RSA - PKCS#1 v1.5 padding - SHA512 Digest**
 - **Elliptic Curve P-256 - SHA256 Digest**
 - **Elliptic Curve P-384 - SHA384 Digest**If no selection is made, then the default is **2048 bit RSA - PSS Padding - SHA256 Digest**.
If you selected **Asymmetric decrypt** as the key purpose, then the following are your options:
 - **2048 bit RSA - OAEP Padding - SHA256 Digest**
 - **3072 bit RSA - OAEP Padding - SHA256 Digest**
 - **4096 bit RSA - OAEP Padding - SHA256 Digest**
 - **4096 bit RSA - OAEP Padding - SHA512 Digest**If no selection is made, then the default is **2048 bit RSA - OAEP Padding - SHA256 Digest**.
5. From the **Key Ring** box, select the key ring to which to add the key.
6. In the **Key Name** box, enter the name of the key.

7. From the **Version Template Protection Level** area, select whether the key and key version will be stored in either **Software** or **HSM**. If no selection is made, then the default is **Software**.
8. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button. It is recommended you include "CCKM" in the name to reflect CCKM-associated information in the labels within Google Cloud list views and reports.
9. Click **Create**. The new key is displayed on the **Google Cloud Keys** page.

Create a Native Key (Symmetric Key)

To create a native key that has a key purpose of **Symmetric encrypt/decrypt** (used for symmetric encryption/decryption):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, select the **New Key** button and then **Create**. The **Create Native Key** dialog box displays.
3. From the **Key Purpose** box, select **Symmetric encrypt/decrypt**.

Note: The **Version template algorithm** box is not available for modification as selecting a version template algorithm is not required when creating a native key that has a key purpose of **Symmetric encrypt/decrypt**.

4. From the **Key Ring** box, select the key ring to which to add the key.
5. In the **Key Name** box, enter the name of the key
6. (Optional) From the **Version Template Protection Level** area, select whether the key and key version will be stored in either **Software** or **HSM**. If no selection is made, then the default is **Software**.
7. (Optional) In the **Rotation Period by Google** box, enter the number of days for the key rotation period. Google Cloud performs and manages this key rotation in Google cloud. The minimum is 1, and the maximum is 36,500 days.
8. (Optional) In the **Rotation Start Date by Google**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. This box displays only if you entered a rotation period in the previous step.
9. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button. It is recommended you include "CCKM" in the name to reflect CCKM-associated information in the labels within Google Cloud list views and reports.
10. Click **Create**. The new key is displayed on the **Google Cloud Keys** page.

Upload your Own Key (BYOK)

Upload the keys created from the **Key Sources** page of CCKM to the Google Cloud KMS. These "BYOK" keys are the keys you generated from the key source you are using (CipherTrust Manager or DSM). Both of these key sources support the algorithm types of AES 256, RSA 2048, RSA 3072, and RSA 4096. Additionally, CipherTrust Manager also supports the algorithm types of EC 256 and EC 384.

For more information, see the following sections:

- ["Upload your Own Key \(BYOK\) \(Asymmetric Key\)" on the next page](#)
- ["Upload your Own Key \(BYOK\) \(Symmetric Key\)" on page 161](#)

Upload your Own Key (BYOK) (Asymmetric Key)

Note

You must have the import permissions for the Google Cloud KMS to which you wish to upload the key.

To upload your BYOK to Google Cloud KMS that has a key purpose of **Asymmetric sign** (used for asymmetric signing) or **Asymmetric decrypt** (used for asymmetric encryption):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, select the **New Key** button and then **Upload**. The **Upload a Key** dialog box displays.
3. From the **Key Purpose** box, select one of the following options depending on whether you will use the key for asymmetric signing or asymmetric encryption:
 - **Asymmetric sign**
 - **Asymmetric decrypt**
4. (Optional) From the **Version template algorithm** box, select the algorithm to be used in the key material to create the new version of the key. If you selected Asymmetric sign as the key purpose, then the following are your options:
 - **2048 bit RSA - PSS Padding - SHA256 Digest**
 - **3072 bit RSA - PSS Padding - SHA256 Digest**
 - **4096 bit RSA - PSS Padding - SHA256 Digest**
 - **4096 bit RSA - PSS Padding - SHA512 Digest**
 - **2048 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
 - **3072 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
 - **4096 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
 - **4096 bit RSA - PKCS#1 v1.5 padding - SHA512 Digest**
 - **Elliptic Curve P-256 - SHA256 Digest**
 - **Elliptic Curve P-384 - SHA384 Digest**

If no selection is made, then the default is **2048 bit RSA - PSS Padding - SHA256 Digest**.

If you selected Asymmetric decrypt as the key purpose, then the following are your options:

- **2048 bit RSA - OAEP Padding - SHA256 Digest**
- **3072 bit RSA - OAEP Padding - SHA256 Digest**
- **4096 bit RSA - OAEP Padding - SHA256 Digest**
- **4096 bit RSA - OAEP Padding - SHA512 Digest**

If no selection is made, then the default is **2048 bit RSA - OAEP Padding - SHA256 Digest**.

5. From the **Key Provider** box, select CipherTrust Manager or DSM depending on the key source you are using for the key.
6. From the **Source Key** box, select one of the available source keys. The keys available in the drop-down menu are the keys generated from CipherTrust Manager or DSM depending on the key source you are using. Enter the first three characters of the name of the key to bring it up for selection.
7. From the **Key Ring** box, select the key ring to which the key is to be added.
8. In the **Key Name** box, enter the name of the key. If you wish to use the key name you entered when you generated this key from the **Key Sources** page, select the **Use source key name** check box. The name will display in the **Key Name** box.

9. (Optional) From the **Version Template Protection Level** box, select whether the key and key version will be stored in either **Software** or **HSM**. If no selection is made, the default is **Software**.
10. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button. It is recommended you include “CCKM” in the name to reflect CCKM-associated information in the labels within Google Cloud list views and reports.
11. Click **Upload to Google Cloud**. The new key is uploaded to Google Cloud KMS and is displayed on the **Google Cloud Keys** page.

Upload your Own Key (BYOK) (Symmetric Key)

Note

You must have the import permissions for the Google Cloud KMS to which you wish to upload the key.

To upload your BYOK to Google Cloud KMS that has a key purpose of **Symmetric encrypt/decrypt** (used for symmetric encrypt/decrypt):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, select the **New Key** button and then **Upload**. The **Upload a Key** dialog box displays.
3. From the **Key Purpose** box, select **Symmetric encrypt/decrypt**.
4. From the **Key Provider** box, select CipherTrust Manager or DSM depending on the key source you are using for the key.
5. From the **Source Key** box, select one of the available source keys. The keys available in the drop-down menu are the keys generated from CipherTrust Manager or DSM depending on the key source you are using. Enter the first three characters of the name of the key to bring it up for selection.
6. From the **Key Ring** box, select the key ring to which the key is to be added.
7. In the **Key Name** box, enter the name of the key. If you wish to use the key name you entered when you generated this key from the Key Sources page, select the **Use source key name** check box. The name will display in the **Key Name** box.
8. (Optional) From the **Version Template Protection Level** box, select whether the key and key version will be stored in either **Software** or **HSM**. If no selection is made, the default is **Software**.
9. (Optional) In the **Rotation Period by Google** box, enter the number of days for the key rotation period. Google Cloud performs and manages this key rotation in Google cloud. The minimum is 1, and the maximum is 36,500 days.
10. In the **Rotation Start Date by Google**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. This box displays only if you entered a rotation period in the previous step.
11. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button. It is recommended you include “CCKM” in the name to reflect CCKM-associated information in the labels within Google Cloud list views and reports.
12. Click **Upload to Google Cloud**. The new key is uploaded to Google Cloud KMS and is displayed on the **Google Cloud Keys** page.

Update Key

Use **Update Key** dialog box to update the current key version of an asymmetric or symmetric key to create a *new* version of it to use. The following applies to **Update Key**:

- If the key purpose of the key you wish to update is Symmetric encrypt/decrypt, then you have the option to update the key rotation period and select a new rotation time. You also have the option to change the primary version of the key.
- If the key purpose is either Asymmetric sign or Asymmetric decrypt, then you have the option to update the Version Template Algorithm of the current key version. Changing the version template algorithm does **not** affect all of the other existing key versions of the key. This update applies to the new version of the key.
- For both asymmetric or symmetric keys, you have the option to update the Labels information.

For more information, see the following sections:

- ["Update Key \(Asymmetric Key\)" below](#)
- ["Update Key \(Symmetric Key\)" on the facing page](#)

Update Key (Asymmetric Key)

Note

Keys cannot be updated when they are in a **DESTROYED** state.

To update a key that has a key purpose of **Asymmetric sign** (used for asymmetric signing) or **Asymmetric decrypt** (used for asymmetric encryption):

1. On the **Google Cloud Keys** page, click **Update Key** in the **Action** column for a particular key. The **Update key** dialog box displays. The **Project**, **Location**, **Key Ring**, and **Key Purpose** boxes and **Version Template Protection Level** area are prepopulated with the values associated with the key. These boxes and area cannot be updated.
2. From the **Version Template Algorithm** box, select the algorithm to be used in the key material of the version of the key. If you selected Asymmetric sign as the key purpose, then the following are your options:
 - **2048 bit RSA - PSS Padding - SHA256 Digest**
 - **3072 bit RSA - PSS Padding - SHA256 Digest**
 - **4096 bit RSA - PSS Padding - SHA256 Digest**
 - **4096 bit RSA - PSS Padding - SHA512 Digest**
 - **2048 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
 - **3072 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
 - **4096 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
 - **4096 bit RSA - PKCS#1 v1.5 padding - SHA512 Digest**
 - **Elliptic Curve P-256 - SHA256 Digest**
 - **Elliptic Curve P-384 - SHA384 Digest**

If you selected Asymmetric decrypt as the key purpose, then the following are your options:

- **2048 bit RSA - OAEP Padding - SHA256 Digest**
- **3072 bit RSA - OAEP Padding - SHA256 Digest**
- **4096 bit RSA - OAEP Padding - SHA256 Digest**

- **4096 bit RSA - OAEP Padding - SHA512 Digest**

3. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button.
4. Click **Update**.

Update Key (Symmetric Key)

Note

Keys cannot be updated when they are in a **DESTROYED** state.

To update a key with a key purpose of **Symmetric encrypt/decrypt** (used for symmetric encrypt/decrypt):

1. On the **Google Cloud Keys** page, click **Update Key** in the **Action** column for a particular key. The **Update key** dialog box displays. The **Project**, **Location**, **Key Ring**, **Key Purpose**, and **Version Template Algorithm** boxes and **Version Template Protection Level** area are prepopulated with the values associated with the key. These boxes and area cannot be updated.
2. (Optional) In the **Rotation Period by Google** box, enter the number of days for the key rotation period. Google Cloud performs and manages this key rotation in Google Cloud. The minimum is 1, and the maximum is 36,500 days.
3. (Optional) In the **Rotation Start Date by Google** box, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. This box displays only if you entered a rotation period in the previous step.
4. (Optional) To set a version of this key as the primary version, select the **Set as Primary Version** check box and then select the version number from the drop-down menu.
5. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button.
6. Click **Update**.

Rotate a Key

CCKM supports both scheduled rotation (or autorotation) and manual rotation of *symmetric* keys within Google cloud. When a key is rotated in CCKM for Google Cloud or in Google Cloud KMS, new key material is added to the key and the new version of the key is set as the primary key version. It is the primary key version of the key that Google Cloud KMS uses to encrypt data.

When you rotate a symmetric key, the following applies:

- Any data that was encrypted with previous versions of the key is not automatically re-encrypted.
- Existing key versions are not automatically disabled or destroyed.

Google Cloud KMS does **not** support autorotation and manual rotation of asymmetric keys. However, CCKM does provide this support partially. With either autorotation or manual rotation of asymmetric keys in CCKM, new key material is added to the key to create a new key version through the use of the **Add Version** action available on the **Google Cloud Keys** page. *However*, to begin using the new key version for either signing or encryption, you are required to take additional steps. Part of these additional steps includes distributing the public key of the new key version. CCKM provides support for downloading the public key of the new key version that you can distribute. For information about these additional steps, refer to [Considerations for Asymmetric Keys](#) within the online documentation for Google cloud. For more information about downloading a public key of a new key version of an asymmetric key, see "[Download Public Key \(Asymmetric Key Only\)](#)" on page 171.

Autorotation of a symmetric or an asymmetric key requires that this feature be configured in the **Schedules** page prior to enabling the feature from the **Google Cloud Keys** page. See ["Schedule Key Rotation" on page 174](#) for more information about scheduling key rotation.

For more information, see the following sections:

- ["Manually Rotate a Key \(Symmetric Key\)" below](#)
- ["Enable Autorotation of Key \(Symmetric Key\)" below](#)
- ["Manually Rotate a Key \(Add New Key Version\) \(Asymmetric Key\)" on the facing page](#)
- ["Enable Autorotation of Key \(Add New Key Version\) \(Asymmetric Key\)" on page 166](#)

Manually Rotate a Key (Symmetric Key)

Manually rotating a symmetric key allows CCKM to add new key material to the selected key and set the new key version as the primary key version to be used for encrypting data.

After a new key version is generated, it will be added to the list of key versions as a primary version on the **Google Cloud Keys** page.

Note

During the rotation of a symmetric key, the template protection level that you initially configured for the key is used to generate the new key version.

To manually rotate a symmetric key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key to manually rotate.
3. Select **Rotate Key** from the **Actions** drop-down menu. The **Rotate Google Cloud Key** dialog box is displayed. The **Version Algorithm** box is prepopulated with the algorithm of Symmetric encrypt/decrypt.
4. From **Key Provider**, select the key provider (also known as key source) from which to generate the new key material for the key. Two options of the three displayed are available for selection (the current key source you are currently using and Native):
 - **DSM**—Key is to be generated from DSM.
 - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager.
 - **Native**—Key is to be generated natively in the designated Google Cloud KMS using the CCKM UI. Note that this option is equivalent to generating a key in the Google Cloud portal.
5. From the **Source Key** box, select the source key to use for the new key version. This box displays only if you selected either **DSM** or **CipherTrust Manager** in the previous step.
6. Click **Rotate**.

Enable Autorotation of Key (Symmetric Key)

Enabling autorotation of a symmetric key allows CCKM, at a scheduled time, to add new key material to the selected key and set the new key version as the primary key version to be used for encrypting data.

After a new key version is generated, it will be added to the list of key versions as a primary version on the **Google Cloud Keys** page.

Note

During the rotation of a symmetric key, the template protection level that you initially configured for the key is used to generate the new key version.

To enable autorotation of a symmetric key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key to autorotate.
3. Select **Rotate Key** from the **Actions** drop-down menu. The **Rotate Google Cloud Key** dialog box is displayed.
4. Select the **Auto Rotate** tab. The **Auto Rotate** dialog box is displayed. The **Version Algorithm** box is prepopulated with the algorithm of Symmetric encrypt/decrypt.
5. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.
6. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
7. From **Key Provider**, select the key provider (also known as key source) from which to generate the new key material for the key. Two options of the three displayed are available for selection (the current key source you are currently using and Native):
 - **DSM**—Key is to be generated from DSM.
 - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager.
 - **Native**—Key is to be generated natively in the designated Google Cloud KMS using the CCKM UI. Note that this option is equivalent to generating a key in the Google Cloud portal.
8. Click **Submit**.

Manually Rotate a Key (Add New Key Version) (Asymmetric Key)

Manually rotating an asymmetric key allows CCKM to add new key material to the selected key to create a new key version. You are required to take additional steps to begin using the new key version for either signing or encryption.

After a new key version is generated, it will be added to the list of key versions on the **Google Cloud Keys** page.

Note

During the rotation of an asymmetric key, the template protection level that you initially configured for the key is used to generate the new key version.

To add a new version of an asymmetric key to the key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the asymmetric key to which to add a new version.
3. Select **Add Version** from the **Actions** drop-down menu. The **Add Version for Google Cloud Key** dialog box is displayed.

4. From **Key Provider**, select the key provider (also known as key source) from which to generate the new key material for the new key version. Two options of the three displayed are available for selection (the current key source you are currently using and Native):
 - **DSM**—Key is to be generated from DSM.
 - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager.
 - **Native**—Key is to be generated natively in the designated Google Cloud KMS using the CCKM UI. Note that this option is equivalent to generating a key in the Google Cloud portal.
5. (Optional) From the **Version Algorithm** box, select the algorithm to use for the new key version. This box is prepopulated with the algorithm from the template version algorithm initially configured for the key. If you selected either **DSM** or **CipherTrust Manager** as the key provider, you have the option of changing the version algorithm. If you selected **Native**, you do not have the option to change the version algorithm.
6. From the **Source Key** box, select the source key to use for this new key version. This box displays only if you selected either **DSM** or **CipherTrust Manager** as the key provider.
7. Click **Add Version**.

Enable Autorotation of Key (Add New Key Version) (Asymmetric Key)

Enabling autorotation of an asymmetric key allows CCKM, at a scheduled time, to add new key material to the selected key to create a new key version. You are required to take additional steps to begin using the new key version for either signing or encryption.

After a new key version is generated, it will be added to the list of key versions on the **Google Cloud Keys** page.

Note

During the rotation of an asymmetric key, the template protection level that you initially configured for the key is used to generate the new key version.

To enable autorotation of adding a new version of an asymmetric key to the key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the antisymmetric key to which to add a new version.
3. Select **Add Version** from the **Actions** drop-down menu. The **Add Version for Google Cloud Key** dialog box is displayed.
4. Select **Auto Rotate** tab. The **Auto Rotate** dialog box displays.
5. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.
6. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
7. From **Key Provider**, select the key provider (also known as key source) from which to generate the new key material for the new key version. Two options of the three displayed are available for selection (the current key source you are currently using and Native):
 - **DSM**—Key is to be generated from DSM.
 - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager.
 - **Native**—Key is to be generated natively in the designated Google Cloud KMS using the CCKM UI. Note that this option is equivalent to generating a key in the Google Cloud portal.

8. (Optional) From the **Version Algorithm** box, select the algorithm to use for the new key version. This box is prepopulated with the algorithm from the template version algorithm initially configured for the key. If you selected either **DSM** or **CipherTrust Manager** as the key provider, you have the option of changing the version algorithm. If you selected **Native**, you do not have the option to change the version algorithm.
9. Click **Submit**.

View All Key Versions

From the **Keys** page, you can access the **Key Versions** dialog box to view all the versions of a key and its associated information. At the top half of this dialog box, the following information about the selected key version is displayed: **Key**, **Organization**, **Purpose**, **Key Labels**, **Template Protection Level**, **Template Algorithm**, **Primary version**, **Creation Date**, **Rotation Period by Google**, and **Next Rotation Time by Google**. Note that the **Rotation Period by Google** and **Next Rotation Time by Google** refers to the key rotation that Google Cloud KMS manages and performs. At the bottom half of this dialog box, a list of all the key versions with the following columns is displayed: **Version**, **State**, **Algorithm**, **Created On**, **Key Material Origin**, and **Actions**. From the **Actions** column, you can select an action available to key versions to apply to a selected key version. See ["Keys" on page 154](#) for more information about the available options for **Actions**. You can display the next 5, 10, 15, 20, 25, 50, or 100 entries at the bottom of the **Key Versions** page using the **Show entries** drop down menu. You can view each available page by scrolling through the pages by clicking the less than "<" or greater than ">" symbols or by clicking on a given page number.

To view all of the key versions of a key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key versions you wish to view.
3. Click the key name to expand the row to show the key versions underneath the key name.
4. Click **View all versions**. The **Key Versions** dialog box displays. You can scroll through list to view all of the key versions.
5. Click the **Close** button to close the **Key Versions** dialog box.

Synchronize Keys

When you initially log into the CCKM User Portal, CCKM retrieves all of the projects and project locations that meet the following criteria and saves this information to MongoDB:

- active projects
- where Google Cloud KMS is enabled
- billing is enabled
- belong to the user's current organization

CCKM *also* performs the *first* synchronization with Google Cloud KMS. During this synchronization, the key metadata of all of your Google Cloud keys from your key rings within the downloaded projects are also downloaded to CCKM and saved to MongoDB. Essentially, it is the key metadata for each of the keys that is downloaded during this first key synchronization. All of the downloaded information is available to you the next time you log back into CCKM as this information is saved to MongoDB.

If you wish to manually synchronize your keys from your projects (all or selected) and key rings from Google Cloud KMS with CCKM, you can do so using the **Synchronize** button in the **Google Cloud Keys** page. Included in the synchronizing of the keys are EKM keys (if these exists within your projects). Synchronizing your key rings is optional. By default, the **Also synchronize Key Rings** check box within the **Synchronize Keys** dialog box is

deselected. When this box is deselected, synchronization does not include downloading your key rings. In this case, only key metadata of the keys from the key rings that were *previously* downloaded to CCKM during the last synchronization will be downloaded. When this box is selected, then all of your key rings are downloaded including new ones that may have been added to your projects within Google Cloud KMS since the last synchronization of key rings took place. In addition, key metadata of all of your keys are also downloaded as part of the synchronization.

Note

If you do not select the **Also synchronize Key Rings** check box, the permissions on the saved key rings are retrieved during key synchronization even though your key rings are not downloaded.

You have the option to schedule key synchronization, which you perform in the **Schedules** page. See ["Schedule Key Synchronization" on page 175](#) for more information.

To synchronize your keys and (optionally) key rings from selected projects:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **Google Cloud Keys** page, click **Synchronize**. The **Synchronize Key** dialog box displays.
3. Under **Select projects to synchronize**, select the **Select All** check box to select all of your projects. To deselect a project, click the **X** next to the project name. You can also click in the **Select projects to synchronize** box and bring up the list of projects from which you can select a project.
4. (Optional) If you wish to also synchronize your key rings, select the **Also synchronize Key Rings** check box.
5. Click **Synchronize**.

Disable or Enable Key Version

You have the option to disable a key version in CCKM. Any key version that is disabled has a version state of **DISABLED** as displayed in the **Version State** column on the **Google Cloud Keys** page. After a key version is disabled, any data that was encrypted using the given key version cannot be accessed. Essentially, the key version can no longer be used for cryptographic operations. However, the key material of the disabled key version is still available for use. You can re-enable a key version and set its version state back to **ENABLED** so that you can use this key version again.

For more information, see the following sections:

- ["Disable Key Version" below](#)
- ["Enable Key Version" on the facing page](#)

Disable Key Version

To disable a key version:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key version you wish to disable.
3. Click the key name to expand the row to show the key versions underneath the key name.
4. Click **View all versions**. The **Key Versions** dialog box displays. Scroll through the list to view the information for each version.
5. Find the key version you wish to disable.
6. Select **Disable** from the **Actions** drop-down menu.

Enable Key Version

To re-enable a key version that is in a DISABLED version state:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key version you wish to enable.
3. Click the key name to expand the row to show the key versions underneath the key name.
4. Click **View all versions**. The **Key Versions** dialog box displays. Scroll through the list to view the information for each version.
5. Find the key version you wish to enable
6. Select **Enable** from the **Actions** drop-down menu.

Schedule (or Cancel) Destruction of Key Version or All Key Versions

CCKM supports the scheduling of a key version for destruction in Google Cloud KMS. A key version that is scheduled for destruction has a version state of **Scheduled for destruction** as displayed in the **Version State** column on the **Google Cloud Keys** page. Destroying a key version equates to destroying the version's key material. After destruction, the key version and the key material cannot be recovered. Google Cloud performs the key version destruction after 24 hours has elapsed. After a key version is destroyed, any data that was encrypted using the given key version cannot be accessed.

If you wish to schedule the destruction of *all* key versions of a key (for example, all 50 key versions of a key) in Google Cloud KMS at once, CCKM also supports this capability. This capability allows you to bypass having to schedule the destruction of each of the key versions of a key saving you time.

Note that you can cancel the destruction of a key version or all key versions in CCKM before the 24 hours has elapsed to restore the key material for the given key version. Google Cloud KMS does not perform the destruction after this cancellation is made in CCKM. After the cancellation, the key version state changes from DESTROY_SCHEDULED to DISABLED. In this version state, you can enable the key. And from the ENABLED version state, the key version is available for use for cryptographic operations once again.

Note

The Audit logs within the **Logs** page of the CCKM user portal provides the status for the actions of schedule destruction of a key version or all key versions and cancel destruction of a key version or all key versions. In the event that any one of these actions fail, the event name of **Schedule_Destruction_GCP_Key_Material** or **Cancel_Destruction_GCP_Key_Material** displays under the **Event Name** column along with the severity level under the **Severity** column for each impacted version. The actual event name that displays depends on which action failed. If the failed action relates to schedule destruction of all key versions, then an additional log message is provided to summarize the total number of key versions that were successfully scheduled for destruction. For those key versions that were not successfully scheduled for destruction, an additional log message is provided listing each of the version numbers that were not scheduled for destruction. If the failed action relates to cancel schedule destruction of all key versions, then an additional log message is provided to summarize the total number of key versions that were restored and the total number that were not restored.

Note

If you schedule the destruction of a key version, be sure to synchronize your keys with the Google Cloud KMS from the CCKM **Google Cloud Keys** page after the 24 hours has elapsed. Otherwise, your CCKM keys will be out of sync with Google Cloud KMS keys, and the option to cancel the destruction of a key version will still display as being available in CCKM. If you attempt to cancel the key version destruction after the 24 hours has elapsed, the operation will fail because the key material of the key version has already been destroyed within Google Cloud KMS.

For more information, see the following sections:

- [Schedule Destroy Key Version](#)
- [Cancel Schedule Destroy Key Version](#)
- ["Schedule Destruction of All Key Versions" below](#)
- ["Cancel Schedule Destruction of All Key Versions" on the facing page](#)

Schedule Destruction of Key Version

To schedule the destruction of a key version for a given key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key version you wish to schedule for destruction and then expand the row to show the key versions underneath the key name.
3. To see all key versions underneath the key name, click **View all versions**. The **Key Versions** dialog box displays. Scroll through the list to find the key version you wish to schedule for destruction.
4. Select **Schedule Destruction** from the **Actions** drop-down menu for the selected key version. The **Schedule Destruction of Key Material** dialog box is displayed. A warning displays indicating that this action will schedule key material destruction for "<path to key version>" from your Google Cloud key ring. The message also indicates to type the confirmation phrase (I wish to destroy the key material, and I understand it cannot be restored.) in the text box if you wish proceed with scheduled destruction of the key version for the given key.
5. Enter or copy/paste the confirmation phrase in the text box.
6. Click **Destroy**.

Schedule Destruction of All Key Versions

To schedule the destruction of all key versions for a given key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key versions you wish to schedule for destruction.
3. Select **Schedule Destruction All Versions** from the **Actions** drop-down menu for the selected key. The **Schedule Destruction of All Key Version Materials** dialog box is displayed. A warning displays indicating that this action will schedule all key version material destruction for "<path to key>" from your Google Cloud key ring. The message also indicates to type the confirmation phrase (I wish to destroy the key material for all the versions, and I understand they cannot be restored.) in the text box if you wish to proceed with scheduled destruction of all key versions for the given key.
4. Enter or copy/paste the confirmation phrase in the text box.
5. Click **Destroy**.

Cancel Schedule Destruction of Key Version

To cancel the scheduled destruction of a key version for a given key and restore it:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key version you wish to cancel the scheduled destruction and then expand the row to show the key versions underneath the key name.
3. Find the key version for which you wish to cancel the scheduled destruction.
4. Select **Cancel Destruction** from the **Actions** drop-down menu. The **Cancel Scheduled Destruction of Key Material** dialog box is displayed. A warning displays indicating that this action will cancel the scheduled destruction of key version "<path to key version>" in your Google Cloud key ring.
5. Click **Restore**.

Cancel Schedule Destruction of All Key Versions

To cancel the scheduled destruction of all key versions of a given key and restore these versions:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key versions you wish to cancel the scheduled destruction.
3. Select **Cancel Destruction All Versions** from the **Actions** drop-down menu. The **Cancel Scheduled Destruction of All Key Material** dialog box is displayed. A warning displays indicating that this action will cancel the scheduled destruction of all key versions for "<path to key>" in your Google Cloud key ring.
4. Click **Restore**.

Download Public Key (Asymmetric Key Only)

Before you can begin using a new key version of an asymmetric key after key rotation, you are required to take additional steps. Part of these additional steps include distributing the public key of the new key version. CCKM provides support for downloading a CCKM public key of the new key version.

To download the CCKM public key of a new asymmetric key version:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key version you wish to download the public key and then expand the row to show the key versions underneath the key name.
3. Find the key version for which to download the public key.
4. Select **Download** from the **Actions** drop-down menu. The public key file is downloaded to your laptop or desktop. When prompted as to whether to keep or discard the file type, select **Keep**.

Export Keys

Export Keys allows you to export the contents of the **Google Cloud Keys** page into a report in a CSV format.

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, click **Export Keys**. The report file is downloaded to your laptop or desktop.

Reports

CCKM includes the following reports:

- **Combined Key Activity Reconciliation Report**

Provides the history of a key for a selected period using the following information (in column headings):

- Key name
- Organization
- Project
- Key Ring
- Key Activity
- Cloud Key Manager Key Activity
- Timestamp
- Cloud Key Manager Timestamp

- **Google Cloud Key Activity Report**

Illustrates the activity type performed by a specific user and at the stated time. Information provided in the column headings of this report is the following:

- Key name
- Organization
- Project
- Key Ring
- Key Activity
- Region
- Origin
- Modified By
- Modified Time

- **Google Cloud Key Aging Report**

Provides information about key expiration using the following information (in column headings):

- Key Name
- Organization
- Project
- Key Ring
- Region
- Key Rotation
- Schedule Key Deletion Date

- **Cloud Key Manager User Action Report**

Lists the users and their actions including creating, uploading, rotating, and deleting keys in the CCKM portal. Information provided in the column headings of this report is the following:

- User Name
- Delete Key
- Upload Key
- Schedule Delete Key Material
- Cancel Delete
- Rotate Key

Generate a Report

1. In the CCKM portal, in the left navigation bar, select **Reports**.
2. Under **Report Name**, select the report you wish to run.
3. From **Date Range**, click **Last day** to run the report for the activities from the previous day or **Specific** to run the report using a date range. If you choose to run by a date range, enter the start and end dates in the format MM/DD/YY or select the dates from the calendar button at the right of the box.

If you are running the **Google Cloud Key Aging Report**, click **Next week** to run the report for the activities for the next week or **Specific** to run the report using a date range.

4. Click **Run Report**. A link to the report is made available under the **Last Run** column. To view the report, click on the link and the report displays.
5. To download the report, click **Download** at the top right of the page and select the report format (PDF or CSV) to download.

Delete a Report

1. From the **All Reports** page, select the specific report you wish to delete. The report displays.
2. From the **Actions** column, click **Delete**. The **Delete Current Report** dialog box with a warning that you are about to delete a current report run on a specified date and time.
3. Click **Delete**. The report is deleted from CCKM.

Schedules

The **Schedules** page allows Cloud administrators to view and schedule the following job types:

- Key Rotation
- Key Synchronization

CCKM provides support for multiple schedules for each of the job types.

At the top right of the **Schedules** page, the **Add Schedule** button is available from which to add a scheduled job type.

Schedule Key Rotation

CCKM provides support for multiple schedules of the rotation (autorotation) of symmetric keys within Google cloud. During a scheduled key rotation, new key material is added to the key and the new version of the key is set as the primary key.

When you rotate a symmetric key, the following applies:

- Any data that was encrypted with previous versions of the key is not automatically re-encrypted.
- Existing key versions are not automatically disabled or destroyed.

Google Cloud KMS does *not* support automatic rotation of asymmetric keys. However, CCKM does support automatic rotation of these keys by adding new key material to the key to create a new version. To begin using the new key version for either signing or encryption, you are required to take additional steps. For information about these additional steps, refer to [Considerations for Asymmetric Keys](#).

Scheduled key rotation (for both symmetric and asymmetric keys) requires that this feature be configured in the **Schedules** page prior to enabling the feature from the **Keys** page. For more information about enabling scheduled key rotation of symmetric and asymmetric keys, refer to ["Rotate a Key" on page 163](#).

You can pause or delete a scheduled key rotation in the **Schedules** page.

Set up a Key Rotation Schedule

To set up a key rotation schedule:

Note

The access permissions of an Admin role (roles/cloudkms.admin) within Google Cloud KMS are required to set up a key rotation schedule.

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays. Your username associated with your Google Cloud account is automatically populated in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_ROTATION**.
4. From **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Click **Save**.

Pause a Key Rotation Schedule

To pause a key rotation schedule, select **Yes** for **Pause** within the **Add Schedule** dialog box and then click **Save**. The setting is saved. To resume the rotation schedule after pausing it, select **No** for **Pause** and then click **Save**. The setting is saved.

Delete a Key Rotation Schedule

To delete a key rotation schedule, select **Delete** from the **Actions** drop-down menu from the **Schedules** page.

Schedule Key Synchronization

CCKM allows for multiple schedules of the synchronization of Google Cloud keys. Synchronizing downloads the key metadata of any keys that were created in Google Cloud KMS, outside of the CCKM portal, into your CCKM. Scheduled key synchronization requires that this feature be configured in the **Schedules** page. You can also pause or delete a scheduled key synchronization in the **Schedules** page.

Set up a Key Synchronization Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays. Your username associated with your Google Cloud account is automatically populated in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_SYNCHRONIZATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the key synchronization by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key synchronization by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Click **Save**.

Logs

CCKM maintains logs for events, such as login, key generation, synchronizing of keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. For each logged event listed on the **Logs** page, the following information is displayed:

- Event name
- Severity level
- Date and time of the event
- Event message
- User (name) associated with the event

You can search for a specific logged event from the **Logs** page using the **Search** box. Allowable filters on the search are event name, severity level, event message, and user (name).

Key Sources

The **Key Sources** page displays **all** of the keys that were generated from the key source you are using. If you are logged into multiple clouds within CCKM, this page displays all of the keys from these clouds.

Three tabs are available at the top of the **Key Sources** page:

- **CipherTrust Manager Key:** Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.
- **DSM Key:** Use this tab to use DSM as your key source. The keys are generated and stored within DSM.
- **nShield Key:** This tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Google Cloud.

Note

CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

Keys listed on the **Key Sources** page are listed by name, cloud ID, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description and actions you can take regarding that key.

A **Search** box is available at the top left of the **Keys Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Cloud ID
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

Add (Generate) a Key

From the **Key Sources** page, you can add (or generate) keys in CCKM using CipherTrust Manager or DSM as a key source for *all* of the clouds into which you are currently logged.

Note

The key algorithms currently supported for a Google Cloud key in CCKM are AES and RSA.

Add (Generate) a CipherTrust Manager Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **CipherTrust Manager Key** tab.
3. Select **Add Key > GoogleCloud Key**.
4. In the **Add CipherTrust Manager Key** dialog box, select the name of the user of the key from the **User** drop-down menu. Service is fixed to **GoogleCloud**.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.

7. From the **Algorithm** box, select the algorithm (**AES256**, **RSA2048**, **RSA3072**, **RSA4096**, **ECP256**, or **ECP384**) to apply to the key.
8. Click **Save** to save the new key to CCKM. The new key is added to the keys list. (Reset clears the screen before it has been saved.)

Add (Generate) a DSM Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key** tab.
3. Select **Add Key > GoogleCloud Key**.
4. In the **Add DSM Key** dialog box, select the name of the user of the key from the **User** drop-down menu.
5. In the **Source Key Tier**, select one of the following as a key source of the key:
 - **DSM**—Generate a key from DSM and import it into a Google cloud.
6. In the **Name** box, enter the name of the new key.
7. Enter a description of the key in the **Description** box.
8. From the **Algorithm** box, select the algorithm (**AES256**, **RSA2048**, **RSA3072**, or **RSA4096**) to apply to the key.
9. Click **Save** to save the new key to CCKM. The new key is added to the keys list. (Reset clears the screen before it has been saved.)

Add (Generate) an nShield Key

The **nShield Key** tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Google cloud.

Delete a Key

Deleting a key from the **Key Sources** page deletes the key from CCKM. When deleting a key from CCKM, the key is deleted from the key source you used to generate the key.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **CipherTrust Manager** or **DSM Key** tab depending on the key source you used to generate this key initially.
3. Click **Delete** next to the target key.
4. The **Delete Key** dialog box with a Warning display asking you to confirm the key deletion by entering the supplied phrase.
5. Enter or copy/paste the confirmation phrase.
6. Click **Delete**.

Manage Keys in Key Sources List Page

In addition to adding and deleting keys, the **Key Sources** page allows you to sort the existing keys by Name, Cloud ID, Key Type, Algorithm, and Created Date.

Settings

To access the **Settings** page in CCKM, select **Settings** from the left-hand navigation bar. From the **Settings** page, you can:

- Enter the email addresses of the recipients who are to receive the email notifications.
- Enable email notifications to be sent out from CCKM when specific types of key actions are performed. For example, when a key is uploaded to the cloud service. You can also enable CCKM to send reminder notifications for when to rotate a key.
- Configure remote syslog servers to which to send syslog messages.
- Configure Service account to use for authentication to CCKM API.
- View information about the software release number of the CCKM you are running including the build and version number.

Adding Recipients' Email Addresses (for Notifications)

Use the **General** tab from the **Settings** page to enter the email addresses of the recipients who are to receive email notifications from CCKM. Enter the recipients' email addresses separated by a comma in the **Notification Recipient** box and then click **Save** to save setting.

Enabling Notifications

Using the **Keys Notification** tab from the **Settings** page, you can configure CCKM to send an email notification when a key is deleted from Google Cloud or restored or uploaded to Google Cloud. You can also use this tab to configure CCKM to send a reminder notification for when to manually rotate a key along with the frequency (in days) at which a notification is sent. Add the email address(es) of the recipients in the **General** tab from the **Settings** page. By default, no notifications are sent.

To enable email notifications:

1. In the **Settings** page, select the **Keys Notification** tab.
2. Under **Notifications**, select **On** for each of the notifications you wish to enable:
 - Rotate Google Cloud Key (reminder notification)
 - Delete Google Cloud Key
 - Restore Google Cloud Key
 - Upload Google Cloud Key
3. If you selected to enable **Rotate Google Cloud Key**, in the **Days** box, enter the number of days for the frequency a reminder notification is sent.
4. Click **Save**.

Configuring Syslog Server

Note

In this document, the terms "syslog server" and "log server" are synonymous.

The **Syslog** tab from the **Settings** page allows you to configure the remote syslog or log server to which to send CCKM log data using TCP, TLS, or UDP protocol as the transport protocol. If you use the TLS protocol, then you are required to upload the syslog server certificate to CCKM for the TLS connection.

The **Syslog** tab also allows you to delete your syslog server configuration so that log data are no longer sent to the server.

Prior to configuring the setting for a remote syslog server, ensure that you have installed and configured the syslog server including configuring the setting for the port number the transport protocol will use to connect to CCKM. This port number configuration must match the port configuration you set in the **Syslog** tab of the **Settings** page.

Note

Support for both Syslog and Splunk as a remote syslog server is provided in CCKM. However, the use of a Splunk cloud server as a remote syslog server is currently not supported in CCKM.

Configure Remote Syslog Server

To configure a remote syslog server to which to send CCKM log data:

Note

The facility name of "LOCAL1" for the syslog server displays in the **Facility** box. This default value is not available for modification.

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. In the **Hostname** box, enter the hostname or IP address of the syslog server.
3. In the **Port** box, enter the port number the transport protocol uses to connect to the syslog server. Enter a value between 1 and 65535. The default value is 514.
4. In the **Protocol** drop-down menu, select **TCP**, **TLS**, or **UDP** for the transport protocol to use when sending the log data to the syslog server. The default value is **UDP**. If you select **TCP**, proceed to Step 6. If you select **TLS**, proceed to the next step. Otherwise, proceed to Step 7.
5. In the **Certificate** box, select **Choose File** and then select the syslog server certificate to upload to CCKM.
6. Click **Test Connection** to test the connection between CCKM and the remote syslog server. This button is only available for use when either **TCP** or **TLS** is the selected transport protocol.
7. In the **Log Format** drop-down menu, select **CEF** (Common Event Format), **Plain Text**, or **RFC5424** for the format of the log message to use when sending the log data. The default value is **RFC5424**.
8. (Optional) In the **Description** box, enter a description of the remote syslog server.
9. Click **Save**.

Delete Remote Syslog Server Configuration

To delete the configuration of a remote syslog server:

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. Click **Delete**. The **Delete Syslog** dialog box displays.
A warning message displays indicating that this action will delete the syslog settings from CCKM. The message also indicates to type the confirmation phrase (I wish to delete the syslog settings) in the text box if you still wish to delete the syslog settings.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Delete**.

Adding Google Cloud Service Account to CCKM

Use the **Service Account** tab from the **Settings** page to add your Google Cloud service account to CCKM and create a password for it. Your service account must already be created in your Google Cloud Platform console before you proceed with adding your service account to CCKM.

After submitting the request to add your Google service account to CCKM in the **Add Service Account** dialog box, CCKM will call the Google Cloud API to add a key pair to the service account. After the key pair is added, CCKM downloads the private key and saves it in the database as an encrypted key. CCKM also saves your password hash in the database.

Later, you can use the service account email and password to authenticate to CCKM REST API.

The following columns display on the **Service Account** tab:

- Service Account Email
- Key ID
- Key Creation Date
- Key Expiration Date
- Actions

Add Service Account

To add a Google service account to CCKM:

1. In the **Settings** page, select the **Service Account** tab. The **Service Account** dialog box displays.
2. Click **Add Service Account**. The **Add Service Account** dialog box displays.
3. In the **Email** box, enter the email of your Google service account.
4. In the **Password** box, enter your password for the service account. The password must be at least 8 characters long and contain one upper case letter, one lower case letter, one digit, and one special character from a limited set of special characters "!@#\$\$%^&*".
5. In the **Confirm Password** box, enter your password for the service account again to confirm the password.
6. Click **Submit**.

Update Service Account

You have the option to change the service account password and change the key pair for service account from the **Service Account** tab. To change the key pair, select to rotate the key. When Google Cloud rotates the key pair, the current key pair will be deleted, and a new key pair will be created. CCKM then encrypts the new private key and your password hash and saves this data in the CCKM database.

To update the information for Google Cloud service account:

1. In the **Settings** page, select the **Service Account** tab. The **Service Account** dialog box displays.
2. Click **Add Service Account**. The **Add Service Account** dialog box displays.
3. In the **Email** box, enter the email of your Google Cloudservice account.
4. In the **Password** box, enter your password for the service account. The password must be at least 8 characters long and contain one upper case letter, one lower case letter, one digit, and one special character from a limited set of special characters "!@#\$%^&*".
5. In the **Confirm Password** box, enter your password for the service account again to confirm the password.
6. Click **Submit**.

Delete Service Account

To delete a Google Cloud service account:

1. In the **Settings** page, select the **Service Account** tab. The **Service Account** dialog box displays.
2. Select **Delete** from the **Actions** drop-down menu. The **Confirm Deletion of Service Account** dialog box is displayed.
3. Click **Confirm**.

About

Use the **About** tab to view the build number, version number, patch number (if applicable), and build number of the CCKM you are running.

Chapter 16: CCKM for IBM Key Management

Overview	182
Types of Encryption Keys Supported within IBM HPCS and Key Protect	182
IBM Cloud Identity and Access Management	183
Action Permissions Mapping for CCKM and IBM HPCS/Key Protect	184
CCKM's Interaction with IBM HPCS and Key Protect	185
Prerequisites on IBM HPCS and Key Protect	185
Creating an IBM Credentials File in a JSON Format	190
Access CCKM for IBM	190
IBM CCKM User Portal	190
Home	191
Keys	191
Reports	205
Schedules	207
Logs	210
Key Sources	210
Settings	212

Overview

CCKM provides support for the following IBM Cloud key management services:

- IBM Cloud Hyper Protect Crypto Services (IBM HPCS)
- IBM Key Protect

Note: This documentation assumes that you are familiar with using IBM HPCS, IBM Key Protect, and IBM Cloud Identity and Access Management (IAM). For details about these IBM technologies, refer to the IBM Cloud documentation at <https://cloud.ibm.com/docs>.

Types of Encryption Keys Supported within IBM HPCS and Key Protect

The following encryption keys are employed and managed within IBM HPCS and Key Protect that you can also manage from CCKM:

- Standard key—This is a symmetric key used to directly encrypt and decrypt secrets, such as passwords or encryption keys.
- Root key—This is a symmetric key-wrapping key used to protect other cryptographic keys that use envelope encryption.

Both an IBM native key and a BYOK (non-native) key can either be a standard or a root key.

IBM Cloud Identity and Access Management

The IBM Cloud key management services of IBM HPCS and Key Protect for IBM Cloud provide a centralized access control system using IBM IAM to assist you in managing users, service IDs, and access for your encryption keys within your organization. Both of these key management services employ IAM roles in the management of your keys. These roles have different access permissions for the key management service. For detailed information about IBM Cloud IAM, refer to topics covering “IBM Cloud Identity and Access Management” within the IBM Cloud documentation.

As an account administrator in IBM Cloud, the actions you can perform include the following:

- assign and manage access for users
- create resource groups
- create access groups
- create service instances

To provide access for users or access groups, or users and access groups, or service IDs in IBM HPCS and Key Protect, you can create policies that define the resource(s) the users or access groups, or users and access groups, or service IDs are permitted to access and a role or roles that define the type of access permissions (actions allowed) on the given resources.

The following are the two types of roles supported within IBM IAM:

- Platform management role—used to grant permissions at the account level, such as creating or deleting instances in your IBM Cloud account. To access your instances for HPCS or Key Protect through CCKM, ensure to set the platform management role of Viewer (at the least). The supported roles are Administrator, Operator, Editor, and Viewer.
- Service access role—used to define the actions a user or service can perform on an IBM HPCS or Key Protect instance, such as performing API calls. The following table provides the supported service access roles along with the IBM HPCS or Key Protect access permissions allowed for each:

Service Access Role	Access Permissions (Allowed Actions) in IBM Cloud	Notes
Reader	• View keys • Wrap keys • Unwrap keys	CCKM does not support wrap and unwrap keys.
Writer	• Create keys • List keys • Retrieve keys • Rotate keys • Wrap keys • Unwrap keys • Set keys for deletion • Unset keys for deletion • Create import tokens • Retrieve import tokens	CCKM does not support the following: • Wrap keys • Unwrap keys

Service Access Role	Access Permissions (Allowed Actions) in IBM Cloud	Notes
Manager	- All actions that a reader or a writer is allowed to perform - Update keys - Enable keys - Disable keys - Delete keys - Restore keys	

For detailed information about these roles, refer to the topics covering “Managing user access” and “Roles and Permissions” within IBM Cloud documentation.

Action Permissions Mapping for CCKM and IBM HPCS/Key Protect

A mapping of each of the supported CCKM actions (you can perform in the IBM Cloud) to an IBM HPCS or Key Protect action along with the required Service access role(s) to perform the action is provided in the following table:

CCKM Action	IBM HPCS or Key Protect Actions Required	Required Service Access Role
Synchronize	List keys Retrieve keys	Reader/Writer/Manager
View keys	List keys Retrieve keys	Reader/Writer/Manager
Create native key	Create keys Retrieve keys	Writer/Manager
Upload standard key	Create keys Retrieve keys	Writer/Manager
Upload root key	Create keys Retrieve keys Create import tokens Retrieve import tokens	Writer/Manager
Update key	Retrieve keys Set key policies List key policies	Manager
Enable key	Enable key	Manager
Disable key	Disable key	Manager
Schedule Delete key	Set keys for deletion	Writer/Manager
Delete key	Retrieve keys Delete keys	Writer/Manager
Cancel Delete key	Unset Keys for deletion	Writer/Manager

CCKM Action	IBM HPCS or Key Protect Actions Required	Required Service Access Role
Restore key	Retrieve keys Restore keys	Writer/Manager
Rotate key	Rotate keys Retrieve keys	Writer/Manager
Export Keys	There is no equivalent action within IBM HPCS or Key Protect for this CCKM action.	N/A

CCKM's Interaction with IBM HPCS and Key Protect

CCKM uses IBM Cloud credentials to work with the IBM Cloud APIs. To log into the IBM key management service of CCKM, you are required to have an IBM username and IBM Cloud API key or IBM service ID and IBM service ID API Key. To obtain an IBM Cloud API key or service ID API key, generate it in the IBM portal. For information about generating an IBM Cloud API key or IBM service ID API key, see section ["Prerequisites on IBM HPCS and Key Protect" below](#). You can create a IBM credentials file in a JSON format to use when logging into the IBM key management service of CCKM. For more information, see section ["Creating an IBM Credentials File in a JSON Format " on page 190](#).

You can access IBM HPCS or Key Protect instances (depending on which IBM Cloud key management service you are using) by setting access policies on a user or an access group with permissions to the given key management service.

The following access policies are allowed for using CCKM:

- setting a service access policy for a single IBM key
- setting a service access policy for a key management service (HPCS or Key Protect) instance
- setting a service access policy for a resource group containing key management service (HPCS or Key Protect) instances

In order to manage IBM keys in CCKM, a user or service ID must have **Viewer** platform management permission on every key management service (HPCS or Key Protect) instance that contains the keys.

For each IBM key management service (HPCS or Key Protect) instance, CCKM retrieves the following:

- Platform management role the user has on the instance (Administrator, Operator, Editor, or Viewer). Note that the Administrator role should only be assigned to the administrator.
- Service access role the user has on the instance (Reader, Writer, or Manager)

Prerequisites on IBM HPCS and Key Protect

In this section, it is assumed you will use access groups and resource groups when configuring the IBM key management service of IBM HPCS or Key Protect in CCKM.

Note

In this section, it is assumed you are familiar with creating a user and service ID in the IBM portal. Refer to the IBM Cloud documentation for information on how to add a user and a service ID in the IBM portal.

Note

If you plan to configure the IBM key management services of IBM HPCS or Key Protect using user(s) or service ID (without adding them first to an access group), you are required to perform steps 1, 2 (if you are using HPCS), and 3 in this section. However, the remaining steps in this section do not apply. Instead, for step 4, create a user or service ID. Then for step 5, assign an access policy to the user or service ID. The actual step you take in steps 4 and 5 depends on whether you are using a user or service ID. For more information about assigning an access policy to a user or service ID, refer to the IBM Cloud documentation.

Prior to configuring and accessing the IBM key management service of IBM HPCS or Key Protect in CCKM, do the following:

Note

Refer to the sections below for more information about each of these steps.

1. Create an instance of IBM HPCS or Key Protect (depending on which service you are using) in the IBM Cloud by using the IBM Cloud console or the IBM Cloud CLI. For more information, see ["Create an Instance of IBM HPCS or Key Protect in IBM Portal" below](#).
2. (**Applicable to HPCS only**) Initialize your HPCS service instance in the IBM Cloud. For more information, see ["Initialize your HPCS Service Instance" below](#).
3. Generate an API key for a user or service ID in the IBM portal. To generate an API key for a user in the IBM portal, see ["Generate a User API Key in IBM Portal" on the next page](#). To generate an API key for a service ID in the IBM portal, see ["Generate Service ID API Key in IBM Portal" on the next page](#).
4. Create an access group. For more information, see ["Create Access Group" on the next page](#).
5. Add users or service ID to your access group. To add user to your access group, see ["Add User\(s\) to Access Group" on page 188](#). To add a service ID to your access group, see ["Add Service ID\(s\) to Access Group" on page 188](#).
6. Assign an access policy to the access group. For more information, see ["Assign Access to your Access Groups \(for Users and Service IDs\)" on page 188](#).
7. Assign additional access for the service ID. For more information, see ["Assign Additional Access Policy to Service ID" on page 189](#).

Create an Instance of IBM HPCS or Key Protect in IBM Portal

For information on how to create an instance in IBM HPCS or Key Protect, refer to topics related to “provisioning service instance” as it relates to these services in the IBM Cloud documentation.

Initialize your HPCS Service Instance

(**Applicable to HPCS only**) For information on how to initialize your HPCS service instance in the IBM Cloud, refer to the topic of “Initializing your service instance” for HPCS in the IBM Cloud documentation.

Generate a User API Key in IBM Portal

To generate an API key for the user in the IBM portal:

Note

Be sure to save the generated API key to a file as you will not be able to view or retrieve it later from the IBM portal.

1. From the menu bar, click **Manage > Access (IAM)**.
2. Select **IBM Cloud API Keys**.
3. Click **Create API key**.
4. Enter a name and description for the API key.
5. Click **Create**.
6. Click **Copy** to copy the API key or click **Download** to download it.
7. Save the API key.

Generate Service ID API Key in IBM Portal

To generate an API key for the service ID in the IBM portal:

Note

Be sure to save the generated API key to a file as you will not be able to view or retrieve it later from the IBM portal.

1. From the menu bar, click **Manage > Access (IAM)**.
2. Select **Service IDs**.
3. Click on the name of the service ID for which you want to generate the API Key.
4. Select **API keys**.
5. Click **Create**.
6. Enter a name and description for the API key.
7. Click **Create**.
8. Click **Copy** to copy the API key or click **Download** to download it.
9. Save the API key.

Create Access Group

To create an access group, you need to first set up the group and then add users or service IDs (or users and service IDs) to your access group.

To create an access group:

1. From the menu bar, click **Manage > Access (IAM)**, and select **Access Groups**.
2. Click **Create**.
3. Enter a name and description (optional) for your group
4. Click **Create**.

Add User(s) to Access Group

To add users to your access group:

1. Select the name of the access group to which to add users.
2. Click **Add users**.
3. Select the user to add from the list, and then click **Add to group**.

Add Service ID(s) to Access Group

To add a service ID to your access group:

1. Select the name of the access group to which to add service ID.
2. Select **Service IDs**.
3. Click **Add Service ID**.
4. Select the service ID to add from the list, and then click **Add to group**.

Assign Access to your Access Groups (for Users and Service IDs)

After you create your access group and add users or service IDs to it (or add users and service IDs to it), you can assign an access policy to the group. Note that any policy that you set for the access group is applicable to all entities within the group.

Be sure to review the sections in ["IBM Cloud Identity and Access Management" on page 183](#) and ["Action Permissions Mapping for CCKM and IBM HPCS/Key Protect" on page 184](#) before assigning an access policy to your access group(s).

To assign an access policy to your access group:

1. From the menu bar, click **Manage > Access (IAM)**, and select **Access Groups**.
2. From the row for the group that you want to assign access, select the **Actions List** of actions icon menu, and click **Assign access**.
3. Select the name of the access group that you want to assign access.
4. Select the **Access policies** tab, and then click **Assign access**.
5. From **Assign access group additional access**, select **IAM services**.
6. From **Which services do you want to assign access to?**, select **Hyper Protect Crypto Services** or **Key Protect** (depending on which service you will use).
7. Under **How do you want to scope the access?**, select **All resources**.
8. Under **Platform access**, select one or more of the following options depending on the permissions you are granting to this access group:
 - **Administrator**
 - **Editor**
 - **Operator**
 - **Viewer**

9. Under **Service access**, select one or more of the following options depending on the permissions you are granting to this access group:
 - **Manager**
 - **Reader**
 - **Writer**
10. Click **Add > Assign**.

Assign Additional Access Policy to Service ID

To assign additional access policy to your service ID:

1. From the menu bar, click **Manage > Access (IAM)**, and select **Service IDs**.
2. From the row for the Service ID that you want to assign access, select the **Actions List** from the actions icon menu, and click **Assign access**.
3. From **Assign service ID additional access**, select the **Account Management** tab.
4. From **Which services do you want to assign access to**, select **IAM Identity Service**.
5. From **How do you want to scope the access**, select **All resources**.
6. From **Platform access**, select one or more of the following options depending on the permissions you are granting to the service ID:
 - **Operator**
 - **Editor**
 - **Administrator**
7. Click **Add**.
8. From **Which services do you want to assign access to**, select **user management**.
9. From **How do you want to scope the access**, select **All resources**.
10. From **Platform access**, select one or more of the following options depending on the permissions you are granting to the service ID:
 - **Viewer**
 - **Editor**
 - **Administrator**
11. Click **Add**.
12. From **Which services do you want to assign access to**, select **IAM Access Groups Service**.
13. From **How do you want to scope the access**, select **All resources**.
14. From **Platform access**, select one or more of the following options depending on the permissions you are granting to the service ID:
 - **Viewer**
 - **Operator**
 - **Editor**
 - **Administrator**
15. Click **Add > Assign**.

Creating an IBM Credentials File in a JSON Format

When prompted for your IBM login credentials on the IBM login page within CCKM, you can either enter your IBM Username and API Key or IBM service ID and API Key. You can also upload an IBM credentials file. The IBM credentials file should be in a .json format (can be written in Notepad and saved for upload).

The following is sample .json file for username and API key (add your username and API key to the file):

```
{
  "ibm_username": "ibmusernamehere",
  "ibm_apikey": "XQq81eP78EHSG9EZbXwgAYJUfRMpC8wNiNkqiFudkZeT"
}
```

The following is sample json file for service ID and API key (add your service ID and API key to the file):

```
{
  "ibm_serviceid": "ServiceId-number-here",
  "ibm_apikey": "Zr65BVx4yueysm2uxpA8osoNpqSh4GNkj_nOphT29cyX"
}
```

Access CCKM for IBM

For details about accessing CCKM as a CCKM user for the CCKM service for IBM Key Management, see [Chapter 3: "About CCKM Admin and User Portals" on page 27](#). This chapter includes information on the ways in which to access the CCKM service for Key Management.

IBM CCKM User Portal

The CCKM User Portal for IBM contains the following menu options in the left-hand navigation bar:

- **Home:** **Home** page displays key operation and count metrics.
- **Keys:** **Keys** page displays key management overview and provides the upload, create, and synchronize functionality, and the manual rotation, auto-rotation, delete, restore, and update key operations available in CCKM.
- **Reports:** **All Reports** overview page links to detailed individual reports.
- **Schedule:** In **Schedule**, you can schedule automatic key rotation and key synchronization.
- **Logs:** **Logs** page lists individual logs that can be searched, sorted, and viewed.
- **Key Sources:** In **Key Sources**, you can generate keys in CCKM to upload to IBM. CCKM provides support for the use of CipherTrust Manager and DSM as key sources to generate, store, and retrieve the encryption keys that CCKM uses and manages.

Note: CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

- **Settings:** The **Settings** page allows you to add the email addresses of the recipients who are to receive email notifications and configure the settings for email notifications and syslog servers. From this page, you can also enable CCKM reporting for IBM keys using IBM Cloud Activity Tracker and LogDNA and view the release number of the CCKM running.

To display the context of which Key Management service you are currently logged into within CCKM, the icon associated with the Key Management service displays at the top of the left-hand navigation bar.

The arrow button at the bottom of the navigation bar allows you to expand the bar to view the names of the menu options and collapse it to view only the icons associated with the menu options. Click the arrow button to either expand or collapse the navigation bar.

On the top-right side of the CCKM portal for IBM Cloud, the following options are available:

- **Azure Cloud Service Context Switcher:** This context switcher allows you to login into another Azure cloud service different from the current Azure cloud service into which you are currently logged. If you select the Azure cloud in which you are currently logged, the Multi-account Login/Logout window for the Azure account(s) displays. For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32.](#)
- **AWS Login:** If you are not already logged into AWS, clicking on the AWS Login icon brings you to the AWS Password Login page where you can enter your AWS credentials to access the CCKM AWS cloud service (from the current cloud service in which you are currently logged). If you are already logged into AWS, clicking on the AWS Login icon brings you to the Multi-account Login/Logout window for the AWS account(s). For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32.](#)
- **Google Cloud Service Context Switcher:** This context switcher allows you to login into another Google Cloud organization different from the current Google Cloud organization into which you are currently logged. If you select the Google Cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Google account(s) displays. For more information about the **Multi-account Login/Logout** window, see ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32.](#)
- **Salesforce Cloud Service Context Switcher:** This context switcher allows you to login into another Salesforce cloud service different from the current Salesforce cloud service into which you are currently logged. This is Salesforce, Salesforce Government, or Salesforce Sandbox. If you select the Salesforce cloud in which you are currently logged, the Multi-account Login/Logout window for the Salesforce account(s) displays. For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32.](#)
- **Logout:** Logout from CCKM.

Home

The **Home** page provides a dashboard to display key operation and count metrics.

Keys

The **IBM Cloud Keys** page allows IBM Cloud administrators to view all of the keys and manage them within the boundary of granted permissions. Only the keys to which you have permissions to access display in **Keys** page.

At the top right of the **Keys** page, the following buttons are available:

- **Export Keys:** Allows you to export the contents of the Keys list into a report in a CSV format.
- **New Key:** Allows you to upload an external key (BYOK) or create a native key. For information on how to upload an external key or create a native key, see section ["Create a Key" on page 194.](#)
- **Synchronize:** Allows you to download any keys that were created in IBM HPCS or Key Protect, outside of the CCKM portal, into CCKM. For information on how to synchronize your keys, see ["Synchronize Keys" on page 204.](#)

In this section, create, delete, update, rotate, synchronize, exports keys are described.

A **Search** box is available at the top the **IBM Cloud Keys** page allowing for a search of a specific key by entering any one of the following parameters:

- Name: The name of the key.
- Key ID: The globally unique identifier for the key.
- Region: The IBM Region to which the key is uploaded.
- Key Protect/HPCS Name: The name of the HPCS or Key Protect instance.
- Creation Date: Indicates the date on which the key was created.
- Type: Indicates whether the key of a Standard or Root Key type.
- Key State: The state of the key. Valid values are:
 - Pre-activation
 - Enabled
 - Disabled
 - Deactivated
 - Destroyed

Note: The key states of Pre-activation and Destroyed transition to other states. For this reason, you may not see keys with these states displayed in the CCKM UI.

- Key Material Origin: Specifies the origin of the key material.
- Rotation Status: Available options are AUTO ON and AUTO OFF.

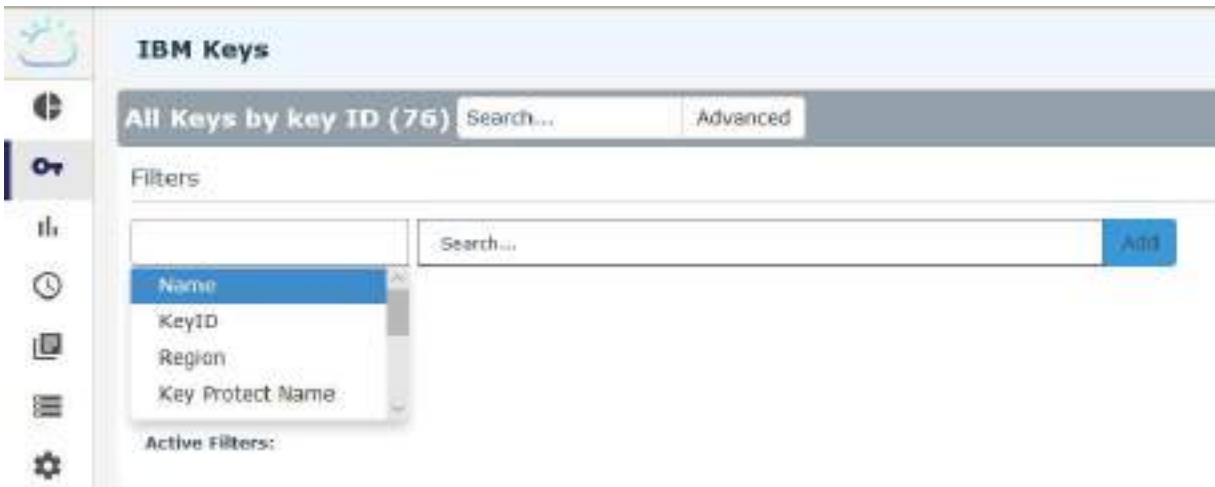
The **Advanced** tab allows you to filter your search on each of the columns displayed on the IBM Cloud Keys page with the exception of the **Actions** column:

- Name
- Key ID
- Region
- Service Name
- Service ID
- Creation Date.
- Type
- Key State
- Dual Auth Delete State
- Key Expiration Date
- Key Material
- Rotation Status

- Service

Each of these columns are described in more detail below in this section.

To select to filter based on a specific column from the **IBM Cloud Keys** page, click **Advanced**. The **Filters** section displays. From the column filter drop-down menu, select the name of the column on which to filter your search. For example, select **Region**. In the **Search** box, enter the value on which to search (for example, Key Protect Name) and click **Add**. For the **Region**, **Type**, **Key State**, **Dual Auth Delete State**, **Rotation Status**, and **Service** columns, a drop-down menu is available from which to select a value for the search. Click the down arrow, select a value from the available options, and then click **Add**. For the **Creation Date** and **Key Expiration Date**, select a start and end date for the search and then click **Add**. The specified search filter is then added to **Active Filters**. The results of the search displays in the **IBM Cloud Keys** table. You can narrow your search by selecting more columns to add to the filter. To clear the filtered results, from **Active Filters**, click on the “X” (or the **Delete** button) for each of the named filters.



The following columns display on the **IBM Cloud Keys** page:

- Name: The name of the key.
- Region: The IBM Region to which the key is uploaded.
- Service Name: The ID of the HPCS or Key Protect instance in which the key stored.
- Service: HPCS or Key Protect service.
- Creation Date: Indicates the date on which the key was created.
- Type: Indicates whether the key is a Root or a Standard key.
- Key State: The state of the key. The valid values are:
 - Pre-activation
 - Enabled
 - Disabled
 - Deactivated
 - Destroyed

- **Dual Auth Delete State:** The state of the dual authorization. The valid values are:
 - Disabled
 - Enabled
 - Scheduled
 - Expired
 - Canceled
 - Deleted
- **Key Material Origin:**
 - If the key was uploaded from CCKM as a BYOK key, then the value is INTERNAL(<source_key_name>).
 - If the key was not uploaded from CCKM as a BYOK key, then the value is EXTERNAL.
 - If the key material was created within IBM HPCS, then the value is HPCS.
 - If the key material was created within IBM Key Protect, then the value is Key Protect.
- **Rotation Status:** Indicates whether the key is in an active rotation schedule or if it has been rotated in the past.
 - Auto—if the autorotation is currently enabled.
 - blank—if the autorotation is disabled.
- **Version Count:** Reflects the number of versions the key currently has. To view information about all of the key versions, click on the number of the version count. The **Key Versions** dialog box displays. In this dialog box, the key name, version ID, and date on which the key version was created (Created On) displays. For more information, see ["View All Key Versions" on page 205](#).

Note

If you have the manager role, you have the permissions of writer and viewer.

- **Actions:** Allows you to select any of the following actions on a given key:
 - Enable (Manager permission required)
 - Disable (Manager permission required)
 - Rotate (Writer permission required)
 - Schedule Delete (Writer or Manager permission required)
 - Delete (Manager permission required)
 - Cancel Delete (Manager permission required)
 - Restore (Writer/Manager permissions required)
 - Update (Manager permission required)

Note

Writer has permissions of writer and viewer.

Create a Key

You have two options to create a new key in an IBM HPCS or Key Protect instance:

- Create a native key
- Upload a BYOK

Create a Native Key

You can create a native key in either IBM HPCS or Key Protect using CCKM. In the process of creating a native key in either of these supported IBM key management services, key material from IBM HPCS or Key Protect (depending on which key management service you are using) along with key metadata from CCKM are used. Once created, the key displays as either an HPCS or Key Protect key (depending on which key management service you are using) under the Key Material Origin column within the Keys > All Keys list page in CCKM.

Note

The access permission of a Writer or a Manager role is required for creating native keys.

For more information, see the following sections:

- ["Create a Native Key \(IBM HPCS\)" below](#)
- ["Create a Native Key \(IBM Key Protect\)" below](#)

Create a Native Key (IBM HPCS)

Note

The access permission of a Writer or a Manager role is required for creating native keys.

To create a native key in an IBM HPCS instance:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New Key** button and then **Create**. The **Create Native Key** dialog box displays.
3. In the **Key Name** box, enter the name of the IBM native key to create.
4. From **IBM Service**, select **HPCS**.
5. From the **Key Protect/HPCS** box, select the IBM HPCS instance to which you will add the new key.
6. For **Key Type**, select whether this key is a **Standard Key** or a **Root Key**.
7. (Optional) Enter the expiration date of the key material in the **Key Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
8. (Optional) In the **Description** box, enter a description for the key you are uploading.
9. (Optional) In the **Tags** box, enter the tag name and then click the **+** button.
10. Click **Create**.
(Reset clears the page before it is created, if necessary.)
The new key is displayed in the list.

Create a Native Key (IBM Key Protect)

Note

The access permission of a Writer or a Manager role is required for creating native keys.

To create a native key in an IBM Key Protect instance:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New Key** button and then **Create**. The **Create Native Key** dialog box displays.

3. In the **Key Name** box, enter the name of the IBM native key to create.
4. From **IBM Service**, select **Key Protect**.
5. From the **Key Protect/HPCS** box, select the IBM Key Protect instance to which you will add the new key.
6. For **Key Type**, select whether this key is a **Standard Key** or a **Root Key**.
7. (Optional) Enter the expiration date of the key material in the **Key Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
8. (Optional) In the **Description** box, enter a description for the key you are uploading.
9. (Optional) In the **Tags** box, enter the tag name and then click the **+** button.
10. Click **Create**.
(Reset clears the page before it is created, if necessary.)
The new key is displayed in the list.

Upload your Own Key (BYOK)

Upload the AES 256 keys created from the **Key Sources** page of CCKM to an IBM key management service instance (HPCS or IBM Key Protect instance) you are using. These "BYOK" keys are the keys you generated from the key source you are using (CipherTrust Manager or DSM).

For more information, see the following sections:

- ["Upload Your Own Key \(BYOK\) \(IBM HPCS\)" below](#)
- ["Upload Your Own Key \(BYOK\) \(IBM Key Protect\)" on the next page](#)

Upload Your Own Key (BYOK) (IBM HPCS)

Note

The access permission of a Writer or Manager role is required for uploading a BYOK key.

To upload a BYOK key from CCKM to an IBM HPCS instance:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New** button and then **Upload**. The **Upload a Key** dialog box displays.
3. In the **Key Name** box, enter the name of the IBM key to upload.
4. From **IBM Service**, select **HPCS**.
5. From the **Key Protect/HPCS** box, select the HPCS instance to which you will upload the new key.
6. For **Key Type**, select whether this key is a **Standard Key** or a **Root Key**.
7. From **Key Provider** box, select **CipherTrust Manager** or **DSM** depending on the key source you are using for the key.
8. From **Source Key** box, select the source key to use for the BYOK key. The keys available in the drop-down menu are the keys generated from CipherTrust Manager or DSM depending on the key source you are using.
9. (Optional) Enter the expiration date of the key material in the **Key Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
10. (Optional) In the **Description** box, enter a description for the key you are uploading.

11. (Optional) In the **Tags** box, enter the tag name and then click the **+** button.
12. Click **Upload to IBM**.
The uploaded key is displayed in the list.

Upload Your Own Key (BYOK) (IBM Key Protect)

Note

The access permission of a Writer or Manager role is required for uploading a BYOK key.

To upload a BYOK key from CCKM to an IBM Key Protect instance:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New** button and then **Upload**. The **Upload a Key** dialog box displays.
3. In the **Key Name** box, enter the name of the IBM key to upload.
4. From **IBM Service**, select **Key Protect**.
5. From the **Key Protect/HPCS** box, select the Key Protect instance to which you will upload the new key.
6. For **Key Type**, select whether this key is a **Standard Key** or a **Root Key**.
7. From **Key Provider** box, select **CipherTrust Manager** or **DSM** depending on the key source you are using for the key.
8. From **Source Key** box, select the source key to use for the BYOK key. The keys available in the drop-down menu are the keys generated from CipherTrust Manager or DSM depending on the key source you are using.
9. (Optional) Enter the expiration date of the key material in the **Key Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
10. (Optional) In the **Description** box, enter a description for the key you are uploading.
11. (Optional) In the **Tags** box, enter the tag name and then click the **+** button.
12. Click **Upload to IBM**.
The uploaded key is displayed in the list.

Delete a Key

You have the following options to delete a key from IBM HPCS or Key Protect using CCKM:

- ["Delete a Key" below](#)
- ["Delete a Key Using Dual Authorization" on page 199](#)

Delete a Key

From the CCKM **Keys** page, you can delete a key from either IBM HPCS or Key Protect. After deletion, the action of **Restore** is available within the **Actions** column for certain keys. For more information, see ["Restore a Key" on page 201](#).

Note

The access permission of a Manager role is required for deleting a key.

To delete a key:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. Find the row that holds the key you wish to delete and click **Delete** from the **Actions** column. The **Delete Key** dialog box displays.
A warning message displays indicating that this action will delete the named key from the named instance. The dialog box also indicates to type the supplied confirmation phrase in the text box if you still wish to delete the key and provides other pertinent information.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Delete**.

Deleting Keys Using Dual Authorization

To prevent accidental or malicious deletion of root or standard keys in IBM HPCS or Key Protect, IBM provides the ability to use a dual authorization process to delete these keys using a policy. CCKM provides support for the key deletions using dual authorization. When the dual authorization delete policy is enabled, thereafter a key covered under this policy requires authorization from two users to delete it. One user provides the authorization to delete the key by scheduling to delete the key. In CCKM, this user uses the **Schedule Delete** button under the **Actions** column available on the **Keys** page to schedule the key deletion. The schedule is valid for 7 days. During this period, the other authorized user can delete the key using the **Delete** button from the **Actions** column. After the key is deleted, it is placed in the **Destroyed** state as displayed in the **Dual Auth Delete State** column on the **Keys** page. If the key is *not* scheduled for deletion after 7 days, it is placed in the **Expired** state. To schedule to delete a key after the key is in this state, you must schedule to delete a key once again.

The policy for dual authorization deletion can be enabled at a key level or at an instance level. From the CCKM, you can enable the policy at a key level using the **Enable Dual Authorization Delete** toggle within the **Update Key** dialog box available on the **Keys** page. And once this policy is enabled at the key level, you cannot disable it. However, to enable dual authorization deletion at the instance level, you must do so from the IBM Cloud portal. *After* the policy is enabled at the instance level, the policy applies only to the new keys that are created on the instance.

The state of **Enabled** displays in the **Dual Auth Delete State** column on the **Keys** page for keys that have the policy of dual authorization deletion enabled.

For more information about enabling the dual authorization for IBM HPCS or Key Protect in general and at the instance level, refer to the appropriate online IBM Cloud documentation. For more information about enabling dual authorization at a key level from CCKM, see ["Update Key for Dual Authorization of Key Deletion" on page 202](#).

In IBM, you also have the option to cancel the scheduled key deletion. In CCKM, you can cancel the scheduled key deletion using the **Cancel Delete** button from the **Actions** column. To delete a key after this cancellation, you must again go through the dual authorization process starting with schedule delete.

For more information, see the following sections:

- ["Schedule to Delete a Key Using Dual Authorization" on the next page](#)
- ["Delete a Key Using Dual Authorization" on the next page](#)
- ["Cancel Schedule Delete Using Dual Authorization " on page 200](#)

Schedule to Delete a Key Using Dual Authorization

From the **Keys** page, you can schedule to delete a key from IBM HPCS or Key Protect using dual authorization. *Prior* to scheduling to delete a key, you must enable the policy for dual authorization deletion at the key level or instance level. Once enabled, the state of **Enabled** displays in the **Dual Auth Delete State** column and the action of **Schedule Delete Key** is available for selection within the **Actions** column on the **Keys** page.

After the key is scheduled for deletion, the state of **Scheduled** displays in the **Dual Auth Delete State** column and the actions of **Delete** and **Cancel Delete** are available within the **Actions** column for the key.

The schedule is valid for 7 days. During this period, the other authorized user can delete the key using the **Delete** button.

Note

The access permission of a Writer or a Manager role is required for scheduling to delete a key using dual authorization.

To schedule to delete a key using dual authorization:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. Find the row that holds the key you wish to schedule to delete and click **Schedule Delete** from the **Actions** column. The **Schedule Delete Key** dialog box displays.
A warning message displays indicating that this action will schedule delete the named key from the named instance. The dialog box also indicates to type the supplied confirmation phrase in the text box if you still wish to schedule delete the key and provides other pertinent information.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Schedule Delete** to confirm.

Delete a Key Using Dual Authorization

From the **Keys** page, you can delete a key from IBM HPCS or Key Protect using dual authorization. After a key is deleted using dual authorization, the action of **Restore** is available within the **Actions** column for certain keys. For more information, see ["Restore a Key" on page 201](#).

Note

The access permission of a Manager role is required for delete a key using dual authorization.

To delete a key using the dual authorization:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. Find the row that holds the key you wish to delete and click **Delete** from the **Actions** column. The **Delete Key** dialog box displays.
A warning message displays indicating that this action will delete the named key from the named instance. The dialog box also indicates to type the supplied confirmation phrase in the text box if you still wish to delete the key and provides other pertinent information.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Delete**.

Cancel Schedule Delete Using Dual Authorization

From the **Keys** page, you can cancel a scheduled key deletion using dual authorization. *After* the scheduled key deletion is canceled successfully, the state of **Canceled** displays in the **Dual Auth Delete State** column, and the action of **Schedule Delete** is once again available from the **Actions** column for the given key.

Note

The access permission of a Manager role is required to cancel the scheduled deletion of a key using dual authorization.

To cancel the scheduled key deletion using dual authorization:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. Find the row that holds the key you wish to cancel the scheduled key deletion and click **Cancel Delete** from the **Actions** column. The **Cancel Delete Key** dialog box displays.
A warning message displays indicating that this action will cancel the scheduled key deletion for the named key in the named instance. The dialog box also indicates to type the supplied confirmation phrase in the text box if you still wish to cancel the scheduled key deletion and provides other pertinent information.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Cancel Delete** to confirm.

Disable and Enable a Root Key

IBM HPCS and Key Protect key management services provides you the ability to disable a root key that is in an active state to temporarily suspend its key operations (encrypt and decrypt) in cases where there is a potential security risk with the associated data. After determining there is no risk or there is no longer a risk, you can re-enable the disabled key to reactivate it for key operations. CCKM provides support for this feature allowing you to disable and enable a root key in CCKM with the use of the **Disable** and **Enable** options within the **Actions** column on the **Keys** page. A disabled root key displays as **Disabled** within the **Key State** column on the **Keys** page while an enabled root key displays as **Enabled**.

Note

The access permission of a Manager role is required for disabling and enabling a root key.

For more information, see the following sections:

- ["Disable a Root Key" below](#)
- ["Enable a Root Key" on the next page](#)

Disable a Root Key

Note

The access permission of a Manager role is required for disabling a root key.

To disable a root key from an HPCS or Key Protect instance (depending on which key management service you are using):

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. Find the row that holds the key you wish to disable and click **Disable** from the **Actions** column. The key is disabled.

Enable a Root Key

Note

The access permission of a Manager role is required for enabling a root key.

To enable a root key on an HPCS or Key Protect instance (depending on which key management service you are using):

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. Find the row that holds the key you wish to enable and click **Enable** from the **Actions** column. The key is enabled.

Restore a Key

CCKM allows a key that has been deleted from a given IBM Key Protect instance through CCKM to be restored to the instance. You have the option to delete any key from a given IBM Key Protect instance through CCKM. This key can be an IBM native or BYOK key of either the root or standard key type. After key deletion, its key state changes from **Enabled** to **Destroyed**, which is reflected in the **Key State** column on the **Keys** page. The **Restore** action within the **Actions** column within the **Keys** page is available for a key in a **Destroyed** key state. In the **Destroyed** key state, a key cannot be used for cryptographic operations, such as data encryption or decryption. However, the key metadata is still available to be viewed.

CCKM allows a key that has been deleted from a given IBM HPCS instance through CCKM to be restored to the instance. *However, only* IBM BYOK keys of the root key type can be restored. After an IBM BYOK key of the root key type is deleted, its key state also changes from **Enabled** to **Destroyed**, which is reflected on the **Keys** page. After key deletion, the **Restore** action within the **Actions** column is also available for this key. After key deletion of an IBM native or BYOK key that is of the standard key type, no data is shown in the **Actions** column because there is no longer any action available to be taken for this key.

For the IBM HPCS and Key Protect keys that are eligible for key restoration, you have a period of up to 30 days to restore a key. After a key from an HPCS or Key Protect instance is restored, the key enters the **Enabled** key state once again. For both HPCS and Key Protect keys, the key metadata is accessible between the period of 31 and 90 days (*inclusive*). However, during this period, the key can no longer be restored. **Any** time *after* 90 days, the key is automatically purged, which means the key no longer exists within the given IBM instance.

Note

If a key that is eligible to be restored is deleted successfully using the dual authorization process, and then it is restored successfully, the key's dual authorization delete state as reflected in the **Dual Auth Delete State** column on the **Keys** page is set to **Enabled**.

For more information, see the following sections:

- ["Restore a Key \(IBM HPCS\)" on the facing page](#)
- ["Restore a Key \(IBM Key Protect\)" on the facing page](#)

Restore a Key (IBM HPCS)

To restore an HPCS BYOK key of the root key type in CCKM:

Note

In CCKM, you cannot restore an IBM native or BYOK key that is of the standard key type.

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list page, select **Restore** from the **Actions** column of the key. The **Restore Key** dialog box displays with a warning indicating that the specified key name will be restored to the specified HPCS instance.
3. Click **Restore**. If successful, a success message is displayed. The key and its metadata are restored to the instance. The key state reflected in the **Key State** column on the **All Keys** list page shows **Enabled**.

Restore a Key (IBM Key Protect)

To restore a Key Protect key in CCKM:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list page, select **Restore** from the **Actions** column of the key. The **Restore Key** dialog box displays with a warning indicating that the specified key name will be restored to the specified Key Protect instance.
3. Click **Restore**. If successful, a success message is displayed. The key and its metadata are restored to the instance. The key state reflected in the **Key State** column on the **All Keys** list page shows **Enabled**.

Update Key

You have the following options to update a key in IBM HPCS or Key Protect using CCKM:

- ["Update Key for Dual Authorization of Key Deletion" below](#)
- ["Update Key to Enable Automatic Key Rotation Policy for Root Key" on the next page](#)

Update Key for Dual Authorization of Key Deletion

In CCKM, you can enable dual authorization to delete a key at the key level. After a key is enabled for dual authorization, the state of **Enabled** displays in the **Dual Auth Delete State** column and the action of **Schedule Delete Key** is available for selection within the **Actions** column on the **Keys** page.

Enabling dual authorization at the key level does not affect the other keys in a given instance (HPCS or Key Protect).

Note

After you enable dual authorization on a key, you cannot disable it.

Note

The access permission of a Manager role is required to enable dual authorization for deletion of a key.

To enable dual authorization for key deletion at the key level:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. Find the row that holds the key you wish to update and click **Update** in the **Actions** column. The **Update key** dialog box displays.

3. From the **Update Key** dialog box, click **Enable Dual Authorization Delete** to turn this toggle to turn it to **On**. By default, this toggle is initially set to **Off**.
4. Click **Update**.

Update Key to Enable Automatic Key Rotation Policy for Root Key

In CCKM, you can enable an automatic rotation policy for a root key using the IBM key management service (IBM HPCS or Key Protect) you are employing. These IBM key management services manage and perform the key rotation. After the key rotation takes place in an IBM HPCS or Key Protect instance (depending on which service you are using), you can manually synchronize your IBM keys in CCKM. You can also schedule key synchronization in CCKM to synchronize keys that have been rotated in IBM HPCS or Key Protect. See section ["Synchronize Keys" on the facing page](#) for more information.

Note

The access permission of a Manager role is required to enable an automatic rotation policy for a Root key.

To enable automatic key rotation policy for a root key:

1. In the CCKM portal, in the left navigation bar, select **Keys**. The **Keys** page displays.
2. Find the row that holds the key you wish to update and click **Update** in the **Actions** column. The **Update key** dialog box displays.
3. From the **Update Key** dialog box, click **Enable autorotation in key policy** to turn this toggle to **On**. By default, this toggle is initially set to **Off**.
4. In the **Specify the key rotation time interval in months** box, enter the number of months for the key rotation interval. The minimum is 1, and the maximum is 12.
5. Click **Update**.

Rotate Key

CCKM allows for a manual rotation and a scheduled rotation (autorotation) of IBM Root keys.

Note

IBM does not support the rotation of IBM standard keys.

When you manually rotate a key in CCKM, use the **Rotate Now** tab in the **Rotate IBM Key** dialog box. Upon receiving the CCKM API call to rotate a specific key, IBM rotates the key. During the rotation of a key (manual or autorotation), new key material from the key source is used to replace the existing key material in the key while the key name and key ID remains the same. A new key version is created after the rotation. The supported key sources are CipherTrust Manager, DSM, and Native. If the key source is Native, then the key material originates from either IBM HPCS or Key Protect (depending on which IBM key management service you are using).

Auto rotating a key is simply rotating a key at a scheduled time. Key autorotation requires that you first schedule the autorotation in the **Schedules** page prior to enabling this feature in the **Auto Rotate** tab in the **Rotate IBM Key** dialog box. For information about scheduling autorotation, see ["Schedules" on page 207](#).

Rotate a Key Manually

Note

The access permissions of a Writer or Manager role are required to manually rotate a key.

To rotate a key manually:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list, find the row that holds the key you wish to rotate and select **Rotate Key** from the **Actions** column. The **Rotate IBM Key** dialog box is displayed.
3. Select the **Rotate Now** tab.
4. In **Key Provider** box, the key source that is preselected depends on the source (provider) of the key. The supported key sources are **CipherTrust Manager**, **DSM**, and **Native**.
5. Click **Rotate**.

Enable Autorotation of Key

Note

The access permissions of a Writer or Manager role are required to enable autorotation of a key.

To enable autorotation of a key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list, find the row that holds the key you wish to rotate and select **Rotate Key** from the **Actions** column. The **Rotate IBM Key** dialog box is displayed.
3. Select the **Auto Rotate** tab.
4. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.
5. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
6. From **Key Provider** box, select **CipherTrust Manager**, **DSM**, or **Native** depending on the key source (provider) you are using for the key.
7. Click **Submit**.

Synchronize Keys

After the first login into the CCKM IBM Cloud, all of your keys (Standard and Root) from all instances (IBM HPCS or Key Protect instances) to which you have access are synchronized with CCKM. IBM key metadata will be downloaded to CCKM during synchronization and displayed in the CCKM UI.

Note

If keys are auto-rotated in IBM HPCS or Key Protect, be sure to synchronize your keys in CCKM.

The access permissions of Viewer, Writer or Manager role are required to synchronize keys.

To synchronize your keys:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. Click **Synchronize**. The **Synchronize Key** dialog box displays.
3. Click **Synchronize** to proceed with the synchronization.

Export Keys

Export Keys allows you to export the contents of the **Keys** list into a report in a CSV format.

Note

No role access permissions are required to export keys.

To export your **Keys** list:

1. In the left navigation bar, click **Keys**.
2. Click **Export Keys**.
This will download all of the keys in a report in a CSV format.

View All Key Versions

From the **Keys** page, you can view information about all of the key versions of a given key by accessing the **Key Versions** dialog box. A new key version is created after a key rotation (manual or autorotation). The **Key Versions** dialog box displays all the versions of a key including the key name, version ID (**Version**), and date on which it was created (**Created On**). You can display the next 5, 10, 15, 20, 25, 50, or 100 entries using the **Show entries** drop down menu. You can view each available page by scrolling through the pages by clicking the less than "<" or greater than ">" symbols or by clicking on a given page number.

To view all of the key versions of a key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Keys** page, find the name of the key whose key versions you wish to view.
3. Under the **Version Count** column, click on the number. The **Key Versions** dialog box displays. You can scroll through the list to view all of the key versions.
4. Click the **Close** button to close the **Key Versions** dialog box.

Reports

To run the **Combined Key Activity Reconciliation** and **IBM Key Activity** reports, you are required to provision an instance of IBM Activity Tracker with LogDNA with the IBM HPCS or Key Protect instances (depending on which IBM key management service you are using) that are to be tracked before IBM keys are represented in the CCKM reports. You also must also set these services up in CCKM using the **General** tab in the **Settings** page.

To run an **IBM Cloud Key Service Usage** report successfully, you must *first* create a key registration between the key and the cloud resource that it protects. For information about key registrations and how to create a registration between a key and a resource, refer to IBM Cloud documentation.

Choose **Reports** in the left-hand navigation to access the following reports:

- **Cloud Key Manager User Action Report**

Lists the users and their actions in creating, deleting, updating and rotating keys in the CCKM portal.

- **Combined Key Activity Reconciliation Report**

Click **Run Report** to generate a new list of events connected to a key. Reconciliation before and after key import the cloud service. The following actions can be re conciliated: creation, deletion, rotate, and policy update.

- **IBM Key Activity Report**

Illustrates the activity type performed by a specific user at the stated time.

- **IBM Key Aging Report**

Provides information about key expiration using the following columns:

- Key Name
- Region
- Key expiration: Date and time at which this key expires.

- **IBM Cloud Key Manager User Action Report**

Lists the users and their actions in creating, deleting, updating and rotating keys in the CCKM portal.

- **IBM Cloud Key Service Usage Report**

Provides information about which IBM cloud resources are protected by which IBM HPCS or Key Protect key using the following columns:

- Key ID
- Key Name
- Service ID
- CRN: The Cloud Resource Name (CRN) uniquely identifies cloud resources.
- Last Modified: The date at which the key was last modified.

IBM Cloud Activity Tracker with LogDNA Service

IBM Cloud Activity Tracker in conjunction with LogDNA allows you to do the following:

- Collect and store logs of API calls made to IBM Cloud resources within your IBM Cloud account.
- Monitor all of the API activities within your IBM Cloud account.

CCKM uses these two applications to generate the Combined Key Activity Reconciliation Report and the IBM Key Activity Report. Prior to generating these two reports in CCKM, you must set up and configure IBM Cloud Activity Tracker and LogDNA. For more information about this setup and configuration, refer to section ["Provisioning IBM Cloud Activity Tracker with LogDNA Service" on the next page](#).

You must also configure the settings of the IBM Cloud Activity Tracker and LogDNA in the Settings page of CCKM so that communication between CCKM and IBM Cloud Activity Tracker with LogDNA can take place and enable reporting on the IBM keys. For more information about this configuration, refer to section "Enabling Reporting for IBM Key Activities Using IBM Cloud Activity Tracker and LogDNA" on page 1.

Note

Only the activities associated with the keys that are managed from CCKM are reflected in these reports. To ensure the Combined Key Activity Reconciliation Report and the IBM Key Activity Report, synchronize your keys with IBM Cloud prior to running these reports. Otherwise, any activities that were performed from the IBM Key Protect portal (if any) will not be reflected in these reports. If you are using only CCKM to manage your keys for the IBM Key Protect instances then this note does not apply to you.

Provisioning IBM Cloud Activity Tracker with LogDNA Service

To set up and configure IBM Cloud Activity Tracker and LogDNA, refer to the "Getting Started Tutorial" under the topic of "IBM Cloud Activity Tracker with LogDNA" within *IBM Cloud Docs*.

The prerequisites to setting up and configuring IBM Cloud Activity Tracker and LogDNA are the following:

- Obtain a user ID (or username) that is a member or an owner of an IBM Cloud account.
- To manage access to the service, your user ID needs administrator platform permissions to manage the IBM Cloud Activity Tracker with LogDNA service.

The following are the high-level steps to take to set up and configure IBM Cloud Activity Tracker and LogDNA described in detail within the "Getting Started Tutorial":

1. Provision an instance of the IBM Cloud Activity Tracker with LogDNA service. (Note the name for the LogDNA instance, IBM Cloud region of the LogDNA instance, and LogDNA Service Key for the instance. You will add this information in the Settings page of CCKM.)
2. Manage access to the monitoring service.
3. Generate IBM Cloud Activity Tracker with LogDNA events.
4. Launch the Web UI for the IBM Cloud Activity Tracker with LogDNA.
5. View events in the Web UI.

Schedules

The **Schedules** page allows cloud administrators to view and schedule the following job types:

- Key Rotation
- Key Synchronization

CCKM provides support for multiple schedules for each of the job types.

At the top right of the **Schedules** page, the Add Schedule button is available from which to add a scheduled job type.

Schedule Key Rotation

CCKM provides support for multiple schedules of the rotation (autorotation) of an IBM key. A new version of the key is added to the IBM key during a scheduled rotation. Scheduled rotation requires that this feature be configured in the **Schedules** page prior to enabling the feature in the **Auto Rotate Key** dialog box (from the **Actions** column of the All Keys list). Your IBM key credentials (IBM username and API key) are required as part of the configuration and are saved to the key source you are using (CipherTrust Manager or DSM).

Note

CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

During a scheduled rotation of a key, CCKM uses your associated IBM key credentials to rotate the key. If the IBM API key expired or changed, then a new IBM API key must be reentered. Otherwise, the scheduled rotation will fail.

You can pause or delete a scheduled key rotation in the **Schedules** page. Note that when a key rotation schedule is deleted from CCKM, then the IBM key credentials are also deleted from the key source you are using (CipherTrust Manager or DSM)

Delete a Key Rotation Schedule

Note

No role access permissions are required to delete a key rotation schedule.

To delete a key rotation schedule, select Delete from the Actions drop-down menu from the Schedules page. Note that the IBM username and IBM Cloud API key associated with the given key rotation schedule is also deleted from the key source you are using (CipherTrust Manager or DSM).

Pause a Key Rotation Schedule

Note

No role access permissions are required to pause a key rotation schedule.

To pause a key rotation schedule, select Yes for Pause within the Add Schedule dialog box and then click **Save**. The setting is saved. To resume the rotation schedule after pausing it, select No for Pause and then click **Save**. The setting is saved.

Schedule Key Synchronization

CCKM allows for a scheduled synchronization of IBM keys. Synchronizing downloads any keys that were created in IBM HPCS or Key Protect, outside of the CCKM portal, into your CCKM. Scheduled key synchronization requires that this feature be configured in the **Schedules** page. You can also pause or delete a scheduled key synchronization in the **Schedules** page. Note that when a key synchronization schedule is deleted from CCKM, then the IBM username and IBM Cloud API key associated with the given key scheduled synchronization are also deleted from the key source you are using (CipherTrust Manager or DSM).

Note

CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

Set up a Key Rotation Schedule

To set up a key rotation schedule:

Note

No role access permissions are required to set up a key rotation schedule.

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_ROTATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).
8. If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.
Set the repeat interval of the key rotation by entering a number in the Repeat Interval and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.
9. If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
10. In the **Username** box, enter your username for the IBM account.
11. In the **IBM API key** box, enter your IBM API key for the IBM account.
12. Click **Save**.

Set up a Key Synchronization Schedule

To set up a key synchronization schedule:

Note

The access permissions of Viewer, Writer or Manager role required to set up a key synchronization schedule. However, it is possible to setup synchronization schedule with none of these permissions on any keys. In this case, synchronization will run on schedule but no key will be downloaded.

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_SYNCHRONIZATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.

7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).
8. If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.
Set the repeat interval of the key rotation by entering a number in the Repeat Interval and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.
9. In the **Username** box, enter your username for the IBM account.
10. In the **IBM API key** box, enter your IBM API key for the IBM account.
11. Click **Save**.

Logs

CCKM maintains logs for events, such as login, key generation, synchronizing of keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. For each logged event listed on the **Logs** page, the following information is displayed:

- Event name
- Severity level
- Date and time of the event
- Event message
- User (name) associated with the event

You can search for a specific logged event from the **Logs** page using the **Search** box. Allowable filters on the search are event name, severity level, event message, and user (name).

Key Sources

The **Key Sources** page displays **all** of the keys that were generated from the key source you are using. If you are logged into multiple clouds within CCKM, this page displays all of the keys from these clouds.

Three tabs are available at the top of the **Key Sources** page:

- **CipherTrust Manager Key:** Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.
- **DSM Key:** Use this tab to use DSM as your key source. The keys are generated and stored within DSM.
- **nShield Key:** This tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for IBM.

Note

CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

Keys listed on the **Key Sources** page are listed by name, cloud ID, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description and actions you can take regarding that key.

A **Search** box is available at the top left of the **Key Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Cloud ID
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

Add (Generate) a Key

From the **Key Sources** page, you can add (or generate) keys in CCKM using CipherTrust Manager or DSM as a key source for *all* of the clouds into which you are currently logged.

Note

The key algorithm currently supported for an IBM key in CCKM is AES256.

Add (Generate) a CipherTrust Manager Key

1. In the CCKM portal, in the left navigation bar, select Key Sources.
2. Select the **CipherTrust Manager Key** tab.
3. Select **Add Key > IBM Key**.
4. In the **Add CipherTrust Manager Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **IBM** and (key) **Algorithm** is fixed to AES256.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.
(Reset clears the screen before it has been saved.)

Add (Generate) a DSM Key

1. In the CCKM portal, in the left navigation bar, select Key Sources.
2. Select the **DSM Key** tab.
3. Select **Add Key > IBM Key**.
4. In the **Add DSM Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **IBM** and (key) **Algorithm** is fixed to AES256.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.
(Reset clears the screen before it has been saved.)

Add (Generate) an nShield Key

The **nShield Key** tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Salesforce.

Delete a Key

Deleting a key from the **Key Sources** page deletes the key from CCKM. If you delete a key from **Key Sources** that has already been uploaded to IBM HPCS or Key Protect (depending on which IBM key management service you are using), the key will not be deleted from the given IBM key management service.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key** or **CipherTrust Manager** tab depending on the key source you used to generate this key.
3. Click **Delete** next to the target key.
4. The **Delete Key** dialog box with a Warning displays asking you to confirm the key deletion by entering the supplied phrase.
5. Enter or Copy/paste the confirmation phrase.
6. Click **Delete**.

Manage keys in Key Sources List Page

In addition to adding and deleting keys, the **Key Sources** page allows you to sort the existing keys by Name, Key Type, Algorithm and Created Date.

Settings

To access the **Settings** page in CCKM, select **Settings** from the left-hand navigation bar. From the **Settings** page, you can:

- Enter the email addresses of the recipients who are to receive the email notifications.
- Enable email notifications to be sent out from CCKM when specific types of key actions are performed. For example, when a key is uploaded to the cloud service. You can also enable CCKM to send reminder notifications for when to rotate a key.
- Enable reporting for IBM keys using IBM Cloud Activity Tracker in conjunction with LogDNA
- Configure a remote syslog server to which to send syslog messages.
- View information about the software release number of the CCKM you are running including the build and version number.

Adding Recipients' Email Addresses (for Notifications)

Use the **General** tab from the **Settings** page to enter the email addresses of the recipients who are to receive email notifications from CCKM. Enter the recipients' email addresses separated by a comma in the **Notification Recipient** box and then click **Save** to save setting.

Enabling Notifications

Using the **Keys Notification** tab from the **Settings** page, you can configure CCKM to send an email notification when a key is deleted from or uploaded to IBM HPCS or Key Protect (depending on which IBM key management service you are using). You can also use this tab to configure CCKM to send a reminder notification for when to manually rotate a key along with the frequency (in days) at which a notification is sent. Add the email address(es) of the recipients in the **General** tab from the **Settings** page. By default, no notifications are sent.

To enable email notifications:

1. In the **Settings** page, select the **Keys Notification** tab.
2. Under **Notifications**, select **On** for each of the notifications you wish to enable:
 - Rotate IBM Cloud Key (reminder notification)
 - Delete IBM Cloud Key
 - Upload IBM Cloud Key
3. If you selected to enable **Rotate IBM Cloud Key**, in the **Days** box, enter the number of days for the frequency a reminder notification is sent.
4. Click **Save**.

Configuring Syslog Server

Note

In this document, the terms "syslog server" and "log server" are synonymous.

The **Syslog** tab from the **Settings** page allows you to configure the remote syslog or log server to which to send CCKM log data using TCP, TLS, or UDP protocol as the transport protocol. If you use the TLS protocol, then you are required to upload the syslog server certificate to CCKM for the TLS connection.

The **Syslog** tab also allows you to delete your syslog server configuration so that log data are no longer sent to the server.

Prior to configuring the setting for a remote syslog server, ensure that you have installed and configured the syslog server including configuring the setting for the port number the transport protocol will use to connect to CCKM. This port number configuration must match the port configuration you set in the **Syslog** tab of the **Settings** page.

Note

Support for both Syslog and Splunk as a remote syslog server is provided in CCKM. However, the use of a Splunk cloud server as a remote syslog server is currently not supported in CCKM.

Configure Remote Syslog Server

To configure a remote syslog server to which to send CCKM log data:

Note

The facility name of "LOCAL1" for the syslog server displays in the **Facility** box. This default value is not available for modification.

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. In the **Hostname** box, enter the hostname or IP address of the syslog server.
3. In the **Port** box, enter the port number the transport protocol uses to connect to the syslog server. Enter a value between 1 and 65535. The default value is 514.
4. In the **Protocol** drop-down menu, select **TCP**, **TLS**, or **UDP** for the transport protocol to use when sending the log data to the syslog server. The default value is **UDP**. If you select **TCP**, proceed to Step 6. If you select **TLS**, proceed to the next step. Otherwise, proceed to Step 7.
5. In the **Certificate** box, select **Choose File** and then select the syslog server certificate to upload to CCKM.
6. Click **Test Connection** to test the connection between CCKM and the remote syslog server. This button is only available for use when either **TCP** or **TLS** is the selected transport protocol.

7. In the **Log Format** drop-down menu, select **CEF** (Common Event Format), **Plain Text**, or **RFC5424** for the format of the log message to use when sending the log data. The default value is **RFC5424**.
8. (Optional) In the **Description** box, enter a description of the remote syslog server.
9. Click **Save**.

Delete Remote Syslog Server Configuration

To delete the configuration of a remote syslog server:

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. Click **Delete**. The **Delete Syslog** dialog box displays.
A warning message displays indicating that this action will delete the syslog settings from CCKM. The message also indicates to type the confirmation phrase (I wish to delete the syslog settings) in the text box if you still wish to delete the syslog settings.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Delete**.

Enabling Reporting for IBM Keys

Use the **General** tab within the Settings page to enable CCKM reporting for IBM keys using IBM Cloud Activity Tracker and LogDNA. However, you are required to provision an instance of IBM Activity Tracker with LogDNA with the IBM key management service instances you are using (IBM HPCS or Key Protect) that are to be tracked before IBM keys will be represented in the CCKM reports. Enabling reporting for IBM keys allows you to generate the Combined Key Activity Reconciliation and IBM Key Activity reports. See section "[Provisioning IBM Cloud Activity Tracker with LogDNA Service](#)" on page 207 for more information about provisioning an instance of IBM Activity Tracker with LogDNA.

Enabling Reporting for IBM Key Activities Using IBM Cloud Activity Tracker and LogDNA

To enable CCKM reporting for IBM key activities using IBM Cloud Activity Tracker and LogDNA:

1. In the **Settings** page, select the **General** tab. The **General** page displays.
2. From the **LogDNA Name** box, provide a name for the LogDNA instance.
3. From the **Select LogDNA Region** drop-down menu, select the IBM Cloud region of the LogDNA instance (that you previously created) for which to enable the reporting.
4. From the **LogDNA Service Key** box, enter the LogDNA service key that is used to authenticate to the LogDNA instance.
5. Click **Add LogDNA Instance**. The **LogDNA Instances** table displays the name and region of the selected LogDNA instance along with the available action of **Remove** under the **Actions** column. This table is dynamic and displays only if you enter information in steps 2 through 4. A message also displays prompting you to click **Save** to save the settings based on the new changes to the LogDNA instance.
6. Click **Save**.

About

Use the About tab to view the build number, version number, patch number (if applicable), and build number of the CCKM you are running.

Chapter 17: CCKM for Salesforce Key Management

Overview	216
Prerequisites on Salesforce	216
Enable and Assign System Permissions for CCKM Login	217
Salesforce Cache-only Keys	218
Access CCKM for Salesforce	220
Salesforce CCKM User Portal	220
Home	222
Tenant Secrets	222
Certificates	231
Reports	232
Schedules	232
Logs	235
Key Sources	235
Settings	237

Overview

CCKM provides support for Salesforce, Salesforce Government, and Salesforce Sandbox. The Salesforce Sandbox is a copy of your production Salesforce environment used primarily for testing, development, or training purposes. Note that in this chapter, the information regarding Salesforce is applicable to Salesforce, Salesforce Government, and Salesforce Sandbox unless otherwise noted. The CCKM UI for Salesforce, Salesforce Government, and Salesforce Sandbox is identical with the exception of the display of the names of the clouds. Your login credentials for Salesforce, Salesforce Government, and Salesforce Sandbox are **unique** to each cloud.

Note

Salesforce provides the following profile interfaces: original profile user interface and enhanced user interface. The procedures provided in this chapter reflect the enhanced user interface. For more information about the Salesforce profile interfaces, refer to the appropriate Salesforce online documentation.

Note

In this document, Salesforce Government Cloud Plus is also referred to as Salesforce Government and Salesforce Government Cloud.

Prerequisites on Salesforce

To use CCKM for Salesforce, you must:

- Have Salesforce's Shield Platform Encryption Service enabled
- Have at least one valid certificate in Salesforce with the following settings:
 - Key Size: 4096
 - "Use Platform Encryption" selected
 - "Exportable Private Key" deselected

- **Be familiar with different types of tenant secrets, Salesforce 'Bring Your Own Key' options, and certificates** in Salesforce.
- **Have access to a Salesforce account with the system permissions of Manage Encryption Keys and View Setup and Configuration enabled**, in order to handle tenant secrets. If you wish to disable Salesforce's key derivation process, and use your uploaded key material as the final data encryption key, then enable the **Allow BYOK to Opt Out of Key Derivation** option within the **Security > Platform Encryption > Advanced Settings** page in Salesforce. If you are using Salesforce cache-only keys, ensure your account also has **Customize Application** system permission enabled. Also ensure the **Allow Cache-Only Keys with BYOK** option within the **Security > Platform Encryption > Advanced Settings** page in Salesforce is also enabled. If you wish to enable the Replay Detection for cache-only keys, ensure the **Enable Replay Detection for Cache-Only Keys** option within the **Security > Platform Encryption > Advanced Settings** page in Salesforce is also enabled.

Enable and Assign System Permissions for CCKM Login

You must have the system permissions of **Manage Encryption Keys** and **View Setup and Configuration** enabled in Salesforce to allow CCKM users to log into CCKM. If you are using Salesforce cache-only keys, then you must also have the **Customize Application** system permission enabled in Salesforce to perform operations related to cache-only keys. After enabling these system permissions, assign these to the user account(s), which should these permission assignments.

For more information about managing permissions on encryption keys in Salesforce, see the “Permission Sets” documentation within the Salesforce online documentation.

To enable system permissions and assign these to user accounts:

1. Log in with the administrator privileges to your Salesforce account.
2. Click **Setup**.
3. In the left navigation bar, expand **Manage Users**.
4. Click **Permission Sets**.
Be sure that **Session Activation Required** is not selected.
5. Click **New**, and then select **New Permission Set**.
6. Click **System Permissions**, then click **Edit**.
7. Select the **Manage Encryption Keys** check box.
8. Select the **View Setup and Configuration** check box.
9. If you are using Salesforce cache-only keys, select the **Customize Application** check box.
10. Click **Save**.
11. Click **Manage Assignments** and then **Add Assignments**.
12. Select the user account(s), which should have these permission assignments.
13. Click **Assign**.
14. Click **Done**.

Salesforce Cache-only Keys

CCKM provides support for Salesforce's cache-only key service as part of its extended BYOK capability. With the use of a cache-only key, Salesforce calls the CCKM REST APIs to fetch a tenant secret (or cache-only key) from CCKM. This tenant secret is used for encrypt and decrypt operations in Salesforce. With the initial versions of Salesforce BYOK capability, the upload of the tenant secret and other key operations in Salesforce are initiated from CCKM using the available Salesforce Rest APIs.

In the API call from Salesforce to CCKM to fetch a tenant secret for the use of Salesforce's cache-only key service, a named credential is employed. The named credential is a combination of a URL of a CCKM callout endpoint along with the configured authentication parameters (CCKM username and password) associated with the URL. When CCKM receives the API call and successfully authenticates it, CCKM passes the tenant secret to Salesforce through a secure and authenticated connection. For more information about Salesforce cache-only keys, refer to the Salesforce online documentation.

CCKM also provides support for Replay Detection for cache-only keys. With this feature enabled, Salesforce inserts a unique marker called a RequestIdentifier (or request ID) in each callout to get the cache-only key from CCKM. The request ID serves as a random, one-time identifier for each valid callout request. Once this feature is enabled from the Salesforce portal, CCKM will accept and return the request IDs. No configuration is required on CCKM. Any request without Request ID or mismatched Request ID is aborted.

Notes

- CCKM currently supports CipherTrust Manager and DSM as the key sources for use with Salesforce's cache-only key service.
- In this document and in the context of support for Salesforce's cache-only key service, the term "tenant secret" is used interchangeably with "cache-only key".
- The CCKM cache-only feature is **not** tested on Salesforce Government Cloud Plus.

Configuring Cache-only Keys

The following is a summary of the steps to take to configure the use of cache-only keys within CCKM *and* Salesforce:

1. Configure CCKM with a SSL certificate that is signed by a Salesforce root Certificate Authority (CA) certificate. For more information, see ["Configure CCKM with SSL Certificate Signed by Salesforce Root CA Certificate" on the next page](#).
2. From the CCKM **Settings** page, configure the username and password to associate with the URL of the CCKM Rest API endpoint (or callout endpoint). For more information, see ["Configure Cache-only Key Endpoint" on page 239](#).
3. Set up a named credential in Salesforce for use with the cache-only keys. For more information, see ["Set up Named Credential in Salesforce" on the next page](#).
4. Generate a key in CipherTrust Manager or DSM for use with the cache-only keys. For more information, see ["Add \(Generate\) a Key" on page 236](#).
5. Configure a cache-only key in CCKM. For more information, see ["Create a Tenant Secret/Key" on page 225](#).
6. Verify that Salesforce can fetch the key from CCKM using the call-out connection. For more information, see ["Verify Call-Out Connection for Cache-only Key " on page 227](#).

Configure CCKM with SSL Certificate Signed by Salesforce Root CA Certificate

As part of cache-only key fetch process, Salesforce calls CCKM API over the HTTPS protocol, which requires CCKM to be configured with a valid SSL certificate. Salesforce cache-only key feature does not support self-signed certificates. CCKM must be configured with a SSL certificate that is signed by a Salesforce root Certificate Authority (CA) certificate. You can find the Salesforce root CA certificate list at <https://INSTANCE.salesforce.com/cacerts.jsp> where 'INSTANCE' is any upgraded instance. For example, <https://cs32.salesforce.com/cacerts.jsp>. You can use any one of these certificates from the list to sign the CCKM SSL certificate.

Note

To upload a valid SSL certificate to CCKM, run the following `system` command from the CCKM CLI:

```
system> security --setservercert.
```

For more information about the `system` category commands, see [Appendix A: "CCKM CLI Commands" on page 244](#). This appendix provides a high-level overview of each of the command categories that are available in the CCKM.

Set up Named Credential in Salesforce

Before setting up a named credential in Salesforce, ensure you have a CCKM deployed on a publicly accessible URL. In addition, ensure you have configured the username and password to be associated with the URL of the CCKM Rest API endpoint (or callout endpoint) in the **Endpoint for Cache-only Key** tab from the CCKM **Settings** page. (See ["Configure Cache-only Key Endpoint" on page 239](#) for information about the configuration of the callout endpoint.) The username and password are used to set up the named credential in Salesforce, which in turn provides Salesforce a method for fetching the cache-only key from CCKM through a secure and authenticated connection.

For detailed information about how to set up a named credential in Salesforce, see the Salesforce online documentation.

To set up a named credential in Salesforce:

1. Login into Salesforce.
2. Go to the **Setup** page.
3. In **Quick find** box, enter "Named Credentials" to search on this topic and then select **Named Credentials** in the results. The **Named Credentials Setup** page displays.
4. Select **New Named Credentials**.
5. In the **Label** box, enter the name of the named credential to use for CCKM. It is recommended you include "CCKM" in the name to reflect CCKM-associated information in the labels within Salesforce list views and reports. The name you provide in the **Label** box is automatically populated in the **Name** box.
6. In the **URL** box, enter the URL of the CCKM Rest API endpoint. To obtain the information to enter in this box, go to the **Endpoint for Cache-only Key** tab of the CCKM **Settings** page, and copy the URL from the **Call out URL** box and paste it into this box. Verify the hostname and port number are correct. For example, if CCKM URL is <https://thalescckm.com:8443/kmaas/> and the org ID is 00DB00007731234MAK, then the callout URL will be: <https://thalescckm.com:8443/kmaas/00DB00007731234MAK/>

7. Under **Authentication**, do the following:
 - a. Leave the **Certificate** box blank.
 - b. In the **Identity Type** drop-down menu, select **Named Principal**.

Note: Named Principal is currently the only option supported for the Salesforce identify type within CCKM.

- c. In the **Authentication protocol** drop-down menu, select **Password Authentication**.

Note: Password Authentication is currently the only option supported for authentication protocol within CCKM.

- d. In the **Username** box, enter the username to associate with the URL of the CCKM Rest API endpoint. This username must match the username configured in the **Endpoint for Cache-only Key** tab of the CCKM **Settings** page.
 - e. In the **Password** box, enter the password to associate with the URL of the CCKM Rest API endpoint. This password must match the password configured in the **Endpoint for Cache-only Key** tab of the CCKM **Settings** page.
 - f. Under the **Callout Options**, ensure the **Generate Authorization Header** check box is selected. This check box is selected by default.
 - g. Click **Save**.

Access CCKM for Salesforce

For details about accessing CCKM as a CCKM user for the CCKM service for Salesforce Key Management, see [Chapter 3: "About CCKM Admin and User Portals" on page 27](#) on page 1. This chapter includes information on the ways in which to access the CCKM service for Key Management.

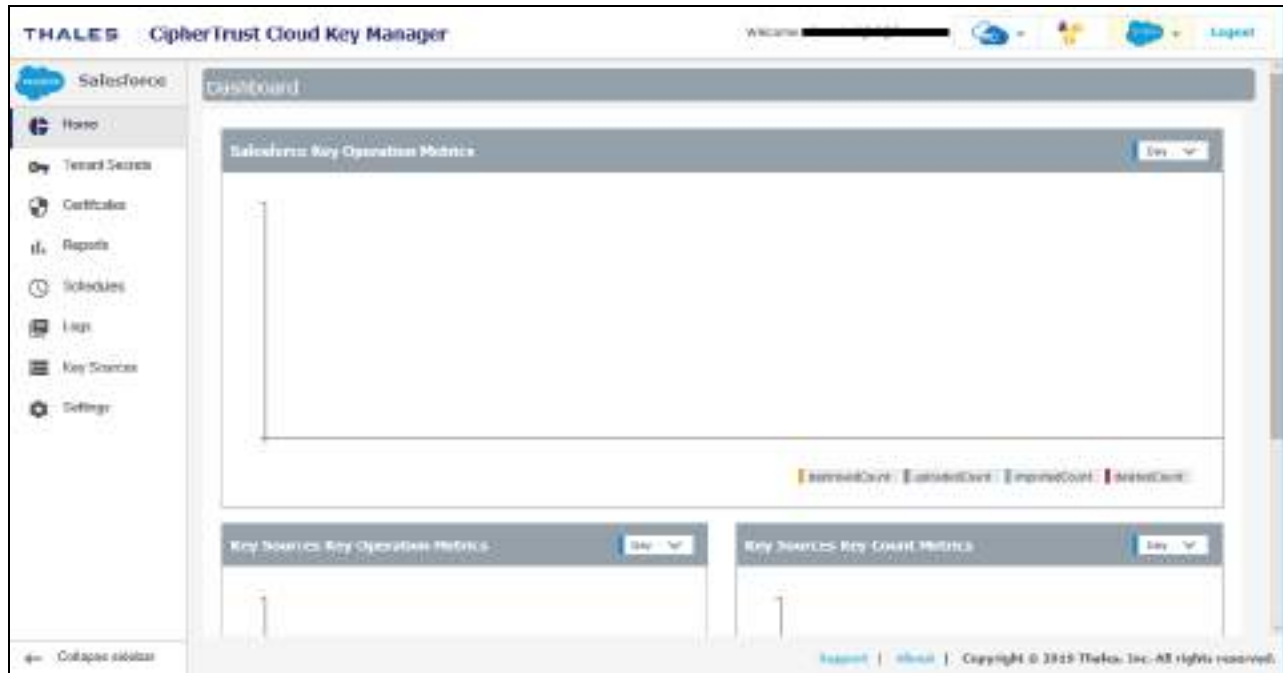
Salesforce CCKM User Portal

The CCKM User Portal for Salesforce contains the following menu options in the left-hand navigation bar:

- **Home:** Home page displays key operation and count metrics
- **Tenant Secrets:** Salesforce terminology for “keys”. The **All Tenant Secrets** page displays key management overview and provides the Upload and Synchronize functionality, and the Create, Destroy, Import, Delete Backup key operations available in CCKM.
- **Certificates:** The **All Certificates** page displays certificate management overview, plus Synchronize functionality.
- **Reports:** **All Reports** overview page links to detailed individual reports.
- **Schedule:** In **Schedule**, you can schedule automatic key rotation and key synchronization.
- **Logs:** **Logs** page lists individual logs that can be searched, sorted, and viewed.
- **Key Sources:** In **Key Sources**, you can create tenant secrets in CCKM to upload to Salesforce. CCKM provides support for the use of either CipherTrust Manager or DSM as a key source to generate, store, and retrieve the encryption keys that CCKM uses and manages.

Note: CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

- **Settings:** The **Settings** page allows you to add the email addresses of the recipients who are to receive email notifications and configure the settings for email notifications, syslog servers, and proxy servers. You can also view the release number of the CCKM running from this page.



To display the context of which Key Management service you are currently logged into within CCKM, the icon associated with the Key Management service displays at the top of the left-hand navigation bar.

The arrow button at the bottom of the navigation bar allows you to expand the bar to view the names of the menu options and collapse it to view only the icons associated with the menu options. Click the arrow button to either expand or collapse the navigation bar.

On the top-right side of the CCKM portal for , the following options are available:

- **Azure Cloud Service Context Switcher:** This context switcher allows you to login into another Azure cloud service different from the current Azure cloud service into which you are currently logged. If you select the Azure cloud in which you are currently logged, the Multi-account Login/Logout window for the Azure account(s) displays. For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **AWS Login:** If you are not already logged into AWS, clicking on the AWS Login icon brings you to the AWS Password Login page where you can enter your AWS credentials to access the CCKM AWS cloud service (from the current cloud service in which you are currently logged). If you are already logged into AWS, clicking on the AWS Login icon brings you to the Multi-account Login/Logout window for the AWS account(s). For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **Google Cloud Service Context Switcher:** This context switcher allows you to login into another Google Cloud organization different from the current Google Cloud organization into which you are currently logged. If you select the Google Cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Google account(s) displays. For more information about the **Multi-account Login/Logout** window, see ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).

- **Salesforce Cloud Service Context Switcher:** This context switcher allows you to login into another Salesforce cloud service different from the current Salesforce cloud service into which you are currently logged. This is Salesforce, Salesforce Government, or Salesforce Sandbox. If you select the Salesforce cloud in which you are currently logged, the Multi-account Login/Logout window for the Salesforce account(s) displays. For more information about the Multi-account Login/Logout window, refer to ["Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 32](#).
- **Logout:** Logout from CCKM.

Home

The **Home** page provides a dashboard to display key operation and count metrics.

Tenant Secrets

The **Tenant Secrets** page allows Cloud administrators to view all the keys and manage them within the boundary of granted permissions. Those keys that do not have List permission for the respective Cloud administrator will not be listed.

Within the **Tenant Secrets** page, the following types of Salesforce tenant secrets are available to specify the kind of data to encrypt using a tenant secret:

- Data in Salesforce: Encrypts data using the probabilistic encryption scheme.
- Data in Salesforce (Deterministic): Encrypts data using the deterministic encryption scheme.
- Search Index: Encrypts search index files.
- Analytics: Encrypts Einstein Analytics data.
- Event Bus (Developer Preview): Encrypts data changes and the associated change event.

You specify the key type on the **Tenant Secrets** page when creating a new key using the **Create Tenant Secret** dialog box or uploading an existing key using the **Upload a Tenant Secret** dialog box. The existing key is one you previously created from the **Key Sources** page. For more information about the types of tenant secrets, see the appropriate Salesforce online documentation.

The option to opt out of using the Salesforce key derivation mode when uploading tenant secrets from the **Tenants Secret** page is available in support of this Salesforce feature. The key derivation mode in CCKM allows you to derive a data encryption key based on the tenant secret you create or upload to Salesforce from the **Tenants Secret** page. Opting out of this mode means the tenant secret that you create or upload will be the key used to encrypt and decrypt your data.

From the **Tenant Secrets** page, a **Filter by** box is also available allowing you to filter on the types of tenant secrets to display in the list of tenant secrets.

A **Search** box is available at the top left of the **Tenant Secrets** page allowing for a search of a specific key by entering any one of the following parameters:

- Key Name
- Key ID
- Version
- Status
- Generated by
- Key Material Origin

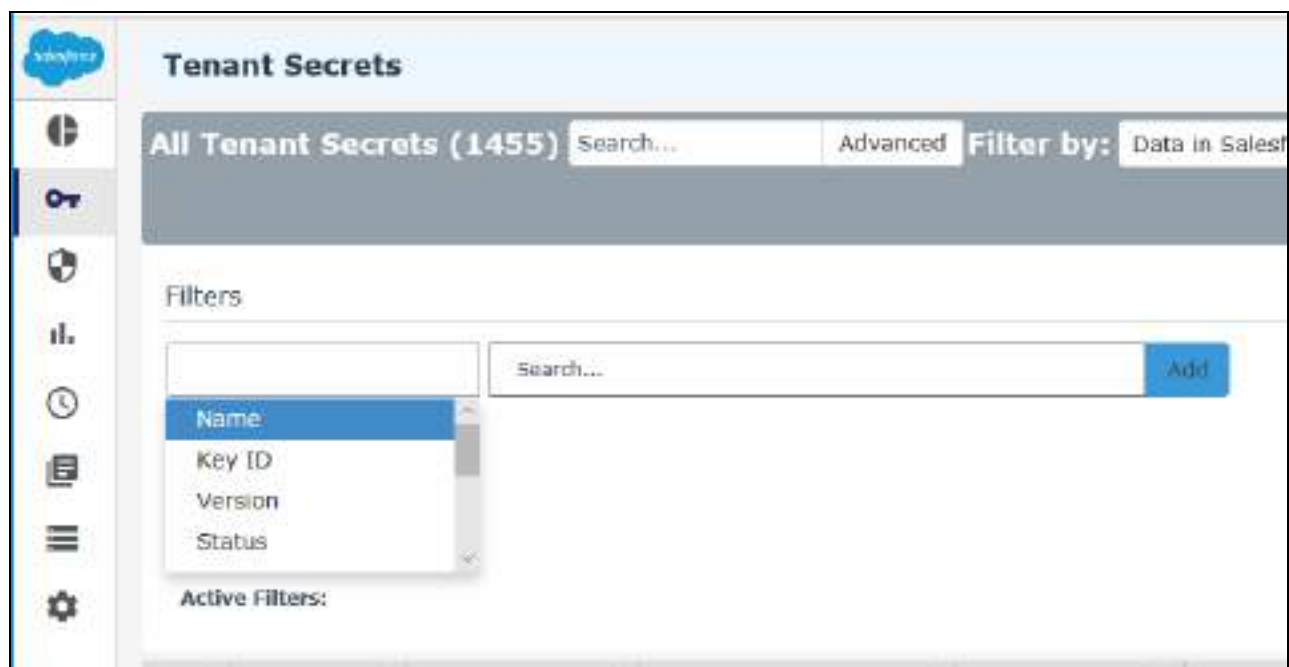
The **Advanced** tab allows you to filter your search on each of the columns displayed on the **Tenant Secrets** page with the exception of the **Action** column:

- Key Name
- Key ID
- Version—Min and Max
- Status—Available options from the drop-down menu are ACTIVE, ARCHIVED, and DESTROYED.
- Generated By
- Generated On—Enter START DATE and END DATE.
- Key Derivation
- Backup—Available options from the drop-down menu are Backed Up and Not Backed UP.
- Uploaded—Available options from the drop-down menu are Uploaded and Not Uploaded
- Key Material Origin

In addition, you can also search on **Key ID**. A **Key ID** for a given key is only available in the **Tenant Secret** box. To view the **Key ID** of a key, from the **Tenant Secrets** page, select the name of the key listed under the **Name** column. The **Tenant Secret** box displays and includes the name of the key, the Salesforce Key ID, key version, description of the key, and other properties.

Each of these columns are described in more detail below in this section.

To select to filter based on a specific column or Key ID from the **Tenant Secrets** page, click **Advanced**. The **Filters** section displays. From the filter drop-down menu, select the name of the column on which to filter your search or Key ID. For example, select **Key Name**. For the **Status**, **Backup**, and **Uploaded** columns, a drop-down menu is available from which to select a value for the search. Click the down arrow, select a value from the available options, and then click **Add**. For the **Version** column, enter a minimum and maximum version number for the search and then click **Add**. For the **Generated On** column, enter a start and end date for the search and then click **Add**. The specified search filter is then added to **Active Filters**. In the example, **Key Name: MK-createkey** is added to **Active Filters**. The results of the search displays in the **Tenant Secrets** table. You can narrow your search by selecting more columns to add to the filter. To clear the filtered results, from **Active Filters**, click on the “X” (or the **Delete** button) for each of the named filters.



At the top right of the **Tenant Secrets** page, the following buttons are available:

- **Export Secrets:** Allows you to export the contents of the Tenants Secrets list into a report in a CSV format. For information on how to export your tenant secrets, see ["Export Tenant Secrets" on page 230](#).
- **New Key:** Allows you to upload an external key (BYOK) or create a native key. For information on how to upload an external key or create a native key, see ["Create a Tenant Secret/Key" on the next page](#).
- **Synchronize:** Allows you to download any keys that were created in Salesforce, outside of the CCKM portal, into CCKM. For information on how to synchronize your keys, see ["Synchronize Tenant Secrets" on page 230](#).

The following columns display in the **Tenant Secrets** page:

- **Name:** The name of the key.
- **Version:** The version number of the key.
- **Status:**
 - **ACTIVE**
 - **DESTROYED**
 - **ARCHIVED**
- **Generated By:** The username associated with generating the key.
- **Generated On:** The date on which the key was generated.
- **Key Derivation:** Indicates whether the tenant secret is in key derivation mode. A green check icon indicates that it is in this mode.
- **Backup:** Indicates whether a backup of the key exists in CCKM. The status can be one of the following:
 - A green check icon indicates the backup exists in CCKM.
 - A red exclamation icon indicates the backup key does not exist in CCKM.
- **Uploaded:** Indicates whether the key is uploaded to Salesforce. A green check icon indicates that it is uploaded. Cache-only keys that are uploaded to Salesforce also display with the green check icon.
- **Key Material Origin:**
 - If the key was uploaded from CCKM, then the value is **Internal(<source_key_name>)**. For a cache-only key, the value is **FETCHED(<source_key_name>)**.
 - If the key was not uploaded from CCKM as a BYOK key, then the value is **EXTERNAL**.
- **Actions:** Allows you to select any of the following actions on a given key:
 - **Destroy**
Note: For a cache-only key, **Destroy** is the only action available within the **Actions** column. After the key is destroyed, the **Actions** column displays **Edit Key**.
 - **Edit key**

Note: This action is only applicable to a cache-only key.
 - **Import**
 - **Delete Backup**

In this section, create, upload, destroy, import, synchronize, and manage tenant secrets, as well as delete a backup key are described.

Create a Tenant Secret/Key

You have three options to create a new tenant secret or key:

- Upload an external key (BYOK) (see ["Upload your Own Tenant Secret" below](#))
- Create a native key (see ["Create a Tenant Secret/Key" on the facing page](#))
- Create cache-only key (see ["Create a Cache-only Key" on the facing page](#))

Upload your Own Tenant Secret

Salesforce supports only one active tenant secret per organization/OrgID. When you upload your own tenant secret, that key becomes active and the previously active key is automatically archived. Use Upload as part of your organization's key rotation policy. You can only upload a tenant secret once every 24 hours, as per Salesforce's policy.

The uploaded key is displayed in the Tenant Secrets list. As part of the upload process, Salesforce generates a corresponding backup key, which CCKM downloads and stores in the key source you are using (CipherTrust Manager or DSM). For more information about the backup key, see ["Backup Key" on page 231](#).

Note

A valid certificate is required, as described in ["Prerequisites on Salesforce" on page 216](#). If no certificate is listed in the certificate drop-down, you must go to Salesforce, create an appropriate certificate, and re-synchronize with CCKM.

To upload your BYOK tenant secret to CCKM:

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**. The **Tenant Secrets** list page displays.
2. From the **Tenant Secrets** list page, within the **New Key** box, select **Upload**. The **Upload a Tenant Secret** dialog box displays.
3. From the **Type** box, select the type of tenant secret to upload. The following are the options (in the order of selection):
 - Data in Salesforce
 - Search Index
 - Data in Salesforce (Deterministic)
 - Analytics
 - Event Bus
4. From the **Tenant Secret** drop-down menu, select the tenant secret to upload. Enter the first three characters of the name of the tenant secret to bring it up for selection.
5. Select whether to apply the Salesforce key derivation mode to the tenant secret. If you choose to apply it, ensure that the **Use Salesforce key derivation** check box is selected. By default, the check box is selected. Otherwise, deselect the check box to opt out of applying the key derivation mode to the tenant secret.
6. From the **Certificate** drop-down menu, select a valid certificate (used to wrap the key for secure upload).
7. Click **Upload to Salesforce** to upload the tenant secret to Salesforce.

Create a Tenant Secret/Key

Create a native key in Salesforce using CCKM. After CCKM creates a native key, it displays the key as **EXTERNAL** under the **Key Material Origin** column within the **Tenant Secrets > All Tenant Secrets** list page in CCKM.

As part of creating a native key, Salesforce generates a corresponding backup key, which CCKM downloads and stores in the key source you are using (CipherTrust Manager or DSM).

To create a native tenant secret:

1. From the **Tenant Secrets** list page, within **New Key** box, select **Create**.
2. From the **Type** box of the **Create a Tenant Secret** dialog box, select the type of tenant secret to create.
3. Select whether to apply the Salesforce key derivation mode to the tenant secret. If you choose to apply it, ensure that the **Salesforce key derivation** check box is selected. By default, the check box is selected. Otherwise, deselect the check box to opt out of applying the key derivation mode to the tenant secret.
4. Click **Create** to create the new tenant secret.

Create a Cache-only Key

In CCKM, create a tenant secret or cache-only key for use with Salesforce's cache-only key service. In creating this key, select the key you generated from your key source (CipherTrust Manager or DSM). Also, select a named credential that matches the named credential you previously defined in Salesforce. (For more information, see ["Add \(Generate\) a Key" on page 236](#).) After this key is created, the name of the key is displayed in the **Name** column, a green checkmark icon is displayed in the **Uploaded** column, and a status of **FETCHED(<key_name>)** is displayed in the **Key Material Origin** column of the **Tenant Secrets** page. In addition, a log message in the **Logs** page is recorded and displayed indicating a call-out connection to Salesforce has been created.

In order for Salesforce to use this key for its cache-only key service, Salesforce must be able to reach the endpoint (specified by the named credential) using the call-out connection that is created after you create the cache-only key in CCKM. If Salesforce can reach the endpoint (meaning the call-out connection between CCKM and Salesforce was set up successfully), then the key specified for the Unique Key Identifier is fetched from CCKM and becomes the active key within Salesforce. After performing the steps in this section, verify that Salesforce can reach the endpoint. See ["Verify Call-Out Connection for Cache-only Key " on the next page](#) for more information.

Note

CCKM currently supports CipherTrust Manager and DSM as the key sources for use with Salesforce's cache-only key service.

To create a cache-only key:

1. From the **Tenant Secrets** list page, within **New Key** box, select **Cache Only**. The **Create cache-only key** dialog box displays.
2. From the **Type** box, select the type of cache-only key to create. The following are the options (in the order of selection):
 - Data in Salesforce
 - Search Index
 - Data in Salesforce (Deterministic)
 - Analytics
 - Event Bus
3. From the **Tenant Secret** drop-down menu, select the key you generated using CipherTrust Manager or DSM depending your key source. Enter the first three characters of the name of the key to bring it up for selection.

4. In the **Certificate** box, select a valid certificate (used to wrap the key for secure upload).
5. In the **Named Credentials**, select the named credential to associate with this cache-only key.
6. Click **Create cache-only key** to create the new cache-only key.

Verify Call-Out Connection for Cache-only Key

After creating the cache-only key, verify that Salesforce can fetch the key from CCKM using the call-out connection. When the CCKM receives the API call from Salesforce and successfully authenticates it, CCKM passes the tenant secret to Salesforce through the call-out connection.

To verify that Salesforce can fetch the key from CCKM using the call-out connection:

1. Login to Salesforce.
2. Click **Setup** and from the Search box of the Setup page, enter **Platform Encryption**.
3. Select **Key Management** and the Key Management page displays. You will see a list of keys.
4. From the list of keys, find the row that contains the cache-only key you just created (using the version number to identify it) and then click **Details** button next to it. The **Cache-Only Key: Callout Check** page displays.
5. Click the **Check** button to check the connection for your cache-only key callout. A callout connection starts to test the connection. Details about the results displays on the page.
6. Under the **Callout Collection Details**, verify that the values for **Unique Key identifier**, **Certificate Unique Name**, and **Named Credential** match the values selected during the creation of cache-only key.
7. Verify that the message in the results pane displays the following:
`"The callout was successful".`
8. Login to CCKM User Portal for Salesforce.
9. In the left navigation bar, select **Logs**. The **Logs** page displays.
10. Verify that two events named "Get_Cache_Only_Key" is listed on the page. (The first event indicates a Salesforce cache-only key call was made. The second indicates that the key was successfully read from your key source (CipherTrust Manager or DSM) and returned to Salesforce.

Destroy a Tenant Secret

On the **Tenant Secret** page, destroying a tenant secret permanently deletes it from Salesforce. However, a backup key will remain in the key source you are using (CipherTrust Manager or DSM). After deletion, two new actions are displayed in the Action column: *Import* and *Delete Backup*.

Note

You can destroy only an archived tenant secret. If you want to destroy the currently active tenant secret, first change its status to ARCHIVED by uploading a new tenant secret (see ["Create a Tenant Secret/Key" on page 225](#)).

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**.
2. Find the name of the tenant secret and click **Destroy** under the **Actions** column. The **Destroy Tenant Secret** dialog box displays with a warning asking you to confirm the key destruction by entering the supplied phrase.
3. Enter or copy/paste the confirmation phrase.

4. Click **Destroy**.

The **Status** column for this tenant secret shows **DESTROYED**, and *Import* and *Delete Backup* actions are listed under the **Actions** column.

Note: If you simultaneously deleted the backup key using the **check box** option, the **Action** column will be blank.

Destroy a Cache-only Key

On the **Tenant Secret** page, destroying a cache-only key removes it from the Salesforce cache. However, a backup of the key still exists in your key source (CipherTrust Manager or DSM). After the key is destroyed, the action of **Edit Key** is available under the **Action** column for the key. For more information about **Edit Key**, see "[Edit a Cache-only Key](#)" below.

Note

CCKM currently only supports CipherTrust Manager and DSM as the key sources for use with Salesforce's cache-only key service.

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**.
2. Find the name of the key and click **Destroy**. The **Destroy Tenant Secret** dialog box displays with a warning asking you to confirm the key destruction by entering the supplied phrase.
3. Enter or copy/paste the confirmation phrase.
4. Click **Destroy**.

The Status column for this key shows **DESTROYED**.

Edit a Cache-only Key

If a certificate configured for use with a cache-only key has expired or you want to use another named credential that is configured with another CCKM hostname in Salesforce other than the one currently in use, edit the cache-only key configuration within the **Tenant Secret** page. You also have the option to edit a cache-only key that has a status of DESTROYED.

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**.
2. Find the name of the key and click **Edit Key**. The **Edit <key_name> Cache-only Key** dialog box displays.
3. To update the certificate replacing the current certificate with another one, in the **Certificate** box, select the certificate you wish to use.
4. To update the named credential replacing the current named credential with another one, in the **Named Credentials** box, select the named credential you wish to use.
5. Click **Update cache-only key**.

Import a Tenant Secret

In the event that someone deletes a Salesforce tenant secret directly in Salesforce, it is possible to restore it from the CCKM as follows:

Notes

- The Import action is available for tenant secrets with the status DESTROYED.
- If anyone deletes the backup key from CCKM, it will not be possible to import/restore the key to Salesforce.

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**.
2. From the **Tenant Secrets** list, select **Import** in the **Action** column of a Destroyed key. The key and metadata are restored to Salesforce. The Success message is displayed and the key is listed at the top of the **All Tenant Secrets** page with the status **ARCHIVED**.

Rotate a Tenant Secret

CCKM allows for a manual or a scheduled rotation of a Salesforce tenant secret by the tenant secret type. When you rotate a tenant secret, that key becomes active and the previously active key is automatically archived. Scheduling a rotation of a key is adding a new version of the key at a scheduled time. Scheduling the rotation of a tenant secret requires that you first configure this feature from the **Schedules** page prior to enabling it using the **Auto-rotate** button at the top of the **Tenant Secrets** page.

Notes

- If you provide your Salesforce user password when configuring the scheduled rotation of a tenant secret, which is an optional step, the password is saved to the key source you are using (CipherTrust Manager or DSM).
- The **Auto-rotate** button is set to **On** or **Off** depending on the key's autorotation status. If the key is scheduled for autorotation, this button is set to **On**. Otherwise, it is set to **Off** indicating the key is not scheduled for autorotation. The **Auto-rotate Tenant Secret for Type** dialog box displays when you click on the **Auto-rotate** button.

Manually Rotate a Tenant Secret (Add a New Version of a Tenant Secret)

Creating and uploading a tenant secret in CCKM is equivalent to manually rotating a tenant secret in CCKM.

Enable a Scheduled Tenant Secret Rotation

Enabling a scheduled rotation of a tenant secret requires that you first schedule to rotation in the **Schedules** page. Be sure to schedule the rotation of a tenant secret prior to enabling this feature.

During the scheduled date and time for the rotation, the tenant secret is rotated. The new tenant secret becomes active and the previously active tenant secret is automatically archived.

Note

A tenant secret that is a cache-only key remains as a cache-only key after rotation.

To enable a scheduled rotation of a tenant secret:

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**. The **Tenant Secrets** list page displays.
2. From the **Tenant Secrets** list page, within the **Filter by** box, select the type of tenant secret to enable for a scheduled rotation. The following are the options:
 - Data in Salesforce
 - Search Index
 - Data in Salesforce (Deterministic)
 - Analytics
 - Event Bus
3. Click the **Auto-rotate** button. The **Auto-rotate Tenant Secret for Type** dialog box displays.
4. For **Enable autorotation on this type?**, select **On** to enable the autorotation of this tenant-secret type. By default, this toggle is initially set to **Off**.
5. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
6. From the **Key Provider** box, select one of the following as the key source to which to add a new version:
 - **DSM**—Key is to be generated from DSM.
 - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager.
7. (Optional) Select whether to apply the Salesforce key derivation mode to the tenant secret. If you choose to apply it, ensure that the **Use Salesforce key derivation** check box is selected. By default, the check box is selected. Otherwise, deselect the check box to opt out of applying the key derivation mode to the tenant secret.
8. From the **Certificate** drop-down menu, select a valid certificate (used to wrap the key for secure upload).
9. Click **Submit**.

Synchronize Tenant Secrets

Synchronization downloads tenant secrets that were directly created in the Salesforce portal (not in CCKM). It does not upload anything from CCKM to Salesforce. Periodically synchronize CCKM with Salesforce to get any newly created tenant secrets from Salesforce.

To synchronize tenant secrets:

1. In the left navigation bar, click **Tenant Secrets**.
2. Click **Synchronize**.

Any new tenant secrets created in Salesforce are retrieved and displayed on the **Tenant Secrets** list.

Export Tenant Secrets

Export Secrets allows you to export the contents of the Tenant Secrets list into a report in a CSV format.

To export your Tenant Secrets list:

1. In the left navigation bar, click **Tenant Secrets**.
2. Click **Export Secrets**.

This will download all of the tenant secrets in a report in a CSV format.

Backup Key

When a tenant secret is uploaded to Salesforce from CCKM, Salesforce creates a backup tenant secret (or key) as an encrypted blob, which contains information about the Salesforce tenant secret. The backup key can be used to restore a tenant secret that has been deleted in Salesforce. After Salesforce creates a backup key, CCKM automatically downloads and stores the tenant secret in the key source you are using (CipherTrust Manager or DSM). In addition, when Salesforce tenant secrets are synchronized, the backup key of any tenant secret that was created directly in the Salesforce portal is also downloaded and stored in the key source you are using.

Delete a Backup Key

Delete a backup key when you definitely do not want the key and have no corresponding data that will ever need decryption. For a tenant secret that has been deleted:

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**.
2. From the list, select **Delete Backup** from the **Action** column of a Destroyed tenant secret. The **Destroy Key** warning page is displayed.
3. Enter or copy/paste the confirmation phrase.
4. Click **Delete**.

The key is removed from CCKM.

Certificates

The **All Certificates** list page allows you to view all Salesforce certificates, including Name, Type, Key Size and Expiration Date. Use the **Synchronize** button to download newly created certificates from Salesforce.

Note

See ["Prerequisites on Salesforce" on page 216](#) for the minimum required parameters for creating Salesforce certificates compatible with CCKM.

Synchronize Certificates

Synchronization downloads certificates that were directly created in the Salesforce portal (not in CCKM). Periodically synchronize CCKM with Salesforce to get any newly created tenant certificates from Salesforce.

1. In the left navigation bar, click **Certificates**.
2. Click **Synchronize**.

Any new certificates created in Salesforce are retrieved and appear in the UI.

Reports

Before the Salesforce keys can be represented in the CCKM reports, Event monitoring and Event log files must be activated in Salesforce. In addition, the log files validity must be set to 30 days in Salesforce.

Note

Currently, there is no support for cache-only keys within CCKM reports.

The **All Reports** page displays links to the following detailed individual reports:

- **Key Activity Report**
Displays the key activity type (such as TS (tenant secret) stored or TS generated) performed by a specific user along with the associated key ID, key version number, key name, and time at which the activity took place.
- **Cloud Key Manager User Action Report**
Lists the users and their actions in creating, uploading, importing, destroying, and deleting keys in the CCKM portal.

For each report displayed on the **All Reports** page, the report name, its description, the email address of the user who had run it (Created By), and the date on which it was previously run (Last Run) is displayed. For a specific report you select to view, the date on which it was previously run (Last Run), the start and end dates selected for the report, the email address of the user who had run it (Created By), and the total number of records available in the report (Report Count) is displayed. In addition, an **Actions** column is displayed, which provides the **Delete Report** link to allow for the deletion of the selected report from CCKM.

Delete a Report

1. From **All Reports** page, select the specific report you wish to delete. The report displays.
2. From the **Actions** column, click **Delete Report**. The **Delete Current Report** dialog box with a warning that you are about to delete a current report run on a specified date and time.
3. Click **Delete**. The report is deleted from CCKM.

Generate a Report

1. In the CCKM portal, in the left navigation bar, select **Reports**.
2. Under **Report Name**, select the report you wish to run.
3. From **Date Range**, click **Last day** to run the report for the activities from the previous day or **Specific** to run the report using a date range. If you choose to run by a date range, enter the start and end dates in the format MM/DD/YY or select the dates from the calendar button at the right of the box.
4. Click **Run Report**. The report displays.
5. To download the report, click **Download** at the top right of the page and select the report format (PDF or CSV) to download.

Schedules

The **Schedules** page allows Cloud administrators to view and schedule the following job types:

- Key Rotation
- Key Synchronization

CCKM provides support for multiple schedules for each of the job types.

At the top right of the **Schedules** page, the **Add Schedule** button is available from which to add a scheduled job type.

Note

Currently, there is no support for cache-only keys within CCKM schedules.

Schedule Key Rotation

CCKM provides support for multiple schedules of the rotation (autorotation) of a Salesforce tenant secret. When you rotate a tenant secret, that key becomes active and the previously active key is automatically archived. Scheduled rotation is configured in the **Schedules** page. Note that if you provide your Salesforce user password when configuring a scheduled key rotation, the password is saved to the key source you are using (CipherTrust Manager or DSM).

During a scheduled rotation of a key, CCKM uses the associated username and password to rotate the key. If the password expired or changed, the new password must be reentered. Otherwise, the scheduled rotation will fail.

Also, during a scheduled rotation of a key, CCKM uses the associated wrapping certificate to rotate the key. If this certificate expired, it must be renewed. Otherwise, the scheduled rotation will fail.

Note

If you have added an email address of the recipient who is to receive email notifications from CCKM and set up an SMTP server to send email notifications from CCKM, then a notification is sent to the recipient when the wrapping certificate is detected as expired during autorotation. The notification indicates that the named certificate expired and includes a reminder to renew the certificate. For more information about adding recipients' email addresses for notifications, see ["Adding Recipients' Email Addresses \(for Notifications\)" on page 237](#). For more information about setting up an SMTP server, see ["Set SMTP Settings" on page 83](#).

An error about an expired wrapping certificate is automatically logged on the **Logs** page.

You can pause or delete a scheduled key rotation in the **Schedules** page. Note that when a key rotation schedule is deleted from CCKM, then the Salesforce user password is also deleted from the key source you are using (CipherTrust Manager or DSM).

Note

CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

Set up a Key Rotation Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The Schedules page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_ROTATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.

7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule). If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.

Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.

8. (Optional) Enter your Salesforce user password in the **User Password** box.
9. Click **Save**.

Pause a Key Rotation Schedule

To pause a key rotation schedule, select **Yes** for **Pause** within the **Add Schedule** dialog box and then click **Save**. The setting is saved. To resume the rotation schedule after pausing it, select **No** for **Pause** and then click **Save**. The setting is saved.

Delete a Key Rotation Schedule

To delete a key rotation schedule, select **Delete** from the **Actions** drop-down menu from the **Schedules** page. Note that the Salesforce user password associated with the given key rotation schedule is also deleted from the key source you are using (CipherTrust Manager or DSM).

Schedule Key Synchronization

CCKM allows for multiple schedules of the synchronization of Salesforce tenant secrets. Synchronizing downloads any keys that were created in Salesforce, outside of the CCKM portal, into your CCKM. Scheduled key synchronization requires that this feature be configured in the **Schedules** page. You can also pause or delete a scheduled key synchronization in the **Schedules** page. Note that when a key synchronization schedule is deleted from CCKM, then the Salesforce user password is also deleted from the key source you are using (CipherTrust Manager or DSM).

Note

CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

Set up a Key Synchronization Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_SYNCHRONIZATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.

7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule). If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.

Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.

7. Enter your Salesforce user password in the **User Password** box.
8. Click **Save**.

Logs

CCKM maintains logs for events, such as login, key generation, synchronizing of keys, calls from Salesforce to CCKM for cache-only keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. If Replay Detection for cache-only keys is enabled, then the request ID is included in logged events for cache-only keys. For each logged event listed on the **Logs** page, the following information is displayed:

- Event name
- Severity level
- Date and time of the event
- Event message
- User (name) associated with the event

You can search for a specific logged event from the **Logs** page using the **Search** box. Allowable filters on the search are event name, severity level, event message, and user (name).

Key Sources

The **Key Sources** page displays **all** of the keys that were generated from the key source you are using. If you are logged into multiple clouds within CCKM, this page displays all of the keys from these clouds.

Three tabs are available at the top of the **Key Sources** page:

- **CipherTrust Manager Key:** Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.
- **DSM Key:** Use this tab to use DSM as your key source. The keys (including cache-only keys) are generated and stored within DSM.
- **nShield Key:** This tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Salesforce.

Note

CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

Note

CCKM currently only supports CipherTrust Manager and DSM as the key sources for use with Salesforce's cache-only key service.

Note

The Salesforce term “tenant secrets” is used interchangeably with “keys” in CCKM. Manage tenant secrets from two places in the CCKM portal: the **Tenant Secrets** page and the **Key Sources** page.

Keys listed on the **Key Sources** page are listed by name, cloud ID, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description and actions you can take regarding that key.

A **Search** box is available at the top left of the **Key Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Cloud ID
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

Add (Generate) a Key

From the **Key Sources** page, you can add (or generate) keys in CCKM using CipherTrust Manager or DSM as a key source for *all* of the clouds into which you are currently logged.

Note

The key algorithm currently supported for a Salesforce key in CCKM is AES.

Add (Generate) a CipherTrust Manager Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **CipherTrust Manager Key** tab.
3. Select **Add Key > Salesforce Key**.
4. In the **Add CipherTrust Manager Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **Salesforce Key** and (key) **Algorithm** is fixed to **AES256**.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.
(Reset clears the screen before it is saved.)

Add (Generate) a DSM Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key** tab.
3. Select **Add Key > Salesforce Key**.
4. In the **Add DSM Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **Salesforce Key** and (key) **Algorithm** is fixed to **AES256**.
5. In the **Name** box, enter the name of the new key.

6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.
(Reset clears the screen before it is saved.)

Add (Generate) an nShield Key

The **nShield Key** tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Salesforce.

Delete a Key

Deleting a key from the **Key Sources** page deletes the key from CCKM. If you delete a key from the **Key Sources** page that has already been uploaded to Salesforce, the key will not be deleted from Salesforce.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Click **Delete** next to the target key. The **Delete Key** dialog box with a Warning displays asking you to confirm the key deletion by entering the supplied phrase.
3. Enter or copy/paste the confirmation phrase.
4. Click **Delete**.

Manage Keys in Key Sources List Page

In addition to adding and deleting keys, the **Key Sources** page allows you to sort the existing keys by Name, Key Type, Algorithm and Created Date.

Settings

To access the **Settings** page in CCKM, select **Settings** from the left-hand navigation bar. From the **Settings** page, you can:

- Enter the email addresses of the recipients who are to receive the email notifications.
- Enable email notifications to be sent out from CCKM when specific types of key actions are performed from the **Tenant Secrets** page. For example, when a key is uploaded to the cloud service. You can also enable CCKM to send reminder notifications for when to rotate a key.
Configure a remote syslog server to which to send syslog messages.
- Configure the username and password to associate with the CCKM Rest API endpoint (or callout endpoint) for use with cache-only keys.
- View information about the software release number of the CCKM you are running including the build and version number.

Adding Recipients' Email Addresses (for Notifications)

Use the **General** tab from the **Settings** page to enter the email addresses of the recipients who are to receive email notifications from CCKM. Enter the recipients' email addresses separated by a comma in the **Notification Recipient** box and then click **Save** to save setting.

Enabling Notifications

Using the **Keys Notification** tab from the **Settings** page, you can configure CCKM to send an email notification when a tenant secret is destroyed from Salesforce or imported or uploaded to Salesforce. You can also use this tab to configure CCKM to send a reminder notification for when to manually rotate a key along with the frequency (in days) at which a notification is sent. Add the email address(es) of the recipients in the **General** tab from the **Settings** page. By default, no notifications are sent.

To enable email notifications:

1. In the **Settings** page, select the **Keys Notification** tab.
2. Under **Notifications**, select **On** for each of the notifications you wish to enable:
 - Rotate Tenant Secret (reminder notification)
 - Destroy Tenant Secret
 - Import Tenant Secret
 - Upload Tenant Secret
3. If you selected to enable **Rotate Tenant Secret**, in the **Days** box, enter the number of days for the frequency a reminder notification is sent.
4. Click **Save**.

Configuring Syslog Server

Note

In this document, the terms "syslog server" and "log server" are synonymous.

The **Syslog** tab from the **Settings** page allows you to configure the remote syslog or log server to which to send CCKM log data using TCP, TLS, or UDP protocol as the transport protocol. If you use the TLS protocol, then you are required to upload the syslog server certificate to CCKM for the TLS connection.

The **Syslog** tab also allows you to delete your syslog server configuration so that log data are no longer sent to the server.

Prior to configuring the setting for a remote syslog server, ensure that you have installed and configured the syslog server including configuring the setting for the port number the transport protocol will use to connect to CCKM. This port number configuration must match the port configuration you set in the **Syslog** tab of the **Settings** page.

Note

Support for both Syslog and Splunk as a remote syslog server is provided in CCKM. However, the use of a Splunk cloud server as a remote syslog server is currently not supported in CCKM.

Configure a Remote Syslog Server

To configure a remote syslog server to which to send CCKM log data:

Note

The facility name of "LOCAL1" for the syslog server displays in the **Facility** box. This default value is not available for modification.

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. In the **Hostname** box, enter the hostname or IP address of the syslog server.
3. In the **Port** box, enter the port number the transport protocol uses to connect to the syslog server. Enter a value between 1 and 65535. The default value is 514.
4. In the **Protocol** drop-down menu, select **TCP**, **TLS**, or **UDP** for the transport protocol to use when sending the log data to the syslog server. The default value is **UDP**. If you select **TCP**, proceed to Step 6. If you select **TLS**, proceed to the next step. Otherwise, proceed to Step 7.
5. In the **Certificate** box, select **Choose File** and then select the syslog server certificate to upload to CCKM.
6. Click **Test Connection** to test the connection between CCKM and the remote syslog server. This button is only available for use when either **TCP** or **TLS** is the selected transport protocol.
7. In the **Log Format** drop-down menu, select **CEF** (Common Event Format), **Plain Text**, or **RFC5424** for the format of the log message to use when sending the log data. The default value is **RFC5424**.
8. (Optional) In the **Description** box, enter a description of the remote syslog server.
9. Click **Save**.

Delete Remote Syslog Server Configuration

To delete the configuration of a remote syslog server:

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. Click **Delete**. The **Delete Syslog** dialog box displays.

A warning message displays indicating that this action will delete the syslog settings from CCKM. The message also indicates to type the confirmation phrase (I wish to delete the syslog settings) in the text box if you still wish to delete the syslog settings.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Delete**.

Configure Cache-only Key Endpoint

Use the **Endpoint for Cache-only Key** tab from the **Settings** page to configure the CCKM username and password to associate with the URL of the CCKM Rest API endpoint (or callout endpoint) for use with cache-only keys.

Note

Note the call-out URL, username, and password when performing the steps in this section. These values are required when you create the Named credential in Salesforce. See ["Set up Named Credential in Salesforce"](#) on page 219 for more information.

To configure the username and password to associate with the URL of the CCKM Rest API endpoint:

1. In the **Settings** page, select the **Endpoint for Cache-only Key** tab. The **Cache-only Key Configuration** page displays.
2. The URL of the CCKM Rest API endpoint automatically displays in the **Call out URL** box.

Note: When defining the named credential for use with cache-only keys within Salesforce's **Named Credentials Setup** page, copy the URL from the **Call out URL** box and paste it into the **URL** box in **Named Credentials Setup** page. Verify that URL points to public facing CCKM instance. Verify hostname and port number is correct. For example, if CCKM URL is `https://thalescckm.com:8443/kmaas/` and the org ID is `00DB00007731234MAK`, then the callout URL will be: `https://thalescckm.com:8443/kmaas/00DB00007731234MAK/`

3. In the **Username** box, enter the username to associate with the URL of the CCKM Rest API endpoint. This username must match the username configured in Salesforce's **Named Credentials Setup** page.
4. In the **Password** box, enter the password to associate with the URL of the CCKM Rest API endpoint. This password must match the password configured in Salesforce's **Named Credentials Setup** page.
5. Click **Save Settings** to save the setting.

About

Use the **About** tab to view the release number of the CCKM you are running.

Chapter 18: Backup and Restore

Backup and Restore Recommendations	241
Backup and Restore for DSM	241
Backup and Restore for CipherTrust Manager	241
Backup and Restore for MongoDB	241

In the event of a software crash or system changes, a backup of the CCKM configuration data stored within the key source you are using (CipherTrust Manager or DSM) and MongoDB can be used to restore CCKM to have your most recent backup of your data. Data stored within your key source and MongoDB include usernames, encryption keys, key attributes, local CAs, external CAs, certificates, signatures, and syslog connections.

Backup and Restore Recommendations

The following are the recommendations for backing up and restoring the CCKM configuration data within the key source you are using and MongoDB:

- CCKM should be inactive when the backups are performed to avoid data corruption.
- Backups of the key source and MongoDB should be performed during the same period (ensuring that the same snapshot of the CCKM configuration is backed up in both).
- Daily backups of your key source and MongoDB are recommended.

Backup and Restore for DSM

See the *Data Security Manager (DSM) Administration Guide* for information on how to backup and restore the CCKM configuration data stored within DSM, in addition to how to configure DSM for high availability.

Backup and Restore for CipherTrust Manager

See the *CipherTrust Manager Administrator Guide* for information on how to backup and restore the CCKM configuration data stored within CipherTrust Manager.

Backup and Restore for MongoDB

For information on how to backup and restore the CCKM configuration data stored within MongoDB or MongoDB Atlas, refer to the appropriate online MongoDB documentation. Likewise, for information on how to backup and restore the CCKM configuration data stored within Percona Server for MongoDB, refer to the appropriate online Percona Server documentation.

Chapter 19: Troubleshooting

Troubleshooting nShield Configurations	242
Troubleshooting Logs Files for Thales Support Team	243
Restrictions	243

Troubleshooting nShield Configurations

This section describes a few of the common issues that may arise during the configuration of the connection from CCKM to nShield Connect HSM(s), and the associated RFS and Security World within the Settings page and the steps to take to attempt to resolve these issues.

For more information on troubleshooting the configuration for nShield Connect, and the associated RFS and Security World, refer to the *nShield Connect User Guide for Unix*.

Table 19-1: Basic Troubleshooting nShield Configurations

Issue	How to Resolve
You entered an incorrect IP address for an nShield Connect during "Configure nShield Connect (for Azure and AWS Clouds)" on page 75 procedure.	Verify the electronic serial number (ESN) of each nShield Connect. To verify this, ensure that the output of the "rfs-sync --setup" command indicates "Configuration successfully written". From the nShield 1 IP Address box of the Key Sources page > nShield tab, enter the correct IP address of the nShield Connect and then click Add to Security World again.
A network connection from CCKM to nShield Connect cannot be detected.	Verify the ESN of each nShield Connect. To verify this, ensure that the output of the "rfs-sync --setup" command indicates "Configuration successfully written". Also, ensure that the output of the "rfs-sync --update" command includes "Updated module ..." or "Finished synchronization".
You entered an incorrect IP address for the RFS computer during the "Configure nShield Connect (for Azure and AWS Clouds)" on page 75 procedure.	The output of the "rfs-sync --update" command does not include "Updated module ...". From the nShield RFS IP Address box of the Key Sources page > nShield tab, enter the correct IP address of the RFS computer and then click Add to Security World again.

Only after you have verified that the output of all commands (displayed on the right side of the nShield Info page) are correct should you proceed to saving the nShield Connect configurations during the Install and Configure nShield Connect as a Key Source procedure. Saving the configuration restarts the hard server (Security World Software server) on which the CCKM is installed and configured.

Troubleshooting Logs Files for Thales Support Team

Thales Support will request for you to download the debug logs from CCKM in order to assist you in troubleshooting your issue(s).

To download the debug logs, click on **Download Debug Logs** on the **Logs** page within the CCKM Admin Portal. You can also download the debug logs through the **upload** command within the CCKM **applog** commands.

Restrictions

The following restrictions apply to the current version of CCKM:

- The Azure key vault name and key name combined cannot exceed 64 characters.
- The DSM domain name prefix must **not** exceed 28 characters.

Appendix A: CCKM CLI Commands

CCKM CLI Navigation	244
Maintenance Commands	245
Network Commands	245
System Commands	246
User Commands	246
Appllog Commands	247

The CCKM Command Line Interface (CLI) enables you to configure the CCKM network and perform other system-level tasks using the CCKM CLI commands.

This appendix provides a high-level overview of each of the command categories that are available in the CCKM. For information about the details of each command and its options, run the command with the “--help” option within the CCKM CLI.

CCKM CLI Navigation

The following are the supported categories of CCKM CLI commands:

- Maintenance
- Network
- System
- User
- Appllog

As a CCKM CLI administrator, log on to the CLI, and then enter a command category by typing the category name at the command line prompt. For example, type `system` to enter the system category. While in the category, you can execute the commands for that category.

Enter the entire category name, command, or argument, or enter just enough characters to identify uniquely the category, command, or argument. You can use the <Tab> key to complete a category, command, or argument. Enter enough characters to uniquely identify a category, command, or argument, and then press the <Tab> key. The CLI will complete it for you.

For example, at the top level, enter `n` and press <Tab>, the CLI expands it to `network`.

Inside the network category, you can enter `d<Tab>` and it expands to `dns`. Type `n<Tab>` and it expands to `netstat`. Note that there are other commands in the network category that start with `n`, such as `nslookup`, and `ntpservice`. To ensure to expand to `netstat` command, enter `ne` to identify it uniquely.

Other supported CLI navigation methods are:

- Enter a question mark (?) to display the next command or argument that is expected. Think of it as a shorthand form of help.
- Enter “up” to return to the top level so that you can enter another category. You can enter another category only from the top level.
- Enter “exit” at any time to end the current CLI session.

Maintenance Commands

The CCKM maintenance commands enable you to upgrade the CCKM and reset the CCKM admin password. [Table A-1: "CCKM Maintenance Commands" below](#) lists the supported CCKM maintenance commands:

Table A-1: CCKM Maintenance Commands

Command	Description
resetwebadmin	Resets the CCKM administrator password to the default value of 'admin123'.
updatedbpassword	Updates the MongoDB password in CCKM configuration. Provide the new password to replace the current password stored in the CCKM conf file.
upgrade	Upgrades the CCKM. As part of the upgrade to CCKM, you have the option to make the upgrade package available for download from a remote host over HTTPS or SCP.
quit q up <ctrl-d>	Quits or returns to the previous menu.
Exit	Exits the application.

Network Commands

The CCKM network commands are used to set, modify, or delete IP addresses on the CCKM, set up DNS servers, and perform other network-related tasks. [Table A-2: "CCKM Network Commands" below](#) lists the supported CCKM network commands:

Table A-2: CCKM Network Commands

Command	Description
checkport	Checks port connection status of a host.
dns	Shows or configures the DNS settings for the CCKM.
ip	Shows the CCKM network interface settings.
netstat	Prints network connections.
nslookup	Queries Internet name servers.
ntpservice	Configures NTP service settings.
ping	Pings an IP address, host name, or FQDN.
route	Sets a static route.
service	Starts, stop, or restart a network service or gets the status of a network service.
set	Configures a network device interface.
setup	Sets up a network configuration.
show	Shows network device configuration.
traceroute	Traces route to an IP address or hostname.
quit q up <ctrl-d>	Quits or returns to the previous menu.
exit	Exits the application.

System Commands

The CCKM system commands enable you to configure the CCKM host settings (such as, date and time or timezone) and security features, reboot or shut down the CCKM, show current CCKM version information, and perform other system administration tasks. [Table A-3: "CCKM System Commands" below](#) lists the supported CCKM system commands:

Table A-3: CCKM System Commands

Command	Description
date	Prints or sets the system date and time.
host	Performs DNS lookups.
hostname	Sets or shows the CCKM hostname.
hosts	Updates the host file.
lastlogin	Shows the last logged-in date for each user on the system.
passwordrecovery	Manages the options for the recovery of the CLI admin user password. When this command is set to enable , you have the option to reset your CCKM CLI admin password (if required) using a password recovery utility. Note: Thales <i>highly recommends</i> that the passwordrecovery is set to enable to ensure you can easily reset your CLI admin password in case you lose or forget it. To reset your admin password, contact Thales Customer Support.
port	Configures the ports of the CCKM Admin Portal and User Portal.
reboot	Reboots the CCKM appliance.
security	Configures the CCKM security features including the certificates.
server	Manages the CCKM server.
shutdown	Stops the CCKM software and powers off the virtual appliance.
sshbanner	Configures, shows, and removes the current ssh banner that displays when you log in to the CCKM CLI. You can also set the current version of the ssh banner to the default version of "Welcome to CipherTrust Cloud Key Manager".
timezone	Sets or shows the system timezone.
uptime	Shows how long the CCKM has been running.
version	Shows the CCKM version information.
who	Shows information about the users who are currently logged into the CCKM appliance using CLI.
quit q up <ctrl-d>	Quits or returns to the previous menu.
exit	Exits the application.

User Commands

The CCKM user commands enable you to add, modify, delete CCKM CLI users, and display information about these users. [Table A-4: "CCKM User Commands" on the next page](#) lists the supported CCKM user commands:

Table A-4: CCKM User Commands

Command	Description
add	Adds a new CLI user.
delete	Deletes a CLI user.
modify	Changes the password of a CLI user.
show	Lists all of the configured CLI users.
quit q up <ctrl-d>	Quits or returns to the previous menu.
Exit	Exits the application.

Applog Commands

The CCKM applog commands enable you to upload debug logs from the CCKM to the remote servers using the secure copy protocol (SCP) to transfer the log files between the two hosts. [Table A-5: "CCKM Applog Commands" below](#) lists the supported CCKM applog commands:

Table A-5: CCKM Applog Commands

Command	Description
upload	Uploads logs to remote server using SCP.
quit q up <ctrl-d>	Quits or returns to the previous menu.
Exit	Exits the application.

Appendix B: How to Export Your DSM Server Certificate from DSM

Exporting your DSM Server Certificate 248

CCKM for a supported cloud performs key operations through REST APIs exposed in the key source (CipherTrust Manager, DSM, or nShield) you are using. If you are using DSM as a key source for CCKM, CCKM requires the DSM server certificate to invoke the APIs. You can download the certificate through the use of a browser pointing to the DSM. Download the file in a Base64 format. When configuring the DSM as your key source in the **Key Sources** page of the CCKM Admin Portal, you are required to upload the DSM certificate to CCKM.

This appendix describes how to export a DSM certificate file from DSM to your laptop or desktop using the Chrome Internet browser. This file is then available to upload to the CCKM. For more information, see ["Configure DSM as Key Source" on page 75](#).

Note

The steps to export a DSM server certificate using the Firefox and IE browsers may differ slightly from the steps for the Chrome browser.

Exporting your DSM Server Certificate

To export your DSM server certificate from DSM using the Chrome browser:

1. Enter "https://<DSM IP address>" in a browser. The DSM user login page is displayed.
2. Log on to the web-based management console of the DSM.
3. In the web URL, click on the **Not Secure** section and click **View Certificate**.
4. Click the **Details** tab of the certificate.
5. Click **Copy to File**. The **Certificate Export Wizard** dialog box displays. This wizard is used to export your certificate in the appropriate format.
6. In the **Export File Format** page, select the **Base-64 encoded X.509 (.CER)** format option.
7. Click **Next**.
8. In the **File to Export** page, enter the directory to which to save the file within your laptop or desktop and specify the file name (for example, "DSM.cer").
9. Click **Finish**.

Appendix C: Configure SSL/TLS Certificate Chain of Trust for CCKM

Create and Configure SSL/TLS Certificate Chain of Trust 249

The chain of trust of an SSL or TLS certificate is an ordered list of certificates. The following certificates are contained within the certificate chain of trust:

- Server Signed Certificate—signed certificate, which would be signed by Intermediate Certificate Authority (CA).
- Intermediate Certificate—Certificate of CA that certifies the server certificate.
- Root Certificate—Certificate of CA that certifies Intermediate CA certificate. This is the root CA.

Note that there can be more than one intermediate CA certificate forming a chain of trust.

The certificate chain of trust enables the receiver of the certificate to verify that the sender and all intermediate certificates are trustworthy and that the certificate is valid.

Using CCKM CLI, you have the ability to generate a Certificate Signing Request (CSR). When you wish to have your CCKM server certificate signed by an external CA, you must first generate a CSR and then send it to a CA. As part of the CSR generation, provide information in the CLI including the following:

- CCKM hostname
- Organizational unit, which is typically a department or group name.
- Organization, which is typically a company name.
- Two-letter country code, which is the abbreviation of the country where the organizational unit is located.

Thereafter, the CA takes this CSR and sends you a digitally signed certificate after it verifies the CSR is a valid.

Using CCKM CLI, you also have the ability to upload a certificate chain of trust for an SSL/TLS certificate, which is sent to the web browser when you open up a browser window to login into the CCKM Admin and User portals.

Create and Configure SSL/TLS Certificate Chain of Trust

To create a chain of trust certificate and then configure a new CCKM server certificate based on the new chain of trust certificate:

1. Log into the CCKM CLI to access CCKM CLI.
2. Generate the Certificate Signing Request using the following command:
3. When prompted, enter the requested information for the CSR ensuring that the information you provide is consistent with the requirements of the external CA you are using:

Please enter the following information for Certificate Signing Request generation.

```
The server hostname[cckm hostname]:
What is the name of your organizational unit? []:
What is the name of your organization? []:
What is the name of your City or Locality? []:
```



```
What is the name of your State or Province? []:  
What is your two-letter country code? [US]:  
What is your email address? []:
```

4. After the CSR is generated, save the contents of the CSR to a file in a directory within your laptop or desktop.
5. Submit the CSR to your external CA to obtain an approved and signed certificate.
6. Obtain the intermediate and root CA certificates from your external CA and save each certificate to your laptop or desktop.
7. Create a certificate chain of trust using the following `cat` command in a Bash shell outside of CCKM CLI. For example:

```
$cat Server.crt IntermediateCA.crt RootCA.crt >> chain_of_trust.crt.
```
8. At CCKM command prompt, enter the following commands to configure a new CCKM server certificate:

```
cckm> system  
system> security --setservercert
```
9. The following message displays:

```
WARNING: Setting a new server certificate operation will implicitly restart the  
application.  
Do you wish to continue? <y/n> [n]:y
```
10. Type `y` to continue.
11. You are prompted to enter the contents of your server certificate and then press **Enter** and **Ctrl+d**. (Copy the contents of the `chain_of_trust.crt` file (from Step 6), paste it into CLI, and then press **Enter** and **Ctrl+d** :

```
Enter server certificate content and press Enter and Ctrl+d:
```
12. After the `chain_of_trust.crt` is uploaded, CCKM restarts the CCKM Admin and User portals. After these portals are back online, open a new browser and login into the CCKM Admin Portal.
13. Verify that the certificate chain of trust for CCKM is valid:

Note

The following steps are based on the use of the Chrome Internet browser. The steps based on the use of the Firefox and IE browsers may differ slightly.

- a. In the web URL, click on the Padlock icon in the address bar, and click **Certificate**.
- b. Click the **General** tab to view that the certificate is valid.

Appendix D: Sample Python Scripts for Authenticate and Call CCKM REST API

cckm-azure.py	252
cckm-aws.py	254
cckm-google.py	255
cckm-salesforce.py	256

This Appendix demonstrates how to authenticate and call CCKM REST API using a Python script. After launching and configuring a CCKM instance, you can access CCKM REST API at <https://<your cckm hostname or ip address>:8443/kmaas/doc>. You can use your preferred tools or scripts to call CCKM REST API. In this appendix, a Python script is used to demonstrate these API calls.

To call CCKM REST API, the script must first authenticate to CCKM using your cloud credentials. The authentication call will differ slightly depending on the following:

- Cloud service you are using
- Whether you are using Azure service principal
- Whether you are using AWS long-term credentials or temporary credentials

After successful authentication, you will get a cookie JSESSIONID and start a new user session. Subsequent calls to CCKM REST API within the user session must include this cookie. In addition, all requests must have a valid "X-XSRF-TOKEN" header to prevent a Cross Site Request Forgery (CSRF) attack. The following is the authentication flow:

1. Call <https://<your cckm>:8443/kmaas>. Get the X-XSRF-TOKEN from the response header, and use it in the request header.
2. Call proper auth API, passing cloud credentials in the Content-Type "application/x-www-form-urlencoded".
3. If auth call succeeds, save the JSESSIONID cookie, and attach it for subsequent calls.
4. Call CCKM REST API, using Content-Type "application/json"
5. Call logout.

The following are the current limitations when using CCKM REST API:

1. If your Azure tenant uses federated authentication, or enable MFA, you cannot use your Azure user credentials in CCKM REST API. You will have to configure CCKM as a service principal and provide admin consent to grant permissions to the CCKM app to access your Azure resources. During the admin consent process, you will choose a tenant password. Thereafter your REST API client or script can use your tenant name and password to authenticate to CCKM REST API.
2. You can use the temporary credentials of the AWS Security Token Service (STS) to authenticate CCKM REST API. However, when you call the add schedule REST API, you have to pass long-term credentials. If you do not have long-term credentials, then create your schedule jobs outside of CCKM instead of using the CCKM schedules.

The sections that follow provide four sample Python scripts used to authenticate and call CCKM REST API. These scripts are used to manage the following:

- Azure keys
- AWS keys
- Google Cloud keys
- Salesforce keys

These scripts use a Python Requests module to handle the session cookie. To see how each script works, enter your cloud credentials and remove commented out calls.

cckm-azure.py

```
import requests
import json
import sys

requests.packages.urllib3.disable_warnings()
session = requests.Session()
session.verify = False
resp = session.get(cckm_url)

# For Azure user login.
# Replace with your host, username and password
cckm_url = "https://host:8443/kmaas/"
username = ""
password = ""

session.headers = {'X-XSRF-TOKEN': resp.headers['X-XSRF-TOKEN']}
data = {'username': username, 'password': password}
resp = session.post(cckm_url + "auth/azure", data=data)
if resp.status_code != 200:
    print resp.content
    sys.exit()

"""
# For Azure service principal login
# Replace your tenant name and tenant password
tenant = "mytenant.onmicrosoft.com"
password = ""

session.headers = {'X-XSRF-TOKEN': resp.headers['X-XSRF-TOKEN']}
data = {'tenant': tenant, 'password': password}
resp = session.post(cckm_url + "auth2/azure", data=data)
if resp.status_code != 200:
    print resp.content
    sys.exit()
"""

session.headers['Content-Type'] = 'application/json'

def call_api(method, api, data=None):
    if method == 'post':
        resp = session.post(cckm_url + api, data=json.dumps(data))
    elif method == 'put':
        resp = session.put(cckm_url + api, data=json.dumps(data))
```

```
        else:
            resp = session.get(cckm_url + api)
print "%s %s" % (method, api)
if data:
    print data
print json.dumps(resp.json(), indent=4)
print
return resp.json()

# Following are example REST API calls, uncomment needed lines

#call_api("get", "me")
#call_api("post", "rest/azureKeys/synchronize")
#call_api("get", "rest/azureKeyVaults")
#call_api("get", "rest/azureKeys")
#call_api("get", "rest/keyvaultkey")

"""
# Add a source key
name_of_new_source_key = "A-new-Azure-source-key"
data = {"name": name_of_new_source_key, "algorithm": "RSA2048", "service": "Azure",
"description": "a new key", "cckmId": username}
call_api("post", "rest/keyvaultkey", data)
"""

"""
# Upload the new source key to Azure
# First search source key by name and find its ID
source_key = call_api("get", "rest/keyvaultkey/keys?search=%s" % name_of_new_source_key)
source_key_id = source_key['content'][0]['id']

# Prepare Azure key, replace with your key vault and key names
azure_key_vault = "jimmy-kv-west-europe"
azure_key_name = "A00cl1-vceg"
data = {
    "keyVaultKeyId": source_key_id,
    "keyVaultName": azure_key_vault,
    "keyVaultDisplayName": azure_key_vault,
    "keyName": azure_key_name ,
    "attributes": {
        "enabled": True,
        "exp": 1546329600
    },
    "hsm": False,
    "key_ops": ["encrypt"],
    "tags": {
        "tag1": "test tag"
    }
}
call_api("post", "rest/azureKeys/upload", data)

# Find Azure key by name
azure_key = call_api("get", "rest/azureKeys/search/findByName?name=%s" % azure_key_name
)
"""
```

```
#call_api("get", "rest/azureAudits")

resp = session.post(cckm_url + "logout")
```

cckm-aws.py

```
import requests
import json
import sys

# Replace with your host, username and password
cckm_url = "https://host:8443/kmaas/"
aws_access_key_id = ""
aws_secret_access_key = ""
# To use AWS STS temporary credentials, provide session_token
aws_session_token = ""

requests.packages.urllib3.disable_warnings()
session = requests.Session()
session.verify = False
resp = session.get(cckm_url)
session.headers = {'X-XSRF-TOKEN': resp.headers['X-XSRF-TOKEN']}

def call_api(method, api, data=None):
    if method == 'post':
        resp = session.post(cckm_url + api, data=json.dumps(data))
    elif method == 'put':
        resp = session.put(cckm_url + api, data=json.dumps(data))
    else:
        resp = session.get(cckm_url + api)
    print "%s %s" % (method, api)
    if data:
        print data
    print json.dumps(resp.json(), indent=4)
    print
    return resp.json()

# Authenticate CCKM REST API with AWS credentials
if aws_session_token:
    data = {'username': aws_access_key_id , 'password': aws_secret_access_key,
'session_token' : aws_session_token}
else:
    data = {'username': aws_access_key_id , 'password': aws_secret_access_key }

resp = session.post(cckm_url + "login/aws", data=data)
print resp.status_code
if resp.status_code != 200:
    print resp.content
    sys.exit()

# Example CCKM REST API calls
session.headers['Content-Type'] = 'application/json'

# Example REST API calls, uncomment needed lines
```

```
#me = call_api("get", "me")
#call_api("post", "rest/awsKeys/synchronize", [ "us-east-1" ])
#call_api("get", "rest/awsKeys/aliases")

"""
# REST API calls for creating and uploading DSM key to AWS
# Add a source key in DSM
name_of_new_source_key = "A-new-AWS-source-key"
user_arn = me["aws"][0]["arn"]
data = {"name": name_of_new_source_key, "algorithm": "AES256", "service": "AWS",
"description": "a new key", "cckmId": user_arn, "provider": "DSM"}
call_api("post", "rest/keyvaultkey", data)

# Upload the new source key to AWS
# First search source key by name and find its ID
source_key = call_api("get", "rest/keyvaultkey/keys?search=%s" % name_of_new_source_key)
source_key_id = source_key['content'][0]['id']

# Prepare AWS key, replace with your key alias
aws_key_alias = "cckm-aws-test"
data = {
    "sourceKeyId": source_key_id,
    "aliasName": aws_key_alias,
    "region": "us-east-1",
    "tags": {
        "description": "Imported from CCKM source key " + name_of_new_source_key
    },
    "allowedExternalAccounts": []
}
call_api("post", "rest/awsKeys/upload", data)

# Find AWS key by alias
aws_key = call_api("get", "rest/awsKeys/aliases?aliasName=%s" % aws_key_alias )
"""

resp = session.post(cckm_url + "logout")
```

cckm-google.py

```
import requests
import json
import sys

# For Google Cloud service account login.
# Replace with your host, email and password
cckm_url = "https://localhost:8443/kmaas/"
email = ""
password = ""

requests.packages.urllib3.disable_warnings()
session = requests.Session()
session.verify = False
resp = session.get(cckm_url)

session.headers = {'X-XSRF-TOKEN': resp.headers['X-XSRF-TOKEN']}
```

```
data = {'email': email, 'password': password}
resp = session.post(cckm_url + "auth/googleCloud", data=data)
if resp.status_code != 200:
    print resp.content
    sys.exit()

session.headers['Content-Type'] = 'application/json'

def call_api(method, api, data=None):
    if method == 'post':
        resp = session.post(cckm_url + api, data=json.dumps(data))
    elif method == 'put':
        resp = session.put(cckm_url + api, data=json.dumps(data))
    else:
        resp = session.get(cckm_url + api)
    print "%s %s" % (method, api)
    if data:
        print data
    print json.dumps(resp.json(), indent=4)
    print
    return resp.json()

# Following are example REST API calls, uncomment needed lines

#call_api("get", "me")
#call_api("post", "rest/googleCloudKeys/synchronize", {"skipKeyRingSynchronization":
False})
#call_api("get", "rest/googleCloudKeys")

resp = session.post(cckm_url + "logout")
```

cckm-salesforce.py

```
import requests
import json
import sys

# Example REST API for Salesforce keys
# Replace with your Salesforce username and password
cckm_url = "https://host:8443/kmaas/"
username = ""
password = ""

requests.packages.urllib3.disable_warnings()
session = requests.Session()
session.verify = False
resp = session.get(cckm_url)

session.headers = {'X-XSRF-TOKEN': resp.headers['X-XSRF-TOKEN']}
data = {'username': username, 'password': password}
resp = session.post(cckm_url + "auth/salesforce", data=data)
if resp.status_code != 200:
    print resp.content
    sys.exit()
```

```
def call_api(method, api, data=None):
    if method == 'post':
        resp = session.post(cckm_url + api, data=json.dumps(data))
    elif method == 'put':
        resp = session.put(cckm_url + api, data=data)
    else:
        resp = session.get(cckm_url + api)
    print "%s %s" % (method, api)
    if data:
        print data
    print json.dumps(resp.json(), indent=4)
    print
    return resp.json()

session.headers['Content-Type'] = 'application/json'

# Example REST API calls, uncomment needed lines
#call_api("get", "me")
#call_api("post", "rest/salesforceKeys/synchronize")
#call_api("post", "rest/salesforceCertificates/download")
#call_api("get", "rest/salesforceKeys")

"""
# Generate a source key in KeySecure
name_of_new_source_key = "A-new-Salesforce-source-key4"
data = {"name": name_of_new_source_key, "algorithm": "AES256", "service": "Salesforce",
"description": "a new key for uploading to Salesforce", "cckmId": username, "provider":
"KeySecure"}
source_key = call_api("post", "rest/keyvaultkey", data)

# Upload the new source key to Salesforce
source_key_id = source_key['id']

# get a certificate for upload wrapping, assume the first certificate
certs = call_api("get", "rest/salesforceCertificates")
cert_label = certs['_embedded']['salesforceCertificates'][0]['name']

# Upload
data = {"certificateName": cert_label, "keyVaultKeyId": source_key_id,
"keyDerivationMode": "PBKDF2", "type": "Data"}
call_api("post", "rest/salesforceKeys/upload", data)
"""

resp = session.post(cckm_url + "logout")
```


Appendix E: Authenticate your REST API Client to CCKM REST APIs for Google Cloud Keys

This appendix shows how to authenticate your API client to the CCKM REST APIs using API calls before making calls to CCKM REST APIs to manage Google Cloud keys.

After submitting a POST call to authenticate to CCKM REST APIs, CCKM will first check your password hash against the password hash in its database. If it matches, CCKM will use the private key of the service account to obtain an access token from Google Cloud. Your REST API client is successfully authenticated to CCKM REST APIs when CCKM receives an access token from Google Cloud. At this point, you will receive a JSESSIONID cookie. Once you have this cookie, you can make calls to CCKM REST APIs to manage Google Cloud keys. You will need to add the JSESSIONID cookie and X-XSRF-TOKEN header to all subsequent REST API calls.

Refer to ["cckm-google.py" on page 255](#) for a sample Python script that demonstrates how to authenticate and call CCKM REST API to manage Google Cloud keys.

To authenticate your API client to the CCKM REST APIs using API calls:

1. Make a GET call:

```
GET https://<cckm hostname>:8443/kmaas
```

2. From the response header, copy the session x-xsrf-token.
3. Make a POST call (paste the session x-xsrf-token in the request header):

```
POST https://<cckm hostname>:8443/kmaas/auth/googleCloud
```

Request body:

```
{
  "email": "you service account email",
  "password": "your service account password",
}
```

Request headers:

```
Content-Type: application/x-www-form-urlencoded
X-XSRF-TOKEN: <your session x-xsrf-token>
```

Appendix F: General Guidelines for CCKM Appliance

This appendix provides the general guidelines for running the CCKM appliance with CipherTrust Manager and DSM.

Note

Each AWS account, Azure subscription, Google project, and IBM and Salesforce organization is considered one cloud unit.

The following are general guidelines for the CCKM Appliance:

- The number of cloud units must **not** exceed the number of CipherTrust Manager or DSM domains:
 - CipherTrust Manager has a limit of 1,000 domains.
 - DSM Enterprise has a limit of 1,000 domains.
 - DSM 25 has a limit of 2 domains.
 - DSM 10 has a limit of 2 domains.
- CipherTrust Manager has a limit of 1,000,000 keys in the CipherTrust Manager k470 and k570 appliance models.
- CipherTrust Manager has a limit of 25,000 keys in the Virtual CipherTrust Manager k470v model.

Note

The use DSM 10 or DSM 25 with CCKM is not supported due to the domain limitations.

Note

The size of your Mongo DB storage may limit the number of keys in CCKM.



Contact us

For office locations and contact information,
visit cpl.thalesgroup.com/contact-us

cpl.thalesgroup.com





CipherTrust Integrations

4 February 2026

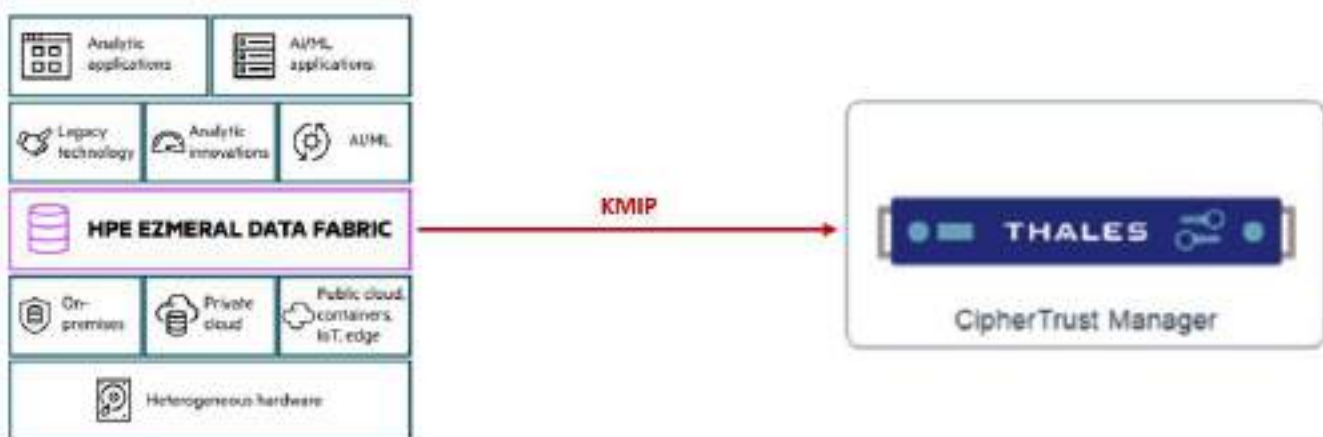
Contents

- HPE Ezmeral Data Fabric 3
 - Integration with CipherTrust Manager 3
 - Verifying Your Integration 16
 - Licensing 16
 - Appendix..... 17

HPE Ezmeral Data Fabric

HPE Ezmeral Data Fabric (EDF) provides a platform that spans core, cloud, and edge environments to ingest any data type or source while handling hundreds of petabytes and thousands of users with ease. It simplifies data management with a single infrastructure that automatically scales as data is ingested without impacting service-level objectives, security, or data resilience. Data on disk (or data-at-rest) on a secure data-fabric cluster can be encrypted, allowing you to protect the data if a disk is compromised.

Encryption of data-at-rest not only prevents unauthorized users from accessing sensitive data, but it also protects against data-theft through sector-level disk access. For encrypted volumes, data-fabric automatically encrypts data-at-rest and manages the keys used to encrypt data seamlessly. You do not need special utilities to encrypt or decrypt the data. Data Fabric uses AES256/XTS to protect data on the disk.



Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

CipherTrust Manager

- CipherTrust Manager 2.7 and higher

HPE EDF Version

- HPE Ezmeral Data Fabric 7.0 and higher

Integration with CipherTrust Manager

This section lists the steps to integrate HPE EDF with CipherTrust Manager.

-
- Prerequisites
 - Configuration on CipherTrust Manager
 - Configuration on HPE EDF

Prerequisites

This section provides the prerequisites for integration of HPE EDF with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager documentation](#).
- HPE EDF communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the HPE EDF system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the HPE EDF cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)
4. [Creating or Adding a CA \(optional\)](#)
5. [Registering a KMIP Client](#)
6. [Configuring the KMIP Interface](#)

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

1. Navigate to **Admin Settings > Domains**.
2. Click **Add Domain**. The **Add Domain** page appears.
3. Specify the following information:

- **Name** - Enter the domain name.
- **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.
- **Parent CA** - Select parent CA as root CA.
- **Allow Subdomain User Management** - Select this check box if you want to enable the sub-domain user management through this domain.

4. Click **Save**.
5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Sub-Domain

Note

To create a user in sub-domain, you must enable **Allow Subdomain User management**.

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the sub-domain.
2. Open the **Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example, **Key Admins** or **Key Users**.
6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.

-
- b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
 4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the knip interface.
 - c. Click **View/Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
-

-
- Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on the CA type that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.

- Specify client name and paste the **Registration Token** that you generated in the above step.

Note

If you are using external CA then you can either upload the client certificate as a file using **Upload Certificate** or paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub-Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.

- h. Click **Create Token**.
- i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the **Certificate Details** section.

- e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.

For domain, the format to enter the Common Name field of the cert is always:

`domainName | | domainUser`

- f. Click **Save**.

4. Create a Registration Token using the following steps:

- a. Go to **Registration Token** and click **New Registration Token > Begin**.
- b. Add a **Name Prefix**.
- c. Specify a Token Lifetime value along with the Client Capacity for the token.
- d. Click **Select CA**.
- e. Select the CA type as Local/External depending on CA that you created [above](#).
- f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
- g. Select the Client Profile from the dropdown which you have created in the above step.
- h. Click **Create Token**.
- i. Copy the value of the Token created and click **Done**.

5. If you are using External CA then you can select the external CA which you created earlier. Refer to **External CA** under [Creating or Adding a CA](#).

6. Go to Registered Clients and click **Add Client**.

- a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- a. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

6. Depending upon the CA that you created [above](#), select an option from the following::
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs**, provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Sub-Domain

1. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.

- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

External CA

Note

If you are using an External CA in the Sub-Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
 - b. Logout of your sub-domain and now login to the root domain.
 - c. Go to **CA > External > Add External CA**.
 - d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
 - e. Click **Add External CA**.
 - f. Go to **Admin Settings > Interfaces**.
 - g. Click the **Add** icon to add the External CA.
 - h. Click **Update**.
 - i. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.
 - j. Select the **Certificate Chain** option and click **Build Certificate Chain**.
 - k. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
 - l. Select certificate Format as **PEM**.
 - m. Click on **Upload Certificate**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
 3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

Configuration on HPE EDF

Perform the following steps to integrate HPE Ezmeral Data Fabric with the CipherTrust Manager:

1. Create a directory named `tokens` at `/opt/mapr/conf/` to store the key files and change the owner to `mapr`.

```
mkdir /opt/mapr/conf/tokens
chown mapr:mapr /opt/mapr/conf/tokens
```

2. Run the following command to create the KMIP token for the first time and initialize the KMIP configuration:

```
mrhsm init
-sopin <PIN for S0>
-label "CipherTrustManager"
-storetype kmip
```

3. After the Keystore is initialized, configure your KMIP settings using `mrhsm set` to configure the IP, Port, Certificates, KMIP Version, and the SO Pin.

```
mrhsm set
-sopin <PIN for S0>
-ip comma-separated list of KMIP server IP addresses>
-port <KMIP port number. Default is 5696>
-cacert <Path to KMIP server CA certificate in PEM format>
-clientcert <Path to client certificate in PEM format>
```

```
-clientkey <Path to client private key in PEM format>
-kmipversion 1.1
-storetype kmip
```

4. Use the `mrhsm info` command to view the updated HSM configuration information and status.

```
mrhsm info
```

5. Run the following command to enable DARE on the current node:

```
mrhsm enable -dare
```

After the above command is executed, the keys are generated and you can verify the integration by matching the UUID keys.

Verifying Your Integration

Perform the following steps to ensure the integration is successful:

1. Use the `mrhsm rekey` command on a CLDB node to rekey the common or core KEK, and use the core KEK to re-encrypt the CLDB and DARE keys. Run the following command with the appropriate parameters:

```
mrhsm rekey
-keytype core|common  Specifies the key type, which is either core or common
-sopin <so-pin>       PIN for SO (Security Officer)
```

Rekeying the Core KEK also involves decrypting the CLDB and DARE keys using the existing Core KEK before generating a new Core KEK, and then re-encrypting the CLDB and DARE keys using the new Core KEK. This command only updates the KMIP configuration on the CLDB node on which this command was invoked.

After the successful re-keying operation, a new key is generated on the CipherTrust Manager with its version upgraded by one.

Licensing

This section provides the licensing requirements for KMIP clients.

Condition	License Requirement
Encryption at cluster Level	x1 KMIP client license for each node in the cluster

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the Autoregistration option is selected in Admin Settings > System > Interfaces > kmip . If it is not selected, turn on the Autoregistration option. Refer to the steps mentioned in Enable KMIP Client Registration .
User corresponding to username location in the certificate (for example, OU/UID) is not created	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence the client is not auto registered in mode <code>tls-cert-pw-opt</code>"</i>	Check whether the user corresponding to UID/OU of the node certificates is created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the OU/UID field of the node certificates. After creating the user, add this user to the Key Admins . Refer to the steps mentioned in Create a New User .
User is not added to the Key Admin group	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "authorization denied: verdict was deny: CreateKey"</i>	Check whether the user corresponding to UID/OU of the node certificates is created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The username location in the Certificate is set incorrectly	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence the client is not auto registered in mode <code>tls-cert-pw-opt</code>"</i>	Check whether the username location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the username location in the Certificate.



CipherTrust Integrations

4 February 2026

Contents

- HPE Nimble Storage 3
 - Integration with CipherTrust Manager 3
 - Verifying Your Integration 16
 - Appendix. 17

HPE Nimble Storage

HPE Nimble Storage arrays are engineered for high performance using flash, low cost using dense, capacity-optimized disks, and easy installation and administration. Data written to the array is compressed and then stored in the disk-drive layer. HPE Nimble Storage arrays take advantage of multi-core processors to provide high-speed in-line variable-block compression without introducing noticeable latency. Nimble Storage arrays provide, accelerated performance for higher throughput or I/Os per second and sub-millisecond latencies, higher storage efficiency, non-disruptive scaling to fit changing application needs, maximized data, and storage availability with integrated data protection and disaster recovery.

Security in an HPE Nimble Storage group of arrays includes several features, including role-based access control for user accounts, encryption of data at rest, secure socket layer (SSL) certificates, and secure SMTP. You can enable encryption at the group level or volume level as required for each group of Nimble Storage arrays in your environment. Before you can create encrypted volumes, you must perform an initialization step that creates the master key. The master key protects the keys that are used to encrypt volume data. The encryption state of a volume is established when the volume is created, and cannot be changed afterward. Cloned volumes inherit the encryption state of their parent. The group configuration contains a default encryption setting, where you can enable or disable AES-256-XTS encryption.



Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

CipherTrust Manager

- CipherTrust Manager 2.8

HPE Nimble Storage Version

- HPE Nimble Storage 6.0.0.400-991061-opt

Integration with CipherTrust Manager

This section lists the steps to integrate Nimble Storage with CipherTrust Manager.

-
- Prerequisites
 - Configuration on CipherTrust Manager
 - Configuration on HPE Nimble Storage

Prerequisites

This section provides the prerequisites for integration of Nimble Storage with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager documentation](#).
- Nimble Storage communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the Nimble Storage system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the Nimble Storage cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)
4. [Creating or Adding a CA \(optional\)](#)
5. [Registering a KMIP Client](#)
6. [Configuring the KMIP Interface](#)

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

1. Navigate to **Admin Settings > Domains**.
2. Click **Add Domain**. The **Add Domain** page appears.
3. Specify the following information:

-
- **Name** - Enter the domain name.
 - **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.
 - **Parent CA** - Select parent CA as root CA.
 - **Allow Subdomain User Management** - Select this check box if you want to enable the sub-domain user management through this domain.

4. Click **Save**.

5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Sub-Domain

Note

To create a user in sub-domain, you must enable **Allow Subdomain User management**.

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the sub-domain.
2. Open the **Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example, **Key Admins** or **Key Users**.
6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.

-
- b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
 4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the knip interface.
 - c. Click **View/Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
-

-
- Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on the CA type that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.

- Specify client name and paste the **Registration Token** that you generated in the above step.

Note

If you are using external CA then you can either upload the client certificate as a file using **Upload Certificate** or paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub-Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.

-
- h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the **Certificate Details** section.

-
- e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.

For domain, the format to enter the Common Name field of the cert is always:

`domainName | | domainUser`

- f. Click **Save**.

4. Create a Registration Token using the following steps:

- a. Go to **Registration Token** and click **New Registration Token > Begin**.
- b. Add a **Name Prefix**.
- c. Specify a Token Lifetime value along with the Client Capacity for the token.
- d. Click **Select CA**.
- e. Select the CA type as Local/External depending on CA that you created [above](#).
- f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
- g. Select the Client Profile from the dropdown which you have created in the above step.
- h. Click **Create Token**.
- i. Copy the value of the Token created and click **Done**.

5. If you are using External CA then you can select the external CA which you created earlier. Refer to **External CA** under [Creating or Adding a CA](#).

6. Go to Registered Clients and click **Add Client**.

- a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- a. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

6. Depending upon the CA that you created [above](#), select an option from the following::
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs**, provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Sub-Domain

1. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.

- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

External CA

Note

If you are using an External CA in the Sub-Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
 - b. Logout of your sub-domain and now login to the root domain.
 - c. Go to **CA > External > Add External CA**.
 - d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
 - e. Click **Add External CA**.
 - f. Go to **Admin Settings > Interfaces**.
 - g. Click the **Add** icon to add the External CA.
 - h. Click **Update**.
 - i. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.
 - j. Select the **Certificate Chain** option and click **Build Certificate Chain**.
 - k. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
 - l. Select certificate Format as **PEM**.
 - m. Click on **Upload Certificate**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
 3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

Configuration on HPE Nimble Storage

Perform the following steps to integrate HPE Nimble Storage with the CipherTrust Manager:

1. Log in to the **Nimble Storage** using the credentials.
2. Navigate to **Administration >> Security >> SSL Certificate**. Add the CA (in case of domains, add the domain CA followed by the Root CA) and the Client Cert.
3. Navigate to **Administration >> Security >> Encryption**. Click on **External Key Manager** followed by **Add Key Manager**.
4. Enter the information for the following fields to set up your key management server:

Create Key Manager

Name *

CipherTrustManager

Description

Hostname or IP address *

Port *

Protocol *

Username

admin

Password

Show typing

Save Cancel

- Once the connection is successful, you can see the status as connected on Nimble Storage. You can also verify the key creation by logging on to CipherTrust Manager.

Verifying Your Integration

Perform the following steps to ensure the integration is successful:

- Restart the instance of Nimble Storage after the EKM has been set and configured.
- Navigate to **Manage > Data Storage**.
- Create a new volume and confirm if the status is online and encrypted.

Total Used: 0B		Settings		IOPS		IOPS	
Volumes		Folders					
				Summary		Status	
				Performance		General	
				Protection			
Name	Application	Volume Usage	Size	IOPS	Total IOPS	Latency (ms)	
Test 01-01-2024	Online	0 B	10 GB	0	0	1	0
Test 01-01-2024	Online	0 B	10 GB	0	0	1	0
Test 01-01-2024	Online	0 B	10 GB	0	0	1	0

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the Autoregistration option is selected in Admin Settings > System > Interfaces > kmip . If it is not selected, turn on the Autoregistration option. Refer to the steps mentioned in Enable KMIP Client Registration .
User corresponding to username location in the certificate (for example, OU/UID) is not created	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence the client is not auto registered in mode <code>tls-cert-pw-opt</code>"</i>	Check whether the user corresponding to UID/OU of the node certificates is created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the OU/UID field of the node certificates. After creating the user, add this user to the Key Admins . Refer to the steps mentioned in Create a New User .
User is not added to the Key Admin group	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "authorization denied: verdict was deny: CreateKey"</i>	Check whether the user corresponding to UID/OU of the node certificates is created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The username location in the Certificate is set incorrectly	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence the client is not auto registered in mode <code>tls-cert-pw-opt</code>"</i>	Check whether the username location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the username location in the Certificate.



CipherTrust Integrations

4 February 2026

Contents

HPE StoreEver. 3

 Integration with CipherTrust Manager 4

 Verifying Your Integration 17

 Appendix. 18

HPE StoreEver

HPE StoreEver is a tape automation product that helps SMB customers to meet long-term data retention requirements, as well as provides air-gapped data protection from the growing threat of ransomware. The HPE StoreEver tape library offers broad capacity options including LTO-9, LTO-8, LTO-7, or LTO-6 Ultrium drives. HPE CVTL makes the HPE StoreEver tape library easy to manage from any location, mitigating the need for remote office IT staff. You can simply manage tape media in and out of the library with the standard bar code reader, configurable mail slots, and removable magazines.

Using the Key Management Interoperability Protocol (KMIP) Wizard, you can set up KMIP key management servers with the MSL3040 and MSL6480 libraries. The wizard is accessible through the Encryption menu on the RMI, but only for the security user and requires a KMIP license.

If a tape is lost or stolen, you can protect important business data from unauthorized access with data encryption.



Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

CipherTrust Manager

- CipherTrust Manager 2.7 and higher

HPE StoreEver Version

- MSL3040
- MSL6480
- HPE 1/8 G3

-
- MSL2024 G4

** Key Management Interoperability Protocol (KMIP)**

- Upto KMIP 1.4

Integration with CipherTrust Manager

This section lists the steps to integrate HPE StoreEver with CipherTrust Manager.

- Prerequisites
- Configuration on CipherTrust Manager
- Configuration on HPE StoreEver

Prerequisites

This section provides the prerequisites for integration of HPE StoreEver with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager documentation](#).
- HPE StoreEver communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the HPE StoreEver system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the HPE StoreEver cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
 2. [Creating a User](#)
 3. [Assigning User to a Group](#)
 4. [Creating or Adding a CA \(optional\)](#)
 5. [Registering a KMIP Client](#)
 6. [Configuring the KMIP Interface](#)
-

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

1. Navigate to **Admin Settings > Domains**.
2. Click **Add Domain**. The **Add Domain** page appears.
3. Specify the following information:
 - **Name** - Enter the domain name.
 - **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.
 - **Parent CA** - Select parent CA as root CA.
 - **Allow Subdomain User Management** - Select this check box if you want to enable the sub-domain user management through this domain.
4. Click **Save**.
5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Sub-Domain

Note

To create a user in sub-domain, you must enable **Allow Subdomain User management**.

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the sub-domain.
2. Open the **Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example, **Key Admins** or **Key Users**.
6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**

- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **View/Edit**.
-

-
- d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on the CA type that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.

-
- h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.

- Specify client name and paste the **Registration Token** that you generated in the above step.

Note

If you are using external CA then you can either upload the client certificate as a file using **Upload Certificate** or paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub-Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:

-
- a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the **Certificate Details** section.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.

For domain, the format to enter the Common Name field of the cert is always:
`domainName | | domainUser`

- f. Click **Save**.
4. Create a Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.
5. If you are using External CA then you can select the external CA which you created earlier. Refer to **External CA** under [Creating or Adding a CA](#).
6. Go to Registered Clients and click **Add Client**.
 - a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- a. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

6. Depending upon the CA that you created [above](#), select an option from the following::
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs**, provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Sub-Domain

1. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

External CA

Note

If you are using an External CA in the Sub-Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.
- i. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.

-
- j. Select the **Certificate Chain** option and click **Build Certificate Chain**.
 - k. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
 - l. Select certificate Format as **PEM**.
 - m. Click on **Upload Certificate**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
 3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

Configuration on HPE StoreEver

For the visual representation of the configuration steps on HPE StoreEver, refer to the GIF displayed below.

Key Management Interoperability Protocol (KMIP) Wizard

► **Wizard Information**

- Certificate Authority Information
- Certificate Authority Certificate Entry
- Library Certificate Information
- KMIP Client Configuration
- Certificate Generation
- Sign Library Certificate
- KMIP Server Configuration
- Setup Summary

Wizard Information

With the KMIP Wizard you can configure the library for use with the KMIP key management server. Before starting the wizard, ensure that the library configuration is complete, including defining all library partitions. Also, ensure that the KMIP server has been put on the network and configured for use with this library.

For questions on the KMIP server setup, see the applicable KMIP server user guide.

To continue the wizard, click **Next**.

To clear the wizard settings when no longer using this encryption method or reset the wizard configuration to its default state, click **Clear All Wizard Settings**.

Clear All Wizard Settings

Back Next Finish Cancel

Perform the following steps to integrate HPE StoreEver with the CipherTrust Manager:

1. Log in to the **StoreEver** console as a security user.
2. Navigate to **Configuration > Encryption > KMIP Wizard**.
3. Click **Next**. The **Certificate Authority Information** screen displays prerequisites for using the KMIP certificate. When the prerequisites are met, click **Next**.
4. The **Certificate Authority Certificate Entry** screen displays instructions for obtaining the certificate for the KMIP server. Follow the instructions to copy the certificate from the management console. Paste the certificate into the wizard and then click **Next**.
5. The **Library Certificate Information** screen displays information about the next wizard steps. Click **Next**.
6. The **KMIP Client Configuration** screen provides options for two types of server authentication.
 - When the KMIP server uses a client username and password for authentication, enter the username and password that were specified on the KMIP management console for the library.
 - When the KMIP server uses only certificate passing for authentication, select **Enable KMIP Certificate-only authentication**.

Note

Select **Enable KMIP Certificate-only authentication** option only if it is required and a client's username and password is not available.

Click **Next**.

7. The **Certificate Generation** screen displays the current library certificate, if one exists.

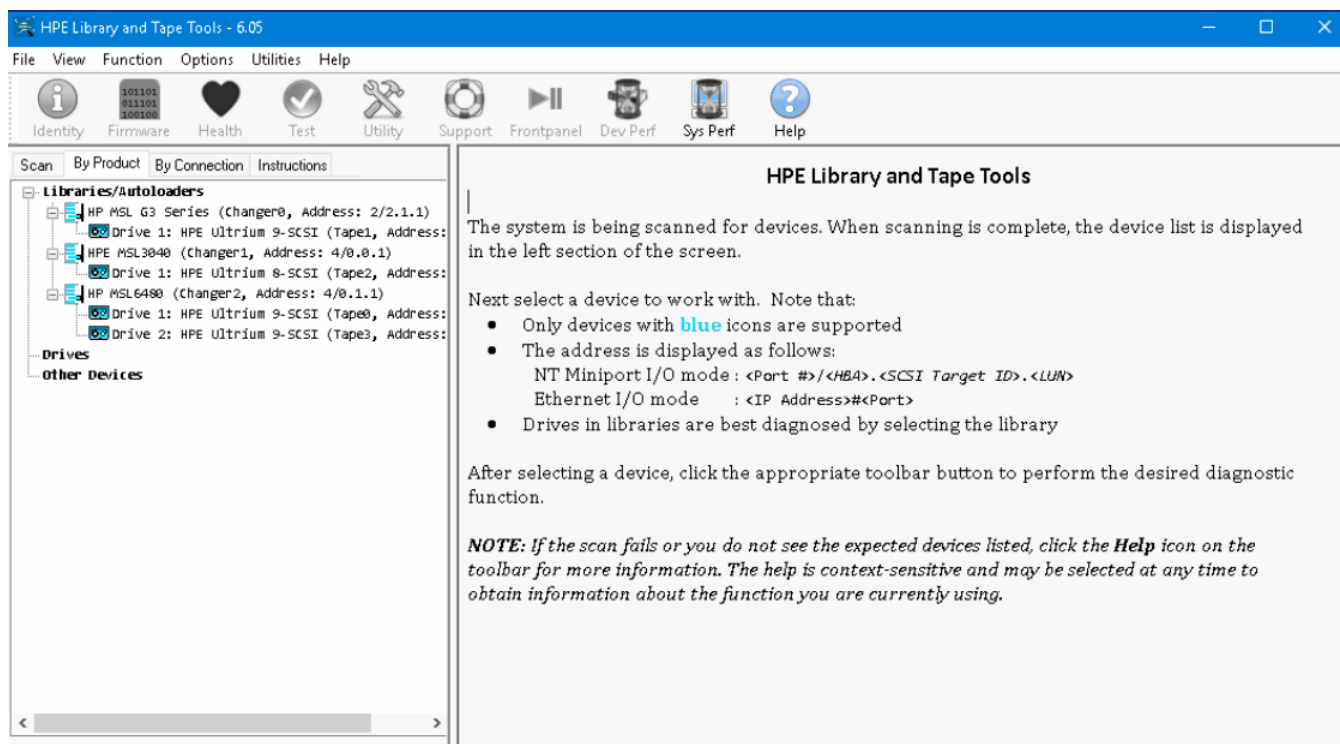
To use the current certificate, select **Keep Current Certificate** and then click **Next**.

To generate a new certificate, select **Generate New Certificate**. The wizard will generate and display a new library certificate. Click **Select Certificate** to copy the new certificate text and then click **Next**.

8. When selected **Generate New Certificate**, the **Sign Library Certificate** screen displays the new certificate for the library. Sign the new library certificate with the certificate authority as a client certificate, paste the new KMIP certificate in the box, and then click **Next**.
9. On the **KMIP Server Configuration** page, enter the IP address or fully-qualified hostname and port number for up to ten KMIP servers. To verify access to the KMIP servers, click **Connectivity Check**.
10. The Setup Summary screen displays the settings that were collected by the wizard. Verify that the settings are correct and that there are no errors in the **Done** column. To modify any settings or fix any issues, either click **Back** to reach the applicable screen or cancel out of the wizard to fix the issues and return later. If the settings are correct and there are no errors, click **Finish**.
11. On the **Configuration** page:
 - Under the **Set Default Encryption Mode for new Partitions** section, select **Controlled by Backup Application** from the dropdown menu and Click **Apply to all existing partitions** to configure partitions for use with KMIP
 - Under the **Set Encryption Mode per Partition**, select **KMIP (Licensed)** from the dropdown menu.
12. Click **Next**.

Verifying Your Integration

For the visual representation of the steps to verify the integration, refer to the GIF displayed below.



Perform the following steps to verify the integration of HPE StoreEver with the CipherTrust Manager:

1. Launch **HPE Library and Tape Tools** and select the following options to perform the Hardware Scan:
 - **Scan Mode:** Select Hardware Scan
 - **I/O Mode:** Select Miniport
 - Select **Check for Backup Application and Services** check box and click **Continue**
2. Select the required libraries and then select **Dev Perf** option available at the menu bar.
3. Select the drives and then click **Options** to set the parameters.
4. Use the parameters for the **Drive Performance Test**.
5. Click **OK**, and click **Start** to begin the test.

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client"</i>	Check whether the Autoregistration option is selected in Admin Settings > System > Interfaces > kmip .

Issue	Error Message	Remediation
	<i>from CLI or API or UI."</i>	If it is not selected, turn on the Autoregistration option. Refer to the steps mentioned in Enable KMIP Client Registration .
User corresponding to username location in the certificate (for example, OU/UID) is not created	The following error messages are displayed in CipherTrust Records: "errorMessage": "username not found: " "errorMessage": "Could not authenticate certificate user, hence the client is not auto registered in mode <code>tls-cert-pw-opt</code> "	Check whether the user corresponding to UID/OU of the node certificates is created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the OU/UID field of the node certificates. After creating the user, add this user to the Key Admins . Refer to the steps mentioned in Create a New User .
User is not added to the Key Admin group	The following error message is displayed in CipherTrust Records: "errorMessage": "authorization denied: verdict was deny: CreateKey"	Check whether the user corresponding to UID/OU of the node certificates is created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The username location in the Certificate is set incorrectly	The following error messages are displayed in CipherTrust Records: "errorMessage": "username not found: " "errorMessage": "Could not authenticate certificate user, hence the client is not auto registered in mode <code>tls-cert-pw-opt</code> "	Check whether the username location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the username location in the Certificate.



CipherTrust Integrations

4 February 2026

Contents

HPE StoreOnce 3

 Integration with CipherTrust Manager 3

 Verifying Your Integration 15

 Appendix. 16

HPE StoreOnce

HPE StoreOnce delivers scalable backup and restore for small to midsize data centers, and provides an ideal replication target device for up to 36 remote and branch offices. This scalable 2U appliance offers from 36 to 108TB of usable local capacity using an upgrade kits and up to 324TB with Cloud bank storage, and provides a solution to shrinking backup windows with speeds of up to 18TB/hr using HPE StoreOnce catalyst.

HPE StoreOnce delivers a unified solution for data movement across the enterprise under the control of your data protection application.



Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

CipherTrust Manager

- CipherTrust Manager 2.8

HPE StoreOnce Version

- HPE StoreOnce 4.3.2-2221.21

Integration with CipherTrust Manager

This section lists the steps to integrate Integration of HPE Primera with CipherTrust Manager.

- [Prerequisites](#)
- [Configuration on CipherTrust Manager](#)
- [Configuration on HPE StoreOnce](#)

Prerequisites

This section provides the prerequisites for integration of HPE StoreOnce with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager documentation](#).
- HPE StoreOnce communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the HPE StoreOnce system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the HPE StoreOnce cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)
4. [Creating or Adding a CA \(optional\)](#)
5. [Registering a KMIP Client](#)
6. [Configuring the KMIP Interface](#)

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

1. Navigate to **Admin Settings > Domains**.
2. Click **Add Domain**. The **Add Domain** page appears.
3. Specify the following information:
 - **Name** - Enter the domain name.
 - **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.
 - **Parent CA** - Select parent CA as root CA.

-
- **Allow Subdomain User Management** - Select this check box if you want to enable the sub-domain user management through this domain.

4. Click **Save**.

5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Sub-Domain

Note

To create a user in sub-domain, you must enable **Allow Subdomain User management**.

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the sub-domain.
2. Open the **Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.

-
4. On the **Users** page, click **Add User**.
 5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter **Password**. Re-enter the password in the **Password Match** field.
 6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example, **Key Admins** or **Key Users**.
6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
-

-
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.
 3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
 4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:

- a. Go to **Registration Token** and click **New Registration Token > Begin**.
- b. Add a **Name Prefix**.
- c. Specify a Token Lifetime value along with the Client Capacity for the token.
- d. Click **Select CA**.
- e. Select the CA type as Local/External depending on CA that you created [above](#).
- f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
- g. Select the Client Profile from the dropdown which you have created in the above step.
- h. Click **Create Token**.
- i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **View/Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on the CA type that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.
-

- Specify client name and paste the **Registration Token** that you generated in the above step.

Note

If you are using external CA then you can either upload the client certificate as a file using **Upload Certificate** or paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub-Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.

- i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the **Certificate Details** section.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.

For domain, the format to enter the Common Name field of the cert is always:

`domainName | | domainUser`

- f. Click **Save**.
4. Create a Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.
5. If you are using External CA then you can select the external CA which you created earlier. Refer to **External CA** under [Creating or Adding a CA](#).
6. Go to Registered Clients and click **Add Client**.
 - a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- a. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
 2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
-

3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

6. Depending upon the CA that you created [above](#), select an option from the following::
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs**, provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Sub-Domain

1. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.

- h. Click **Update**.

External CA

Note

If you are using an External CA in the Sub-Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
 - b. Logout of your sub-domain and now login to the root domain.
 - c. Go to **CA > External > Add External CA**.
 - d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
 - e. Click **Add External CA**.
 - f. Go to **Admin Settings > Interfaces**.
 - g. Click the **Add** icon to add the External CA.
 - h. Click **Update**.
 - i. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.
 - j. Select the **Certificate Chain** option and click **Build Certificate Chain**.
 - k. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
 - l. Select certificate Format as **PEM**.
 - m. Click on **Upload Certificate**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
 3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

Configuration on HPE StoreOnce

Perform the following steps to integrate HPE StoreOnce with the CipherTrust Manager (CM):

1. Download the client certificate and corresponding CA from the CM and save them.
2. Go back to **StoreOnce** and click **Enroll**.
3. Click **Backup the key manager configuration**, add a password, and click **Backup**. A backup file gets downloaded.
4. Click **Configure External Key Manager Server** and select **SafeNet**.
5. Enter the IP address of CM with port and password and click **OK**.
6. Click **Import Root Certificate** and paste the CA certificate that you have downloaded from the CM.
7. Click **Import Signed Certificate** and paste the client certificate that you have downloaded from the CM.
8. Click **Enroll** and a connection is created with the CM.
9. On successful integration, two keys will be created on the CM.

Verifying Your Integration

Perform the following steps to ensure the integration is successful:

1. Navigate to **Data Services > NAS Shares**.
2. Click **Create Share**, the **Create Share** Window appears.
3. Add name of the share and under **Security Settings** enable **Share Encryption** and click **Create**.

On successful creation of the NAS share, the status of NAS share will turn green and a key is created on CM.

Settings and more (Alt+F): An update is available.

←	 NAS Shares so3b40				
Name	Status	User Data Stored	Size on Disk	Dedupe Ratio	Replication Rule
myNAS	● Online	0 B	3.3 MB	—	Non-Replicating

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the Autoregistration option is selected in Admin Settings > System > Interfaces > kmip . If it is not selected, turn on the Autoregistration option. Refer to the steps mentioned in Enable KMIP Client Registration .
User corresponding to username location in the certificate (for example, OU/UID) is not created	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence the client is not auto registered in mode <code>tls-cert-pw-opt</code>"</i>	Check whether the user corresponding to UID/OU of the node certificates is created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the OU/UID field of the node certificates. After creating the user, add this user to the Key Admins . Refer to the steps mentioned in Create a New User .
User is not added to the Key Admin group	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "authorization denied: verdict was deny: CreateKey"</i>	Check whether the user corresponding to UID/OU of the node certificates is created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The username location in the Certificate is set incorrectly	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence the client is not auto registered in mode <code>tls-cert-pw-opt</code>"</i>	Check whether the username location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the username location in the Certificate.



CipherTrust Integrations

4 February 2026

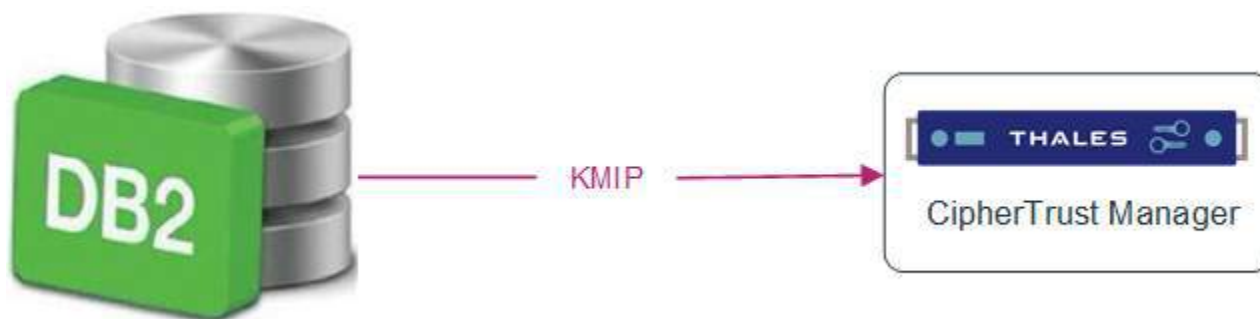
Contents

- IBM DB2..... 3
 - Integration with CipherTrust Manager 3
 - Verifying Your Integration 20

IBM DB2

IBM Db2 is a family of data management products, including the Db2 relational database. The products feature AI-powered capabilities to help you modernize the management of both structured and unstructured data across on-premises and multicloud environments. By helping to make your data simple and accessible, the Db2 family positions your business to pursue the value of AI.

DB2 integrates with the CipherTrust Manager to maintain the confidentiality of the keys and certificates. The integration provides a trusted system for Data At Rest encryption.



Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

CipherTrust Manager

- CipherTrust Manager 2.3 and higher

IBM DB2

- DB2 11.5 and higher

This integration is validated on **Red Hat Enterprise Linux 8.2**.

Integration with CipherTrust Manager

This section lists the steps to integrate IBM DB2 with CipherTrust Manager.

- [Prerequisites](#)
- [Configuration on CipherTrust Manager](#)

-
- Adding the CA Cert and Generating the CSR
 - Generating the Client Certificate
 - Configuration on IBM DB2

Prerequisites

This section provides the prerequisites for integration of IBM Db2 with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager documentation](#).
- Ensure that Db2 is installed and configured. For more details, refer to the [IBM Documentation](#).
- Db2 communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the Db2.
- Ensure that the required licenses are activated. For more details, refer to the [CipherTrust Manager documentation](#).

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)
4. [Creating or Adding a CA \(optional\)](#)
5. [Registering a KMIP Client](#)
6. [Configuring the KMIP Interface](#)

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

1. Navigate to **Admin Settings > Domains**.
 2. Click **Add Domain**. The **Add Domain** page appears.
 3. Specify the following information:
 - **Name** - Enter the domain name.
-

-
- **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.
 - **Parent CA** - Select parent CA as root CA.
 - **Allow Subdomain User Management** - Select this check box if you want to enable the sub-domain user management through this domain.

4. Click **Save**.
5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Sub-Domain

Note

To create a user in sub-domain, you must enable **Allow Subdomain User management**.

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the sub-domain.

-
2. Open the **Access Management** application.
 3. On the left pane, click **Users**. The Users page is displayed.
 4. On the Users page, click **Add User**.
 5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter **Password**. Re-enter the password in the **Password Match** field.
 6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example, **Key Admins** or **Key Users**.
6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.

-
- b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
 4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the knip interface.
 - c. Click **View/Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
-

-
- Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on the CA type that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.

- Specify client name and paste the **Registration Token** that you generated in the above step.

Note

If you are using external CA then you can either upload the client certificate as a file using **Upload Certificate** or paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub-Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager into your domain.
 2. Go to **Products > KMIP**.
 3. Create **Client Profile** using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
 4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
-

-
- h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the **Certificate Details** section.

-
- e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.

For domain, the format to enter the Common Name field of the cert is always:

`domainName | | domainUser`

- f. Click **Save**.

4. Create a Registration Token using the following steps:

- a. Go to **Registration Token** and click **New Registration Token > Begin**.
- b. Add a **Name Prefix**.
- c. Specify a Token Lifetime value along with the Client Capacity for the token.
- d. Click **Select CA**.
- e. Select the CA type as Local/External depending on CA that you created [above](#).
- f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
- g. Select the Client Profile from the dropdown which you have created in the above step.
- h. Click **Create Token**.
- i. Copy the value of the Token created and click **Done**.

5. If you are using External CA then you can select the external CA which you created earlier. Refer to **External CA** under [Creating or Adding a CA](#).

6. Go to Registered Clients and click **Add Client**.

- a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- a. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

6. Depending upon the CA that you created [above](#), select an option from the following::
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs**, provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Sub-Domain

1. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.

- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

External CA

Note

If you are using an External CA in the Sub-Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
 - b. Logout of your sub-domain and now login to the root domain.
 - c. Go to **CA > External > Add External CA**.
 - d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
 - e. Click **Add External CA**.
 - f. Go to **Admin Settings > Interfaces**.
 - g. Click the **Add** icon to add the External CA.
 - h. Click **Update**.
 - i. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.
 - j. Select the **Certificate Chain** option and click **Build Certificate Chain**.
 - k. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
 - l. Select certificate Format as **PEM**.
 - m. Click on **Upload Certificate**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
 3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

Further, you need to configure the following:

- Adding the CA Cert and Generating the CSR
- Generating the Client Certificate

Adding the CA Cert and Generating the CSR

You can download and add the CA cert in the local keystore for any of the following Certificate Authorities:

Local CA

1. In the CipherTrust Manager, download the Local CA Cert (Certificate.pem).
2. Move the downloaded CA Cert to work folder that you created in DB2 Server.
3. Rename the `Certificate.pem` to `ca.crt`.

```
mv <work_folder_location/Certificate.pem> <work_folder_location/ca.crt
```

For example: `mv /opt/wallet/KMIP/Certificate.pem /opt/wallet/KMIP/ca.crt`

4. Add the CA cert to the local keystore.

```
</home/<DB2 user instance>/sqlllib/gskit/bin/gsk8capicmd_64 -cert -add -db <work_folder_location/local_keystore> -stashed -label <LABEL FOR CA CERTIFICATE> -file <CA CERTIFICATE FILE>
```

For Example: `/home/db2user/sqllib/gskit/bin/gsk8capicmd_64 -cert -add -db "/opt/wallet/KMIP/clientkeydb.p12" -stashed -label "DB2_CA_CERT" -file "/opt/wallet/KMIP/ca.crt"`

External CA

1. Generate the External CA using OpenSSL or any other CA generator.
2. Upload the CA on the CipherTrust Manager, using the following steps:
 - a. Go to **CA > External**.
 - b. On the CA page, click **Add External CA**.
 - c. Paste the contents of the `rootCACert.pem` and click **Save**.
3. Move the downloaded CA Cert to work folder that you created in DB2 Server.
4. Add the CA cert to the local keystore.

```
</home/<DB2 user instance>/sqllib/gskit/bin/gsk8capicmd_64 -cert -add -db <work_folder_location/local_keystore> -stashed -label <LABEL FOR CA CERTIFICATE> -file <CA CERTIFICATE FILE>
```

For Example: `/home/db2user/sqllib/gskit/bin/gsk8capicmd_64 -cert -add -db "/opt/wallet/KMIP/clientkeydb.p12" -stashed -label "DB2_CA_CERT" -file "/opt/wallet/KMIP/ca.crt"`

Generating the CSR for Client Certificate

After adding the CA cert to the local keystore, generate the Client Certificate request using the following steps:

Root

In the DB2 server, generate the Client Certificate Request. Use the newly created user created on CipherTrust Manager (DB2_CLIENT_CERT in this case) as reference while generating the CSR.

```
<DB2_HOME>/sqllib/gskit/bin/gsk8capicmd_64 -certreq -create -db < LOCAL KEYSTORE FILE> -stashed -label <LABEL FOR CLIENT CERTIFICATE> -dn <CLIENT CERTIFICATE DETAILS> -target <CLIENT CSR FILE> -size 2048 -sigalg SHA256
```

For Example: `/home/db2user/sqllib/gskit/bin/gsk8capicmd_64 -certreq -create -db "/opt/wallet/KMIP/clientkeydb.p12" -stashed -label "DB2_CLIENT_CERT" -dn "CN=DB2_CLIENT_CERT, O=Gemalto, OU=PA, L=Noida, ST=UP, C=IN" -target "/opt/wallet/KMIP/client_cert_request.arm" -size 2048 -sigalg SHA256`

Sub-Domain

To generate the CSR for client certificate in a sub-domain, pass the CN as `CN=DB2_CLIENT_CERT| |username`, where **username** is the user created above.

```
<DB2_HOME>/sqllib/gskit/bin/gsk8capicmd_64 -certreq -create -db < LOCAL KEYSTORE
```



```
FILE> -stashed -label <LABEL FOR CLIENT CERTIFICATE> -dn <CLIENT CERTIFICATE DATA>
ILS> -target <CLIENT CSR FILE> -size 2048 -sigalg SHA256
```

For Example: /home/db2user/sql/lib/gskit/bin/gsk8capicmd_64 -certreq -create -db "/opt/wallet/KMIP/clientkeydb.p12" -stashed -label "DB2_CLIENT_CERT" -dn "CN=DB2_CLIENT_CERT|user1, O=Gemalto, OU=PA, L=Noida, ST=UP, C=IN" -target "/opt/wallet/KMIP/client_cert_request.arm" -size 2048 -sigalg SHA256

Generating the CSR for Server Certificate

Note

Server certificate is required only when you are using External CA. You may skip this step if you are using the Local CA.

To create a server certificate, perform the following steps:

1. Create a server key using the following command.

```
openssl genrsa -out server.key 2048
```

2. Create a CSR for the server, using the above generated key.

```
openssl req -key server.key -new -sha256 -out server.csr -subj /O=my-org/OU=my-org/OU=server/CN=server
```

After generating the CSR (`server.csr` in this case), create a server certificate using the following openssl command:

```
openssl x509 -req -days 7300 -in server.csr -CA rootCACert.pem -CAkey rootCAKey.pem -CAcreateserial -out server.cert -sha256
```

This Server key and Server certificate will be used in configuring the KMIP interface.

Generating the Client Certificate

Before generating the client certificate, ensure that you have registered the KMIP client.

Tip

To register the KMIP client, click [here](#).

You can generate the client certificate using any of the following two options:

Manual Registration

1. Generate a registration token using the Local Root CA and the profile [generated in the previous step](#) under Manual Registration.
2. Copy the registration token and go to Registered clients > Add client. Specify the client name and paste the registration token and click **Save**.
3. Download the resultant client certificate.
4. Move the downloaded client certificate in the working folder (/opt/wallet/KMIP) of DB2 server and rename it to **client.crt**.

After moving and renaming the client certificate as mentioned above, add the signed client certificate into local KeyStore. To do so, execute the following command:

```
/home/<db2 instance user>/sqllib/gskit/bin/gsk8capicmd_64 -cert -receive -db <LOCAL KEYSTOREFILE> -stashed -file <CLIENT CERTIFICATE>
```

For Example: /home/db2user/sqllib/gskit/bin/gsk8capicmd_64 -cert -receive -db "/opt/wallet/KMIP/clientkeydb.p12" -stashed -file "client.crt"

Auto Registration

1. Copy the CSR generated [previously](#).
2. On the CipherTrust Manager GUI perform the following steps:
 - a. Go to **CA > Local** , click on the Local CA present there.
 - b. Click on **Upload CSR**.
 - c. Add a specified name in **Display name** and paste the content of CSR in the **CSR** column and select **Certificate Purpose** as **client**.
 - d. Click on **Issue Certificate** and save the certificate.
3. Move the downloaded client certificate in the working folder (/opt/wallet/KMIP) of DB2 server and rename it to **client.crt**.

After moving and renaming the client certificate as mentioned above, add the signed client certificate into local KeyStore. To do so, execute the following command:

```
/home/<db2 instance user>/sqllib/gskit/bin/gsk8capicmd_64 -cert -receive -db <LOCAL KEYSTOREFILE> -stashed -file <CLIENT CERTIFICATE>
```

For Example: /home/db2user/sqllib/gskit/bin/gsk8capicmd_64 -cert -receive -db "/opt/wallet/KMIP/clientkeydb.p12" -stashed -file "client.crt".

Configuration on IBM DB2

Generating Master Encryption Key in the CipherTrust Manager

To generate the Master Encryption Key in the CipherTrust Manager, perform the following steps:

1. Create a centralized KeyStore.
 - a. To store a Master Encryption Key in CipherTrust Manager for a DB2 native encryption, you need to create a configuration file that contains all details about this centralized KeyStore i.e. the CipherTrust Manager.
 - b. In the work folder (/opt/wallet/KMIP) of the DB2 server, create kmip.cfg file with the following details:

```
VERSION=1
PRODUCT_NAME=KEYSECURE
ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP=true
SSL_KEYDB=/opt/wallet/KMIP/clientkeydb.p12
SSL_KEYDB_STASH=/opt/wallet/KMIP/clientkeydb.sth
SSL_KMIP_CLIENT_CERTIFICATE_LABEL=DB2_CLIENT_CERT
ALLOW_NONCRITICAL_BASIC_CONSTRAINT=false
MASTER_SERVER_HOST=IP of CM
MASTER_SERVER_KMIP_PORT=<port for KMIP interface>
CLONE_SERVER_HOST=
CLONE_SERVER_KMIP_PORT=
```

where:

- **SSL_KEYDB (Required):** Absolute path and name of the local keystore file that holds the SSL certificates for communication between DB2 server and the CipherTrust Manager.
- **SSL_KEYDB_STASH (Optional):** Absolute path and name of the stash file for the local keystore that hold the certificates for communication between DB2 server and the CipherTrust Manager.
- **SSL_KMIP_CLIENT_CERTIFICATE_LABEL (Required):** The label of the SSL certificate for authenticating the client during communication with the CipherTrust Manager.
- **MASTER_SERVER_HOST** is the CipherTrust Manager IP.
- **MASTER_SERVER_KMIP_PORT** is the KMIP interface Port in the CipherTrust Manager.

2. Configure a DB2 instance to use CipherTrust Manager as centralized keystore.

- a. Enter into DB2 command line using the **db2** command with the DB2@instance user.
- b. Update the keystore to point to our new config file **kmip.cfg**.

```
db2 => update dbm cfg using keystore_location <file path>/kmip.cfg
```

```
db2 => update dbm cfg using keystore_type kmip
```

c. Stop and Start the database.

```
db2 => db2stop  
db2 => db2start
```

Creating an Encrypted Database

To create an encrypted database you need to add the parameter **encrypt** in the create database command.

```
db2 => create db <database_name> encrypt
```

For Example: db2 => create db myencdb encrypt

Note

- In the above example, an encrypted database **myencdb** is created. This database is using the native encryption of DB2.
- After executing the above command, a master encryption key is created in the CipherTrust Manager.

Verifying Your Integration

On successful integration a master encryption key should be generated in the CipherTrust Manager under **Keys**.

To verify your database has been successfully encrypted by DB2 Native encryption, execute the following command:

```
db2 => get db cfg for <database_name>
```

Ensure that the value of the **Encrypted database** configuration parameter is set to **YES**.



CipherTrust Integrations

4 February 2026

Contents

- IBM FileNet Content Manager 3
 - Integration with CipherTrust Manager 3
 - Verifying Integration 7
 - Appendix. 8

IBM FileNet Content Manager

IBM FileNet Content Manager is a flexible, full-featured content management solution that provides the foundation for IBM Cloud Pak® for Business Automation. It uses AI to deliver deep insights from your unstructured content. Use it to create innovative business applications on any cloud and more effectively manage all your content, from any source.

Supported Product Versions

Note

This integration has been validated in the field by a partner or in a customer environment with the following software versions. It is recommended to test the integration in a non-production environment with desired software versions before deploying it to production. Thales will provide best-effort support.

CipherTrust Manager

- CipherTrust Manager 2.4.0

IBM FileNet Content Manager

- IBM FileNet Content Manager 5.5

Integration with CipherTrust Manager

This section lists the steps to integrate FileNet with CipherTrust Manager.

- [Prerequisites](#)
- [Configuration on CipherTrust Manager](#)
- [Configuration on FileNet Content Manager](#)

Prerequisites

This section provides the prerequisites for integration of FileNet Content Manager with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager documentation](#).
- FileNet Content Manager communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the FileNet

Content Manager system.

- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the FileNet Content Manager cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.

Configuration on CipherTrust Manager

Creating and Signing Certificates Using KSCTL Utility

Note

KSCTL is used to create CA and CSRs at the CipherTrust Manager. To know about how to install and configure KSCTL refer to the [CipherTrust Manager Documentation](#).

Creating the CA Certificate

The rootCA is by default available at the CipherTrust Manager. If you want to create a Local CA other than rootCA, use the following command:

```
ksctl ca locals create --cn "Test CA" --csr-outfile csrfile
```

To self-sign the CA for a specific duration, use the id returned in the above command:

```
ksctl ca locals self-sign --id <id of CA> -x <duration>
```

For Example:

```
ksctl ca locals self-sign --id 3593c53b-fbeb-4edb-b84d-c85526ae2f83 -x 365
```

Creating the Client Certificate

To create the client certificate, use the following steps:

1. Create CSR and its private key.

```
ksctl ca csr --cn "FileNet Content Manager" --csr-outfile client.csr --key-outfile client.key
```

This command generates a CSR along with its private key.

Note

In this document, common name **CN=FileNet Content Manager** is used, which will be further used as a user name on the CipherTrust Manager.

2. Sign the CSR with CA.

```
ksctl ca locals certs issue --ca-id <id of CA> --csr-infile client.csr -x <duration> -o client
```

For Example:

```
ksctl ca locals certs issue --ca-id c84d66e1-2101-4a46-ad44-3656354b5097 --csr-infile client.csr -x 700 -o client
```

--ca-id is the id which can be obtained from CA details on the Ciphertrust Manager.

Note

--csr-infile is the csr name which is created in the previous step.

Creating the Server Certificate

Note

Server Certificate is needed only in case if you are using external CA instead of Local CA.

To create a server certificate, create CSR and its private key.

```
ksctl ca csr --cn "server" --csr-outfile server.csr --key-outfile server.key
```

This command generates a CSR along with its private key.

You can sign the above created CSR using external CA. The generated certificate will be uploaded to the CipherTrust Manager.

Configuring the CipherTrust Manager

1. Create User on CipherTrust Manager.

- a. In the CipherTrust Manager UI, go to **Access Management > Users** and create a new user.
- b. Click on the newly created user, click on **GROUPS** and select the Key Users and Key Admins Group.

Note

Username must be same as common name mentioned above while creating the client certificate.

2. Register KMIP client.

To perform KMIP client registration, refer to [KMIP Client Registration](#).

3. Configure KMIP Interface.

- a. Go to **Admin Settings > Interfaces**.
- b. On the KMIP Interface, click the action button (...) and then click **Edit**.

Configure KMIP popup opens.

- c. Check **Auto Registration** option if you are going with Auto Registration or uncheck if you are going with the Manual Registration.
- d. Select the mode as **TLS, verify client cert, username taken from client cert, auth request is optional**.
- e. Specify selections for **Local CA for Automatic Server Certificate Generation and Local Trusted CAs** as desired.

Note

Local CA for Automatic Server Certificate Generation should be set to **Turn off auto generation from Local CA** in case of External CA.

- f. Select the CA according to your preference.
 - If you are using External CA then select the CA under **External Trusted CAs**
 - If you are using Local CA then select the CA under **Local Trusted CAs**
- g. This step is required if you are using external CA.

Expand the **Upload Certificate** section:

- In the **Certificate** field, paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space or character or symbol between the contents of these files.
- Select certificate **Format** as **PEM**.
- **Password** field is optional and can be skipped.
- Click **Update**.

Configuration on FileNet Content Manager

Connect to the instance using SSH. The FileNet Content Manager shell appears. Perform the following steps on the FileNet Content Manager shell:

1. Import the External CA root certificate to truststore on application server.
2. Import the PFX certificate file for FileNet client to certificate store on application server.
3. On KMIP configuration (Key Management Tab)
 - a. Select **KMIP** for **Key Service type** dropdown box.
 - b. Enter the IP address or hostname on **Primary Host** field.
 - c. Enter the the port for KMIP interface you use on CTM for **Primary port**.
 - d. If you do not have secondary server, enter the same values to secondary IP and port otherwise FileNet will throw an error.
4. Click **Save** button on top left.

This completes the configuration on FileNet Content Manager.

Verifying Integration

Perform the following steps to ensure the successful integration:

After performing the integration steps, you can create keys to verify if the integration is successful.

Execute the following command to create a key from IBM FileNet Content Manager:

```
security key-manager key create
```

In the CipherTrust Manager GUI, navigate to **Keys**. Check if there is an entry for this newly created key. Usually, most recently created keys are listed on the top of the table.

If the key is created, the integration is successful.

Appendix

Troubleshooting

This section describes the how to troubleshoot the issue that you may encounter while integrating CipherTrust Manager with IBM FileNet Content Manager.

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the KMIP registration is completed or not in Admin Settings > System > Interfaces > kmip . If it is not completed, refer to the steps mentioned in KMIP Client Registration .
User corresponding to username location in certificate (example: CN) has not been created CN = Common Name	The following error messages are displayed in CipherTrust Records: *"errorMessage": "username not found: " "errorMessage": "Could not authenticate certificate user, *	Check whether the user corresponding to CN of the client certificates has been created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the CN field of the client certificates. After creating the user, add this user to the Key Admins and Key Groups . Refer to steps mentioned in
User has not been added to Key Admin group	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "authorization denied: verdict was deny: CreateKey"</i>	Check whether the user corresponding to CN of the client certificates has been created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The Username location in Certificate has been set incorrectly	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user,</i>	Check whether the Username Location in the Certificate option is set correctly to CN in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the Username Location in the Certificate.



CipherTrust Integrations

4 February 2026

Contents

- IBM Spectrum Scale 3
 - Pre-Integration 4
 - Integration with CipherTrust Manager 9
 - Verifying Your Integration 15
 - High Availability 16
- Appendix 19

IBM Spectrum Scale

Enterprises and organizations are creating, analyzing, and keeping more data than ever before. Those that can deliver insights faster while managing rapid infrastructure growth are the leaders in their industry. In delivering those insights, an organization's underlying storage must support new-era big data and artificial intelligence workloads along with traditional applications while ensuring security, reliability, and high performance.

IBM Spectrum Scale™ meets these challenges as a high-performance solution for managing data at scale with the distinctive ability to perform archive and analytics in place.

Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

This integration is validated on the following software versions:

CipherTrust Manager

- CipherTrust Manager 2.3 and higher

IBM Spectrum Scale

- IBM Spectrum Scale 5.1.1.1 and higher

Prerequisites

- Spectrum Scale communicates with the CipherTrust Manager through KMIP interface. Ensure that the CipherTrust Manager is installed and configured properly. For more details, refer to the [Administrator Guide of CipherTrust Manager](#).
- Ensure that IBM Spectrum Scale is installed and configured. For more information, refer to the [IBM documentation](#).
- Ensure that the required licenses are activated. For more details, refer to the [CipherTrust Manager documentation](#).
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that KMIP client is registered. Refer [KMIP client Registration](#) for more details.

Pre-Integration

This section lists the following procedure that needs to be performed before integration.

1. [Creating Certificates](#)
2. [Creating Client Keystore \(PKCS#12 file\)](#)
3. [Configuring the RKM.conf File](#)

Creating Certificates

You can create and use the certificate using any of the following methods:

- [Using KSCTL Utility](#): Use this method if you want to create and use certificates generated on the CipherTrust Manager using KSCTL utility.
- [Using openssl](#): Use this method if you want to create and use certificates generated using openssl.

Creating Certificates Using KSCTL Utility

Note

KSCTL is used to create CA and CSRs at the CipherTrust Manager. To know about how to install and configure KSCTL refer to the [CipherTrust Manager Documentation](#)

Creating the CA Certificate

The rootCA is by default available at the CipherTrust Manager. If you want to create a Local CA other than rootCA, use the following command:

```
ksctl ca locals create --cn "Test CA" --csr-outfile csrfile
```

To self-sign the CA for a specific duration, use the id returned in the above command:

```
ksctl ca locals self-sign --id <id of CA> -x <duration>
```

For Example:

```
ksctl ca locals self-sign --id 3593c53b-fbeb-4edb-b84d-c85526ae2f83 -x 365
```

Creating Client Certificate

To create the client certificate, use the following steps:

1. Create CSR and its private key.


```
ksctl ca csr --cn "client" --csr-outfile client.csr --key-outfile client.key
```

This command generates a CSR along with its private key.

Note

In this document, common name CN=client is used, which will be further used as a user name on the CipherTrust Manager.

2. Sign the CSR with CA.

```
ksctl ca locals certs issue --ca-id <id of CA> --csr-infile client.csr -x <duration> -o client
```

For Example:

```
ksctl ca locals certs issue --ca-id c84d66e1-2101-4a46-ad44-3656354b5097 --csr-infile client.csr -x 700 -o client
```

--ca-id is the id which can be obtained from CA details on the Ciphertrust Manager.

Note

--csr-infile is the csr name which is created in the previous step.

Creating Server Certificate

Note

This section is applicable if you want to generate certificates at the CipherTrust Manager and sign them using external CA.

To create a server certificate, you need to create the CSR and its Private key, using the following command:

```
ksctl ca csr --cn "server" --csr-outfile server.csr --key-outfile server.key
```

This command generates a CSR along with its private key.

You can sign the above created CSR using external CA. The generated certificate will be uploaded to the CipherTrust Manager. Refer to [Adding User and Configuring Interface](#) for details.

Creating Certificates Using openssl

You can use openssl to create external CA and CSRs outside CipherTrust Manager.

Before creating the CA and CSRs, ensure that the openssl is configured properly.

- To create the external CA, client, and server certificate, you will need:
 - bash (version 4.4.20 or above)
 - openssl (version 1.1.1 or above)
- `openssl.conf` file is at the following location:
 - `/etc/pki/tls` for RHEL
 - `/etc/ssl` for Ubuntu
- Open `openssl.conf` file and go to [`v3_ca`] tab and add `basicConstraints = critical, CA: true`.

Creating the CA Certificate

To create an External CA and upload it on the CipherTrust Manager, follow the below steps:

1. Create a RSA key using the following openssl command:

```
openssl genrsa -out rootCAKey.pem 2048
```

This command will generate `rootCAKey.pem` file.

2. Create a CA certificate using the generated RSA key.

```
openssl req -x509 -sha256 -new -nodes -key rootCAKey.pem -days 3650 -out rootCACert.pem
```

The above command uses `rootCAKey.pem` generated in previous step to generate the CA Certificate, `rootCACert.pem`.

3. Upload the CA on the CipherTrust Manager, using the following steps:
 - a. Go to **CA > External**.
 - b. On the CA page, click **Add External CA**.
 - c. Paste the contents of the `rootCACert.pem` and click **Save**.

Creating Client Certificate

To create a client certificate, perform the following steps:

1. Create a key to be used in the client certificate.

```
openssl genrsa -out client.key 2048
```

This command generates the RSA key `client.key`.

2. Create a client CSR using the key generated above.

```
openssl req -key client.key -new -sha256 -out client.csr -subj /O=my-org/OU=my-org/OU=client/CN=client
```

In this document, common name `CN=client` is used, which will be further used as a user name on the CipherTrust Manager.

This command uses the `client.key` generated in previous step to generate the client CSR `client.csr`.

3. Sign the CSR using the RSA key generated above and the CA.

```
openssl x509 -req -days 360 -in client.csr -CA rootCACert.pem -CAkey rootCAKey.pem -CAcreateserial -out client.cert -sha256
```

Creating Server Certificate

To create a server certificate, perform the following steps:

1. Create a server key using the following command.

```
openssl genrsa -out server.key 2048
```

2. Create a CSR for the server using the above generated key.

```
openssl req -key server.key -new -sha256 -out server.csr -subj /O=my-org/OU=my-org/OU=server/CN=server
```

3. Create a server certificate using the following openssl command.

```
openssl x509 -req -days 7300 -in server.csr -CA rootCACert.pem -CAkey rootCAKey.pem -CAcreateserial -out server.cert -sha256
```

Creating a Client Keystore (PKCS#12 file)

Create a PKCS#12 file by packaging the client key, client certificate, and CA file using the following openssl command.

```
openssl pkcs12 -export -in client.cert -inkey client.key -certfile rootCACert.pem  
-out client.p12 -name client
```

Note

If you are creating certificates using `ksctl`, then you need to download the client certificate, client key, and Local CA(rootCA) from the CipherTrust Manager and use it while creating the PKCS#12 file.

- You will be prompted to set a password for the keystore file. Specify the **client keystore password**, then the `client.p12` file will be created. This password and the keystore path will be used later in the Remote Key Management configuration file `RKM.conf` of the Spectrum Scale. For simplicity, let's assume that the client keystore password specified for this integration is `abcd`.
- In the above command, the name `client` is applied to the collection of files contained within the PKCS#12 package.

Configuring the RKM.conf File

1. Create a directory `/var/mmfs/etc/RKMcerts` on all nodes of the Spectrum Scale cluster and copy the client keystore `client.p12` created in the [previous step](#) to this directory.
2. Create the Remote Key Management (RKM) configuration file (`/var/mmfs/etc/RKM.conf`) and add a stanza that contains the information necessary to communicate with the CipherTrust Manager. The stanza should be in the following format:

```
client {  
  type = KMIP  
  kmipServerUri = tls://10.164.x.xxx:5696  
  keyStore = /var/mmfs/etc/RKMcerts/client.p12  
  passphrase = abcd  
  clientCertLabel = client  
}
```

Stanza contains the following fields:

- **client:** Name of the RKM stanza. It can be set as desired, "client" here is just an example name for the stanza.

- **type:** Must be KMIP.
- **kmipServerUri:** IP addresses or the DNS names of the CipherTrust Managers and their SSL ports.

In the above example, 10.164.x.xxx is the CipherTrust Manager IP and 5696 is the SSL port.

Note

- For high availability (HA), multiple *kmipServerUri* entries can be added.
- In an event when a CipherTrust Manager fails or is inaccessible, then request is sent to the other CipherTrust Managers configured using *kmipServerUri*.

Refer to [High Availability](#) for High Availability.

- **keyStore:** The name and path of the `client.p12` file (for example: `/var/mmfs/etc/RKMcerts/client.p12`).
 - **passphrase:** The password of the client keystore and client certificate. This is the same password which you specified while creating `client.p12` file.
 - **clientCertLabel:** The label of the client certificate in the client keystore. This is the same label which you specified while creating `client.p12` file.
3. Copy the `/var/mmfs/etc/RKM.conf` and the client keystore `/var/mmfs/etc/RKMcerts/client.p12` file to all nodes in the cluster.

Caution

Ensure the user has READ and WRITE permissions (mode '0600') for the files `var/mmfs/etc/RKM.conf` and `var/mmfs/etc/RKMcerts/client.12`.

Integration with CipherTrust Manager

To integrate Spectrum Scale with the CipherTrust Manager, you need to:

- [Configure the CipherTrust Manager](#)
- [Configure Spectrum Scale](#)

Configuring the CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. Add User
2. Registering a KMIP Client
3. Configure Interface
4. Create Key
5. Update the Key Attributes

Add User

Note

Username should be same as the CN (Common Name) specified while creating the client certificate.

1. In the CipherTrust Manager UI, go to **Access Management -> Users** and create a new user.
2. Select the newly created user and then click **GROUPS**. Further, select the **Key Users** and **Key Admins** Group.

Registering a KMIP Client

You can register a KMIP client on the CipherTrust Manager using:

- Auto Registration
- Manual Registration

Using Auto-Registration

1. Create a registration token using the following steps:
 - a. Log on to the CipherTrust Manager.
 - b. Go to **Access Management > Registration Tokens** in the sidebar.
 - c. Click **Create New Registration Token**.

Copy the **Registration Token** once it is created.

2. Turn ON **Auto Registration** using the following steps:

-
- a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button corresponding to the kmip interface.
 - c. Click **Edit**.
 - d. Under **Configure KMIP** window, select **Auto Registration**.
 - e. Paste the **Registration Token**.
 - f. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - g. Click **Update**.

Using Manual Registration

1. Log on to the CipherTrust Manager.
 2. Go to **Products > KMIP**.
 3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. Paste the content of **client.csr**.
 - f. Click **Save**.
 4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a Name Prefix.
 - c. Click **Select CA**.
 - d. Select CA type as Local if you are using **Local CA** or select external if you are using **External CA**.
 - e. Select appropriate CA from dropdown menu and click **Select Profile**.
 - f. Select the Client Profile from dropdown which you have created in the above step.
 - g. Click **Create Token**.
 - h. Copy the Token created and click **Done**.
-

Note

If you are using External CA then you can select the external CA which was created using openssl and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.

- Specify client name and paste the **Registration Token** generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Configure Interface

Configure a KMIP Interface using the following steps:

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the action button (...) and then click **Edit**.

Configure KMIP popup opens.

3. Check **Auto Registration** option if you are going with Auto Registration or uncheck if you are going with the Manual Registration.
4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

Local CA for Automatic Server Certificate Generation should be set to **Turn off auto generation from Local CA** in case of External CA.

6. Select the CA according to your preference.

- If you are using External CA then select the CA under **External Trusted CAs**
- If you are using Local CA then select the CA under **Local Trusted CAs**

7. This step is required if you are using external CA.

Expand the **Upload Certificate** section:

- In the **Certificate** field, paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space or character or symbol between the contents of these files.
- Select certificate **Format** as **PEM**.
- **Password** field is optional and can be skipped.
- Click **Update**.

Creating Key using API Playground

1. At the top right corner of the main toolbar, click **API** to go to the API playground.
2. Click **Authenticate** on the top right corner. Specify username and password, then click **Post**.

Note

The session will be valid for 300 seconds. You can also authenticate later if you are unable to perform the following steps within this time frame.

3. Go to **Keys** section in the left sidebar. Click **Create - Post**.
4. Specify the following content in the body, and click **POST**. Specify a value equal to or less than 60 characters.

```
{
  "idSize": 60
}
```

Here **idSize:60** will be set to the **ID** of the key, which will be further used while configuring the policy file of the Spectrum Scale.

Caution

Specifying a value greater than 60 leads to integration failure. To view the list of possible errors, go to

the [Troubleshooting](#) section.

Updating the Key Attributes

1. In the CipherTrust Manager UI, go to **Keys** and click the Key you have created using API.
2. Ensure that **Exportable** option is checked.
3. Select the user ([created above](#)) as the **Key Owner**.
4. Click **Update**.

Configuring the Spectrum Scale

1. IBM Spectrum Scale connects with the CipherTrust Manager configured using the `RKM.conf` stanza.

For more details, refer to the [Pre-Integration](#) section.

2. Create a policy file with any desired name, ex `hsm62` in any of the client node. Paste the following content in this policy file.

```
RULE 'p1' SET POOL 'system' /* one placement rule is required at all times */
RULE 'Encrypt all files in file system with rule E1'
SET ENCRYPTION 'E1'
WHERE NAME LIKE '%'
RULE 'rule1' ENCRYPTION 'E1' IS
ALGO 'DEFAULTNISTSP800131A'
KEYS( 'cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56:client' )
```

Note

- `cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56` is the **ID** of the key created on the CipherTrust Manager.
- `client` is the RKM stanza ID in `/var/mmfs/etc/RKM.conf`.
- Ensure that the correct key ID and RKM ID are used in the policy. For details regarding the IBM Spectrum Scale policy rules, click [here](#).

3. Install the encryption policy, using the following command.

```
/usr/lpp/mmfs/bin/mmchpolicy fs0 hsm62
```

Note

fs0 is an IBM Spectrum Scale file system that is created during installation and deployment phase of IBM Spectrum Scale.

Upon successful encryption, following message is displayed:

```
Policy 'hsm62' installed and broadcast to all nodes.
```

If all steps were followed correctly, the integration between CipherTrust Manager and IBM Spectrum Scale will be successful. Both devices can now communicate with each other, and the IBM Spectrum Scale can now encrypt the data present in its nodes using the HSM key.

You can further verify if the integration is successful, using the steps mentioned in [Verifying Your Integration](#).

Verifying Your Integration

Perform the following steps to ensure the integration is successful:

1. Create a test file on any node and then copy it to **/fs0**. Consider that the user created a file **test.txt**, and then copied it to **/fs0**.

```
cp /tmp/test.txt /fs0
```

2. Verify that file is encrypted with the key present in the CipherTrust Manager, using the following commands:

```
[root@ss1 ~]# /usr/lpp/mmfs/bin/mmlsattr -n gpfs.Encryption /fs0/test.txt
```

```
[root@ssl ~]# /usr/lpp/mmfs/bin/mmfsattr -n gpfs.Encryption /fs0/test.txt
file name: /fs0/test.txt
gpfs.Encryption: "EAGC????#???????????????? ????]??,???????????????? ?6(?6?
?B??d?L????l?????????m??f?test.txt aaaaaaaaaacccccccccccccccceerrrrrrff?client?"
EncPar 'AES:256:XTS:FEC:HMF
type: wrapped FEK wrpPar 'AES:KWRAP' CmbPar 'XORHMACSHA512'
aaaaaaaaaaaaaaaaacccccccccccccccceerrrrrrff:client
```

```
[root@ss1 ~]# /usr/lpp/mmfs/bin/mmlsattr -L /fs0/test.txt
```

```
[root@ssl ~]# /usr/lpp/mmfs/bin/mmlsattr -L gpfs.Encryption: /fs0/test.txt
gpfs.Encryption: Stat failed: No such file or directory

file name:                /fs0/test.txt
metadata replication: 1 max 2
data replication:         1 max 2
immutable:                no
appendOnly:               no
flags:
storage pool name:        system
fileset name:              root
snapshot name:
creation time:             Wed May 20 07:03:46 2020
Misc attributes:          ARCHIVE
Encrypted:                 yes
[root@ssl ~]#
```

Since the Encryption flag is true (Encrypted: **yes**), the integration is successful.

High Availability

This section explains the procedure to integrate the CipherTrust Manager present in a cluster with Spectrum Scale.

Note

- The steps mentioned below are to be performed only after performing the steps specified in the [Pre-Integration](#) section.
- To know about cluster creation on the CipherTrust Manager, refer to the [CipherTrust Manager Documentation](#)

Integration Steps

In a high availability setup, primary and backup CipherTrust Manager server(s) can be configured in the `RKM.conf` file.

- The `kmipServerUri` attribute represents the **DNS name** or **IP address** of the CipherTrust Manager and the KMIP SSL port.
- The backup server(s) should be listed in the `kmipServerUri2`, `kmipServerUri3`, etc.

Note

- If you're using External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.
- In case of External CA , update the certificate chain (server cert , CA , server key) in the KMIP interface of the backup server, same as of the primary server.
- For high availability (HA), it is recommended to add multiple backup servers, that is, multiple *kmipServerUri* entries. You can have up to five backup key servers.

The following example has the primary key server and a secondary key server.

1. Add the following RKM (Remote Key Managment) stanza to the `RKM.conf` file (created in the [Pre-Integration](#) section). This stanza contains the information required to connect to the CipherTrust Manager server.

```
RKM_Stanza {
type = KMIP
kmipServerUri = tls://10.164.x.xxx:5696
kmipServerUri2 = tls://10.164.x.abc:5696
keyStore = /var/mmfs/etc/RKMcerts/client.p12
passphrase = abcd
clientCertLabel = client
}
```

Brief description of fields contained in the stanza:

- **RKM_Stanza:** Name of the RKM Stanza. It can be set as desired, "RKM_Stanza" here is just an example name for the stanza.
- **type:** Must be KMIP.
- **kmipServerUri:** IP addresses or the DNS names of the CipherTrust Manager and their SSL ports.
 - For high availability (HA), multiple *kmipServerUri* entries may be added.
 - In an event when a CipherTrust Manager fails or is inaccessible, then request is sent to the other nodes specified here.
- **keyStore:** The name and path of the `client.p12` file (for example: `/var/mmfs/etc/RKMcerts/client.p12`).
- **passphrase:** The password of the client keystore and client certificate. This is the same password which you specified while creating `client.p12` file.

- **clientCertLabel:** The label of the client certificate in the client keystore. This is the same label which you specified while creating `client.p12` file.

2. Install the encryption policy on the file system using the following command.

```
/usr/lpp/mmfs/bin/mmchpolicy fs0 hsm62
```

Verifying Integration

To verify the integration performed using High Availability, follow the below steps:

1. Create a *Key Export* call in the primary CipherTrust Manager of the cluster.
2. Shut down the CipherTrust Manager.
3. Clear the key cache on the IBM Spectrum Scale client.

```
/usr/lpp/mmfs/bin/tsctl encKeyCachePurge 'cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56:client'
```

Note

- `cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56` is the ID for the key created in the CipherTrust Manager.
- `client` is the RKM stanza ID.

Both values can be retrieved by running the command: `mmlspolicy fs0 -L`

For example:

```
[root@ss1 etc]# /usr/lpp/mmfs/bin/mmlspolicy fs0 -L
RULE 'p1' SET POOL 'system' /* one placement rule is required at all time
s */
RULE 'Encrypt all files in file system with rule E1'
SET ENCRYPTION 'E1'
WHERE NAME LIKE '%'
RULE 'rule1' ENCRYPTION 'E1' IS
ALGO 'DEFAULTNISTSP800131A'
KEYS('cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56:client')
```

- Access an existing encrypted file or try to create a new encrypted file. The Spectrum Scale daemon log will show a warning message that indicates its inability to retrieve the encryption key from the primary key server:

```
2020-04-07_11:51:19.637-0400: [W] The key server 10.164.x.xxx (port 5696) had a failure and will be quarantined for 1 minute(s).
```

Spectrum Scale will try the second key server in the list, and the I/O operation should be successful if the secondary/backup server is able to serve the encryption key.

- Check CipherTrust Manager activity logs or records for Key Export calls.

Note

If integration is successful, then KMIP encryption operation is processed by secondary server present in the cluster. That means the *Key Export* calls would be created in secondary server activity logs if the primary is down.

- If the *Key Export* calls are created on the secondary CipherTrust Manager present in the cluster, then the integration is successful.

Appendix

Troubleshooting

You may face following errors while performing the integration steps mentioned in previous sections. The document lists errors from both appliances separately:

IBM Spectrum Scale Errors

Error	Troubleshooting
If the IBM Spectrum Scale client is not able to contact the key server when accessing an encrypted file, the following error message can be found in the daemon log: 2020-04-23_13:06:47.564-0400: [E] Key 'cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56:client' could not be fetched. The TCP connection with the RKM could not be established.	Check the connectivity between the client and the key server.
The following error message is printed in the daemon log when a bad certificate is detected: 2020-04-22_18:13:06.623-0400: [E] Key 'cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56:client' could not be fetched. Bad	Check and ensure the client and server certificates are valid.

Error	Troubleshooting
certificate.	
The following error message indicates that the <code>/var/mmfs/etc/RKM.conf</code> file is missing the RKM stanza "RKM3": <pre>2020-04-28_12:00:23.459-0400: [E] Key 'cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56:client' could not be fetched. The specified RKM ID does not exist; check the RKM.conf settings.</pre>	Ensure that <code>/var/mmfs/etc/RKM.conf</code> on all nodes contains the required RKM stanza.
The following error message indicates that the client key store file specified in the <code>/var/mmfs/etc/RKM.conf</code> file is missing: <pre>2020-05-01_17:20:22.972-0400: [E] Error while validating policy 'for file system dsm64': rc=778: While parsing file '/var/mmfs/etc/RKM.conf':[E] Could not open the key store file ('/var/mmfs/etc/RKMcerts/dsm64Client.p12' was specified).</pre>	Check the <code>/var/mmfs/etc/RKM.conf</code> file and ensure that the correct client keystore file is specified in the <code>keyStore</code> attribute. Verify that the client keystore (<code>.p12</code>) file is present on all nodes in the cluster.
The following error message indicates that the permissions on the client keystore file are incorrect. Access to this file should be granted to root only: <pre>2020-05-01_17:25:24.173-0400: [E] Error while validating policy 'for file system dsm64': rc=778: While parsing file '/var/mmfs/etc/RKM.conf':[E] The keyStore permissions are incorrect for /var/mmfs/etc/RKMcerts/dsm64Client.p12. Access should be only granted to root, and no execute permission is allowed for the file</pre>	Check and fix the permissions on the client keystore file.
If <code>RKM.conf</code> file is not copied to all nodes, following error may occur: <pre>[root@ss1 etc]# /usr/lpp/mmfs/bin/mmchpolicy fs0 hsm62 Validated policy 'hsm62': Parsed 3 policy rules. The RKM identifier specified for key 'cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56:client' does not exist; check the RKM.conf file. Make sure the RKM.conf and keystore(s) files are copied to all nodes in the cluster.</pre>	Copy the <code>RKM.conf</code> file on all nodes.
Setting incorrect or insufficient permissions for <code>RKM.conf</code> file may cause the following error: <pre>[root@ss1 etc]# /usr/lpp/mmfs/bin/mmchpolicy fs0 hsm62 [E] Error while validating policy 'hsm62': rc=778:[E] Incorrect permissions for the configuration file /var/mmfs/etc/RKM.conf on node ss2. mmchpolicy: Command failed. Examine previous error messages to determine cause.</pre>	Set correct permissions for <code>RKM.conf</code> file
If keystore file is missing from <code>/var/mmfs/etc/RKMcerts</code> folder, following error may occur: <pre>[root@ss1 etc]# /usr/lpp/mmfs/bin/mmchpolicy fs0 hsm62 [E] Error while validating policy 'hsm62': rc=778: While parsing file '/var/mmfs/etc/RKM.conf':[E] Could not open the key store file ('/var/mmfs/etc/RKMcerts/client.p12' was specified). mmchpolicy: Command failed. Examine previous error messages to determine cause.</pre>	Make sure that keystore file is in <code>/var/mmfs/etc/RKMcerts</code>

CipherTrust Manager Errors

Error	Troubleshooting
If the username specified while creating a key is not same as the CN (Common Name) specified in the client certificate, following error may occur: <pre>[root@ss1 etc]# cp /root/test.txt /fs0 cp: cannot create regular file '/fs0/test.txt': Operation not permitted [root@ss1 etc]# vim /var/adm/ras/mmfs.log.latest 2020-05-20_08:25:47.961-0400: [E] Unable to create encrypted file test.txt (inode 86532, fileset 0, file system fs0). 2020-05-20_08:25:47.962-0400: [E] Key 'cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56:client' could not be fetched (RKM reported error -1256). "/var/adm/ras/mmfs.log.latest" 537L, 49635C</pre>	Specify the Username same as the CN (Common Name) specified while creating the client certificate.
If mode of the KMIP Interface is selected as TLS, Verify client cert, user must supply password, following error may occur. Error <pre>[root@ss1 etc]# cp /root/test2.txt /fs0 cp: cannot create regular file '/fs0/test2.txt': Operation not permitted [root@ss1 etc]# vim /var/adm/ras/mmfs.log.latest 2020-05-20_08:34:13.693-0400: [W] The key server 10.164.10.118 (port 9889) had a failure and will be quarantined for 1 minute(s). 2020-05-20_08:34:13.924-0400: [W] The key server 10.164.10.123 (port 9889) had a failure and will be quarantined for 1 minute(s). 2020-05-20_08:34:13.924-0400: [E] Unable to create encrypted file test2.txt (inode 86541, fileset 0, file system fs0). 2020-05-20_08:34:13.924-0400: [E] Key 'cd1929c52e4b2e5a8047d6e6815cf41df8d4d664bb78b85991ecce5bce56:client' could not be fetched (RKM reported error -115).</pre>	IBM Spectrum Scale does not support this setting. Choose one of the other two options: • Optional • Not Used
If you are using external CA and primary server fails to serve the key	In KMIP interface, Local CA for Automatic Server Certificate Generation should be set to Turn off auto generation from Local CA.



CipherTrust Integrations

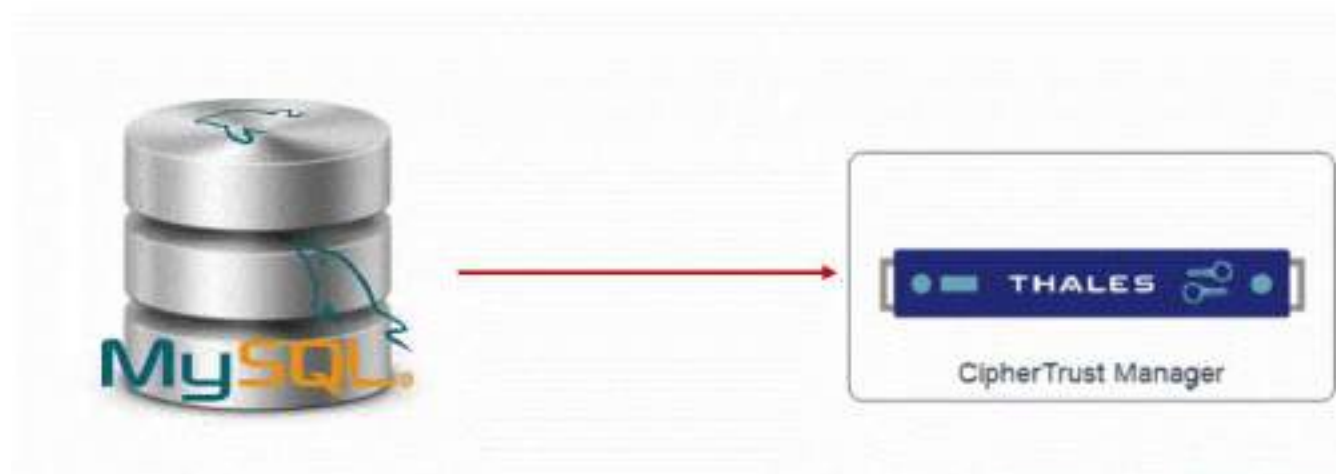
4 February 2026

Contents

- MySQL..... 3
 - Integration with CipherTrust Manager 3
 - Verifying Your Integration 18
 - Appendix..... 19

MySQL

MySQL supports transparent data encryption, which provides data-at-rest encryption for physical tablespace data files. This feature uses a two-tier encryption key architecture, consisting of a master encryption key and tablespace keys. When a table is encrypted, a tablespace key is encrypted and stored in the tablespace header. When an application user wants to access the encrypted tablespace data, a master encryption key is used to decrypt the tablespace key. When integrated with CipherTrust Manager, the master encryption key resides on CipherTrust Manager.



Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

CipherTrust Manager

- CipherTrust Manager 2.3 and higher

MySQL Version

- MySQL 8.0.27 and higher

Operating System

- Red Hat Enterprise Linux 7

Integration with CipherTrust Manager

This section lists the steps to integrate MySQL with CipherTrust Manager.

-
- Prerequisites
 - Configuration on CipherTrust Manager
 - Configuration on MySQL

Prerequisites

This section provides the prerequisites for integration of MySQL with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager documentation](#).
- MySQL communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the MySQL system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the MySQL cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)
4. [Creating or Adding a CA \(optional\)](#)
5. [Registering a KMIP Client](#)
6. [Configuring the KMIP Interface](#)

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

1. Navigate to **Admin Settings > Domains**.
2. Click **Add Domain**. The **Add Domain** page appears.
3. Specify the following information:
 - **Name** - Enter the domain name.

-
- **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.
 - **Parent CA** - Select parent CA as root CA.
 - **Allow Subdomain User Management** - Select this check box if you want to enable the sub-domain user management through this domain.

4. Click **Save**.
5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Sub-Domain

Note

To create a user in sub-domain, you must enable **Allow Subdomain User management**.

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the sub-domain.

-
2. Open the **Access Management** application.
 3. On the left pane, click **Users**. The Users page is displayed.
 4. On the Users page, click **Add User**.
 5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter **Password**. Re-enter the password in the **Password Match** field.
 6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example, **Key Admins** or **Key Users**.
6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.

-
- b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
 4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the knip interface.
 - c. Click **View/Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
-

-
- Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on the CA type that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.

- Specify client name and paste the **Registration Token** that you generated in the above step.

Note

If you are using external CA then you can either upload the client certificate as a file using **Upload Certificate** or paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub-Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.

-
- h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the **Certificate Details** section.

-
- e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.

For domain, the format to enter the Common Name field of the cert is always:

`domainName | | domainUser`

- f. Click **Save**.

4. Create a Registration Token using the following steps:

- a. Go to **Registration Token** and click **New Registration Token > Begin**.
- b. Add a **Name Prefix**.
- c. Specify a Token Lifetime value along with the Client Capacity for the token.
- d. Click **Select CA**.
- e. Select the CA type as Local/External depending on CA that you created [above](#).
- f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
- g. Select the Client Profile from the dropdown which you have created in the above step.
- h. Click **Create Token**.
- i. Copy the value of the Token created and click **Done**.

5. If you are using External CA then you can select the external CA which you created earlier. Refer to **External CA** under [Creating or Adding a CA](#).

6. Go to Registered Clients and click **Add Client**.

- a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- a. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

6. Depending upon the CA that you created [above](#), select an option from the following::
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs**, provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Sub-Domain

1. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.

-
- e. Click **Add External CA**.
 - f. Go to **Admin Settings > Interfaces**.
 - g. Click the **Add** icon to add the External CA.
 - h. Click **Update**.

External CA

Note

If you are using an External CA in the Sub-Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
 - b. Logout of your sub-domain and now login to the root domain.
 - c. Go to **CA > External > Add External CA**.
 - d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
 - e. Click **Add External CA**.
 - f. Go to **Admin Settings > Interfaces**.
 - g. Click the **Add** icon to add the External CA.
 - h. Click **Update**.
 - i. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.
 - j. Select the **Certificate Chain** option and click **Build Certificate Chain**.
 - k. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
 - l. Select certificate Format as **PEM**.
 - m. Click on **Upload Certificate**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
 3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.
-

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

Configuration on MySQL

To integrate MySQL with the CipherTrust Manager:

- Installing the `keyring_okv` plugin
- Creating a Client Certificate
- Importing the KMIP Certificates to `keyring_okv` plugin
- Verify that `keyring_okv` plugin is working

Installing the `keyring_okv` plugin

1. The configuration directory used by `keyring_okv` to store its support files should have restricted permissions, allowing access only to the account running the MySQL server. For instance, on Linux-based systems, you can create a directory structure like: `/usr/local/mysql/mysql-keyring-okv` and set its ownership and permissions by running the following commands as the root user:

```
cd /usr/local
mkdir -p mysql/mysql-keyring-okv/ssl
chmod -R 750 mysql
chown -R mysql mysql
chgrp -R mysql mysql
```

2. To ensure the `keyring_okv` plugin is available during server startup, it must be loaded using the `--early-plugin-load` option. Additionally, the `keyring_okv_conf_dir` system variable should be set to specify the location of the configuration directory for `keyring_okv`.

To do this, edit the `/etc/my.cnf` file and add the plugin under the `mysqld` section:

```
[mysqld]
early-plugin-load=keyring_okv.so
keyring_okv_conf_dir=/usr/local/mysql/mysql-keyring-okv
```

3. In the configuration directory, create a file named `okvclient.ora` file and add the IPs of both the nodes of CipherTrust Manager to it in the following format:

```
SERVER=<IP of Primary Node>:5696
STANDBY_SERVER=<IP of Secondary Node>:5696
```

4. Configure the permissions for these files as follows:

```
cd /usr/local/mysql/mysql-keyring-okv
cd /usr/local/
chmod -R 750 mysql
chown -R mysql
chgrp -R mysql
```

5. If the firewall is active, allow traffic on port 5696. Run the following commands as the root user on the MySQL server:

```
firewall-cmd --zone=public --add-port=5696/tcp --permanent
firewall-cmd --zone=public --add-port=5696/udp --permanent
firewall-cmd --reload
```

6. Disable SELinux on the next server reboot by setting `SELINUX=disabled` in the `/etc/selinux/config` file.

For immediate disabling in the current shell, use the following command:

```
setenforce 0
```

7. After completing the steps above, restart the MySQL server:

```
sudo systemctl restart mysqld
sudo systemctl status mysqld
```

Creating a Client Certificate

Note

This section is applicable to KMIP clients registered using Auto Registration.

1. Log on to the CipherTrust Manager.
2. Navigate to the **CA > Local CA** and click on the name of the CA certificate you need to issue the certificate from.
3. Click on **Issue Certificate**.
4. Enter the **Display Name**, followed by **Common Name**.

Note

- The common Name of the certificate should be the same as the name of the user you created on CipherTrust Manager. Refer to [Creating a User](#) section.
- For more information on the format of common name of the certificate, refer to [LDAP and Multi-Domain Client Usernames in KMIP Certificates](#).

5. Select the **Algorithm** and **Size**, and click **Issue Certificate**.
6. Save the **Private Key** and the **CSR**.
7. Select the **Certificate Purpose** as client, specify the validity of the certificate in days, and click **Issue Certificate**.
8. Download a copy of this certificate by clicking the ellipsis next to the certificate name.
9. Navigate to the **CA > Local CA** and download a copy of the CA that issued the certificate in the previous steps.

Importing the KMIP Certificates to keyring_okv plugin

Tip

Certificates are necessary to enable KMIP communication between the CipherTrust Manager and MySQL.

1. Navigate to the directory `mysql/mysql-keyring-okv/ssl` and import the downloaded copies of all the certificates (CA, Client Certificate, and Key).

Verify that keyring_okv plugin is working

To confirm the keyring_okv plugin installation with the MySQL server running, log in into your MySQL database and run the following query:

```
SELECT PLUGIN_NAME, PLUGIN_STATUS FROM INFORMATION_SCHEMA.PLUGINS WHERE PLUGIN_NAME LIKE 'keyring%';
```

If you receive the plugin name along with the active status in the query response, your installation is successful. If not, investigate the logs at `/var log/mysqlld.log` to ensure there are no connection issues.

Verifying Your Integration

After all the files have been configured, you can verify your integration using either of the following.

Identifying the Encrypted Tablespaces

```
SELECT TABLE_SCHEMA, TABLE_NAME, CREATE_OPTIONS FROM INFORMATION_SCHEMA.TABLES WHERE CREATE_OPTIONS LIKE '%ENCRYPTION%';
```

Encrypting a table using keyring_okv plugin

1. Create a new user and grant all privileges to the user.

```
CREATE USER 'TestUser'@'localhost' IDENTIFIED BY '*****';  
GRANT ALL PRIVILEGES ON *.* TO 'TestUser'@'localhost';
```

2. Log on to MySQL as “TestUser”, and create a database called `Players` and a table named `BaseballPlayers`.

```
mysql -u TestUser -p  
create database Players;  
use Players;  
CREATE table BaseballPlayers (lastname varchar(16), firstname varchar(12), position varchar(16));
```

3. Insert some sample data into the table `BaseballPlayers`.

```
INSERT INTO BaseballPlayers ( firstname, lastname, position ) values ('John', 'Doe', 'Outfield' );
```

```
INSERT INTO BaseballPlayers ( firstname, lastname, position ) values ('Joe','Smith','Pitcher' );
INSERT INTO BaseballPlayers ( firstname, lastname, position ) values ('Willie','Mays', 'Outfield' );
```

4. Encrypt the table.

```
ALTER TABLE BaseballPlayers ENCRYPTION='Y';
```

This generates your master encryption key on the CipherTrust Manager and completes your integration. Move to the next section to verify your integration.

Rotating Master Key

The master encryption key should be rotated periodically or whenever you suspect that the key has been compromised.

To rotate the master key:

1. Run the following query:

```
ALTER INSTANCE ROTATE INNODB MASTER KEY;
```

2. Check the server records on the CipherTrust Manager to verify the status of the operation.

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP auto-registration is not enabled	The following error message is displayed in the CipherTrust Manager Records: "errorMessage": "Unregistered client, please register a new client from CLI or API or UI."	Check whether the auto registration option is selected in Admin Settings > System Interfaces > kmip . If it is not selected, turn on the auto registration option. Refer to Configuration steps for CipherTrust Manager using the GUI or ksctl.
Incorrect KMIP interface mode has been selected	The following error message is displayed in the mongod logs: "initandlisten Unable to retrieve key .system, error: socket exception [CONNECT_ERROR] for The server certificate does not match the host name."	Check whether the mode selected in Admin Settings > System Interfaces > kmip is "TLS, verify client cert, allow anonymous logins". If not, make appropriate changes.

Issue	Error Message	Remediation
Unable to view load the keyring_okv plugin after configuration	<pre> 2021-05-24T16:44:45.524984Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.25-commercial) starting as process 2399 2021-05-24T16:44:45.544113Z 0 [ERROR] [MY-011400] [Server] Plugin keyring_okv reported: 'Error setting the key file.' 2021-05-24T16:44:45.544307Z 0 [ERROR] [MY-011386] [Server] Plugin keyring_okv reported: 'Could not initialize ssl layer' 2021-05-24T16:44:45.544351Z 0 [ERROR] [MY-011377] [Server] Plugin keyring_okv reported: 'keyring_okv initialization failure. Please check that the keyring_okv_conf_dir points to a readable directory and that the directory contains Oracle Key Vault configuration file and ssl materials. Also, please check that Oracle Key Vault is up and running.' </pre>	Restrict permission on the complete set of directories as mentioned in the integration steps. Investigate the logs in <code>/var log/ mysqld.log</code> to make sure there are no errors when connecting to CipherTrust Manager.



CipherTrust Integrations

4 February 2026

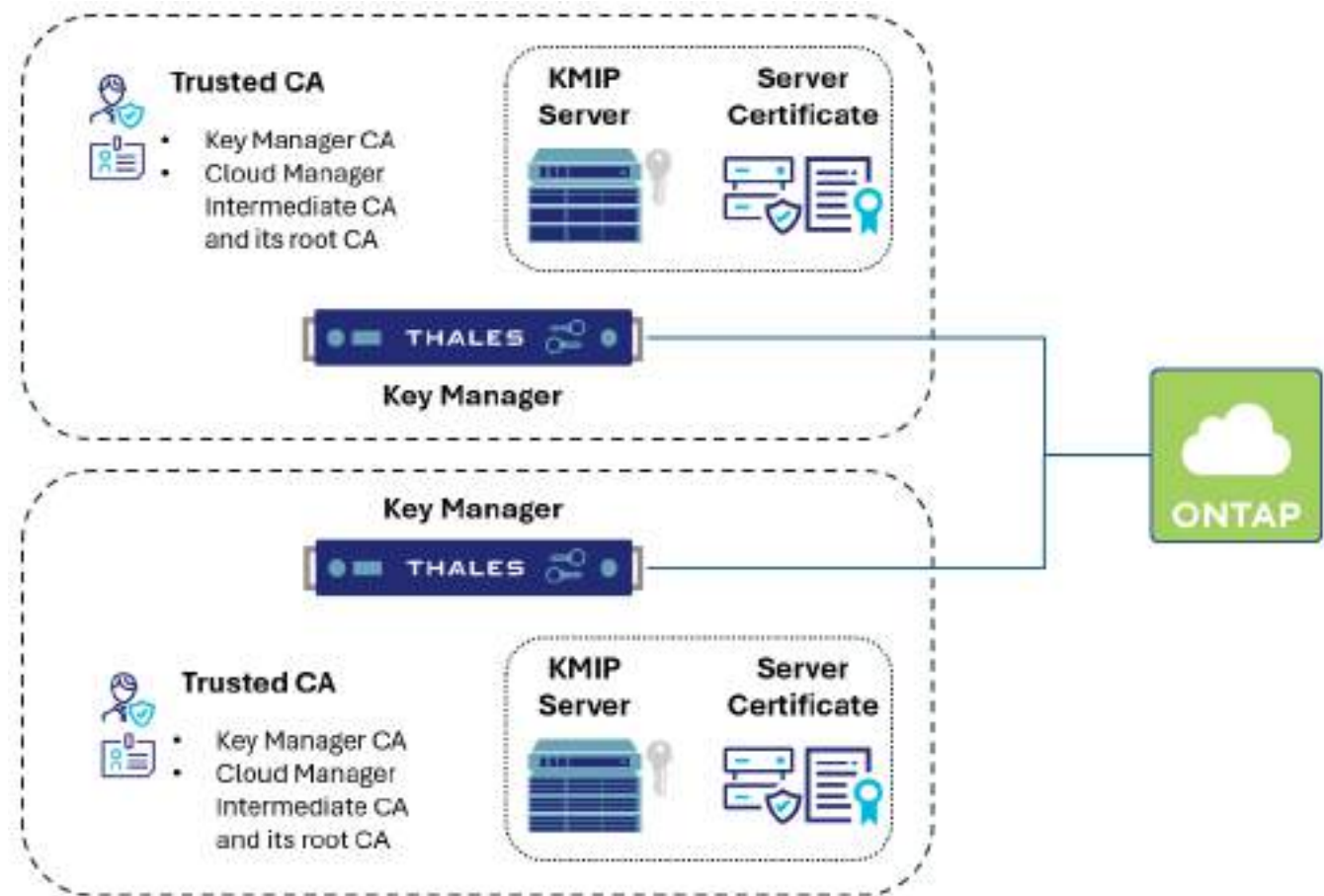
Contents

- NetApp ONTAP Data Management Software 3
 - Pre-Integration 4
 - Integration with CipherTrust Manager 7
 - Verifying Integration 9
 - Appendix. 10

NetApp ONTAP Data Management Software

This document describes how to configure and integrate the CipherTrust Manager in NetApp ONTAP Data Management Software.

ONTAP is an operating system, which is used to store disk arrays such as NetApp FAS/AFF, ONTAP Select, and Cloud Volumes ONTAP.



Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

This integration is validated on the following software versions:

CipherTrust Manager

- CipherTrust Manager 2.3 or higher

ONTAP

- ONTAP version 9.6 or higher

KMIP

- Upto KMIP 1.4

Prerequisites

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager documentation](#).
- ONTAP communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the Ontap system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the Ontap cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.

Pre-Integration

This section lists the steps to be performed before integrating Cloud ONTAP with the CipherTrust Manager.

Creating and Signing Certificates Using KSCTL Utility

Note

KSCTL is used to create CA and CSRs at the CipherTrust Manager. To know about how to install and configure KSCTL refer to the [CipherTrust Manager Documentation](#).

Creating the CA Certificate

The rootCA is by default available at the CipherTrust Manager. If you want to create a Local CA other than rootCA, use the following command:

```
ksctl ca locals create --cn "Test CA" --csr-outfile csrfile
```

To self-sign the CA for a specific duration, use the id returned in the above command:

```
ksctl ca locals self-sign --id <id of CA> -x <duration>
```

For Example:

```
ksctl ca locals self-sign --id 3593c53b-fbeb-4edb-b84d-c85526ae2f83 -x 365
```

Creating the Client Certificate

To create the client certificate, use the following steps:

1. Create CSR and its private key.

```
ksctl ca csr --cn "ontap" --csr-outfile client.csr --key-outfile client.key
```

This command generates a CSR along with its private key.

Note

In this document, common name **CN=ontap** is used, which will be further used as a user name on the CipherTrust Manager.

2. Sign the CSR with CA.

```
ksctl ca locals certs issue --ca-id <id of CA> --csr-infile client.csr -x <duration> -o client
```

For Example:

```
ksctl ca locals certs issue --ca-id c84d66e1-2101-4a46-ad44-3656354b5097 --csr-infile client.csr -x 700 -o client
```

--ca-id is the id which can be obtained from CA details on the Ciphertrust Manager.

Note

--csr-infile is the csr name which is created in the previous step.

Creating the Server Certificate

Note

Server Certificate is needed only in case if you are using external CA instead of Local CA.

To create a server certificate, create CSR and its private key.

```
ksctl ca csr --cn "server" --csr-outfile server.csr --key-outfile server.key
```

This command generates a CSR along with its private key.

You can sign the above created CSR using external CA. The generated certificate will be uploaded to the CipherTrust Manager.

Configuring the CipherTrust Manager

1. Create User on CipherTrust Manager.
 - a. In the CipherTrust Manager UI, go to **Access Management > Users** and create a new user.
 - b. Click on the newly created user, click on **GROUPS** and select the Key Users and Key Admins Group.

Note

Username must be same as common name mentioned above while creating the client certificate.

2. Register KMIP client.

To perform KMIP client registration follow the steps mentioned in the [Appendix](#).

3. Configure KMIP Interface.

- a. Go to **Admin Settings > Interfaces**.
- b. On the KMIP Interface, click the action button (...) and then click **Edit**.

Configure KMIP popup opens.

- c. Check **Auto Registration** option if you are going with Auto Registration or uncheck if you are going with the Manual Registration.
- d. Select the mode as **TLS, verify client cert, username taken from client cert, auth request is optional**.

- e. Specify selections for **Local CA for Automatic Server Certificate Generation and Local Trusted CAs** as desired.

Note

Local CA for Automatic Server Certificate Generation should be set to **Turn off auto generation from Local CA** in case of External CA.

- f. Select the CA according to your preference.
- If you are using External CA then select the CA under **External Trusted CAs**
 - If you are using Local CA then select the CA under **Local Trusted CAs**
- g. This step is required if you are using external CA.

Expand the **Upload Certificate** section:

- In the **Certificate** field, paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space or character or symbol between the contents of these files.
- Select certificate **Format** as **PEM**.
- **Password** field is optional and can be skipped.
- Click **Update**.

Integration with CipherTrust Manager

This section outlines the steps to integrate Cloud ONTAP with the CipherTrust Manager.

Integrate CipherTrust Manager with ONTAP

Connect to the ONTAP instance using SSH. The ONTAP shell appears. Perform the following steps on the ONTAP shell:

1. Install Client Certificate for KMIP Server.

```
::> security certificate install -type client
Please enter Certificate: Press <Enter> when done
-----BEGIN CERTIFICATE-----
...
...
-----END CERTIFICATE-----
```

```

Please enter Private Key: Press <Enter> when done
-----BEGIN RSA PRIVATE KEY-----
...
...
-----END RSA PRIVATE KEY-----
Please enter certificates of Certification Authorities (CA) which form the cer
tificate chain of the client certificate. This starts with the issuing CA cert
ificate of the client certificate and can range up to the root CA certificate.
Do you want to continue entering root and/or intermediate certificates
{y|n}: n
You should keep a copy of the private key and the CA-signed digital certificat
e for future reference.
The installed certificate's CA and serial number for reference:
CA: CA_Name
Serial: xxx
The certificate's generated name for reference: Client_Common_Name

```

Note

Here, the client certificate and key are the ones that you have generated in [Creating the Client Certificate](#) section.

2. Install the `server-ca` certificate for KMIP server.

```

::> security certificate install -type server-ca
Please enter Certificate: Press <Enter> when done
-----BEGIN CERTIFICATE-----
...
...
-----END CERTIFICATE-----
You should keep a copy of the CA-signed digital certificate for future
reference.
The installed certificate's CA and serial number for reference:
CA: CA_Name
Serial: xxxx
The certificate's generated name for reference: ServerCA_Common_Name

```

Note

Here, the server-ca belongs to the Certificate Authority which is used for signing the certs.

3. Install and enable the external key-management setup.

```
::> security key-manager external enable -key-servers CipherTrust-Manager-private-ip:5696 -client-cert Client_Common_Name -server-ca-certs ServerCA_Common_Name
```

Here,

- **CipherTrust-Manager-private-ip** refers to the IP of CipherTrust Manager.
- **Client_Common_Name** refers to the certificate's generated name received after uploading client certificate and key on ONTAP shell.
- **ServerCA_Common_Name** refers to the certificate's generated name received after uploading CA on ONTAP Shell.

4. Verify that external key-management is configured, and its status is available.

```
::> security key-manager external show-status
Node Vserver Key Server Status
-----
Node1
Admin_vserver_name
KMS-ip:5696 available
1 entries were displayed.
```

If the status is set to **available** it means that the CipherTrust Manager is now configured as the external KMS for the ONTAP.

Verifying Integration

Perform the following steps to ensure the successful integration:

After performing the integration steps, you can create keys to verify if the integration is successful.

Execute the following command to create a key from ONTAP:

```
security key-manager key create
```

In the CipherTrust Manager GUI, navigate to **Keys**. Check if there is an entry for this newly created key. Usually, most recently created keys are listed on the top of the table.

If the key is created, the integration is successful.

Appendix

Troubleshooting

This section describes the how to troubleshoot the issue that you may encounter while integrating CipherTrust Manager with ONTAP.

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the KMIP registration is completed or not in Admin Settings > System > Interfaces > kmip . If it is not completed, complete it by using the steps mentioned in Appendix .
User corresponding to username location in certificate (example: CN) has not been created CN = Common Name	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, *</i>	Check whether the user corresponding to CN of the client certificates has been created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the CN field of the client certificates. After creating the user, add this user to the Key Admins and Key Groups . Refer to steps mentioned in
User has not been added to Key Admin group	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "authorization denied: verdict was deny: CreateKey"</i>	Check whether the user corresponding to CN of the client certificates has been created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The Username location in Certificate has been set incorrectly	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user,</i>	Check whether the Username Location in the Certificate option is set correctly to CN in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the Username Location in the Certificate.

KMIP client Registration

Registering a KMIP Client

Note

You need to switch the domain before performing this operation.

You can register a KMIP client on the CipherTrust Manager using:

- [Auto Registration](#)
- [Manual Registration](#)

Using Auto-Registration

1. Create a registration token using the following steps:
 - a. Log on to the CipherTrust Manager.
 - b. Go to **Access Management > Registration Tokens** in the sidebar.
 - c. Click **Create New Registration Token**.
2. Copy the **Registration Token** once it is created.
3. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis icon corresponding to the KMIP interface.
 - c. Click **Edit**.
 - d. Under the **Configure KMIP** window, select **Auto Registration**.
 - e. Paste the **Registration Token**.
 - f. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - g. Click **Update**.

Using Manual Registration

1. Log on to the CipherTrust Manager.
 2. Go to **Products > KMIP**.
-

3. Create a Client Profile using the following steps:

- a. Go to **Client Profile** and click **Add Profile**.
- b. Add a **Profile Name**.
- c. Select **CN** in **Username Location in Certificate**.

Note

For Domain, the CN will be domain||username.

- d. Click **Certificate Details**.
 - e. Paste the content of the generated `client.csr`.
 - f. Click **Save**.
4. Create a Registration Token using the following steps:
- a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a Name Prefix.
 - c. Click **Select CA**.
 - d. Select the CA type as Local if you are using **Local CA** or select external if you are using **External CA**.
 - e. Select appropriate CA from the dropdown menu and click **Select Profile**.
 - f. Select the Client Profile from the dropdown which you have created.
 - g. Click **Create Token**.
 - h. Copy the Token value and click **Done**.

Note

If you are using an external CA then you can select the external CA which was created using openssl and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**. Specify the client's name and paste the generated **Registration Token**.
-

Note

If you are using an external CA then you need to paste the signed client certificate in the **Client Certificate** field.

6. Click **Save > Save Certificate** to save the Client Certificate.



CipherTrust Integrations

4 February 2026

Contents

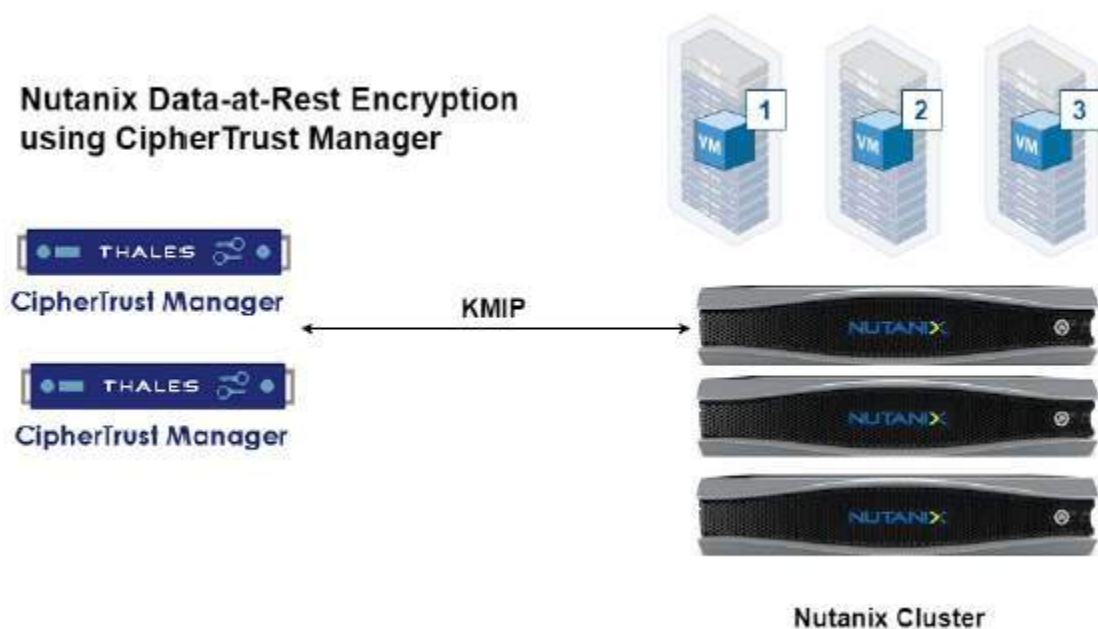
- Nutanix Virtual Computing Platform 3
 - Integration with CipherTrust Manager 4
 - Verifying Your Integration 17
 - Licensing 18
 - Appendix. 18

Nutanix Virtual Computing Platform

Nutanix Virtual Computing Platform (VCP) is a converged infrastructure solution that consolidates the Compute (server) tier and the Storage tier into a single, integrated appliance.

Nutanix VCP enables its users to secure their data when it is at rest using software-only encryption and key-based access management (cluster's native or external Key Management Server (KMS) for software-only encryption). Nutanix VCP supports Key Management Interoperability Protocol (KMIP) for interoperability and strong security.

CipherTrust Manager can be integrated with Nutanix VCP to act as an external KMS, as shown in the following diagram.



Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

CipherTrust Manager

- CipherTrust Manager 1.10 and higher

Nutanix VCP

- AOS version 5.15.x LTS

Key Management Interoperability Protocol (KMIP)

- Upto KMIP 1.4

Integration with CipherTrust Manager

This section lists the steps to integrate Nutanix AHV with CipherTrust Manager.

- Prerequisites
- Configuration on CipherTrust Manager
- Issuing CSR(s) for Each Node of Nutanix AHV
- Getting the CSR(s) signed from CipherTrust Manager
- Configuring CipherTrust Manager as an External KMS on Nutanix AHV
- Enabling encryption on Nutanix AHV

Prerequisites

This section provides the prerequisites for integration of Nutanix AHV with CipherTrust Manager.

- Ensure that CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager Documentation](#).
- IP address of the CipherTrust Manager and port of the KMIP interface must be accessible from the Nutanix cluster.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the Nutanix cluster is registered as a KMIP client on the CipherTrust Manager.

Configuration on CipherTrust Manager

To configure the CipherTrust Manager for the integration, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)
4. [Registering a KMIP Client](#)
5. [Configuring the KMIP Interface](#)

Creating a Domain (Optional)

Perform the following steps on CipherTrust Manager:

Note

This step is optional and needs to be performed only if you want to integrate within a domain.

1. Navigate to **Admin Settings > Domains**.
2. Click **Add Domain**. The **Add Domain** page appears.
3. Specify the following information:
 - **Name**: Enter the domain name.
 - **Admins**: Select the admins (one or more) from the list available in the drop down. For example, admin.
 - **Parent CA**: Select parent CA as root CA.
 - **Allow Subdomain User Management**: Select this check box if you want to enable user management in the subdomain through this domain.
4. Click **Save**.
5. Click on the **current domain name** at the top right corner to switch to this newly created domain.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager UI.
2. Navigate to **Access Management > Users**.
3. On the Users page, click **Create User**.
4. Provide the following details:
 - Username
 - Password
 - Email
5. Click **Add User**. The newly created user is now listed on the Users page.

Sub Domain

To create a user, perform the following steps:

Note

You can only create users in sub domains if you have enabled **Allow Subdomain User Management** while creating a domain.

1. Log on to the CipherTrust Manager UI with the User you created within the Sub Domain.
2. Navigate to **Access Management > Users**.
3. On the Users page, click **Create User**.
4. Provide the following details:
 - Username
 - Password
 - Email
5. Click **Add User**. The newly created user is now listed on the Users page.

Assigning the User to a Group

Perform the following steps to add user to a group:

1. Click on the ellipsis corresponding to the user that you created in the previous step.
2. Click **Edit/View > Groups**.
3. In the search bar, type **Key Admins** or **Key Users** and select **Add**, depending on the level of access you want to grant the user.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Create a Registration Token using the following steps:
 - a. Log on to the CipherTrust Manager in the root domain.
 - b. Go to **Access Management > Registration Tokens**.
 - c. Click **Add New Registration Token > Begin**.
 - d. Add a **Name Prefix** and specify a **Token Lifetime** value along with the **Client Capacity** for the token.
 - e. Click **Select CA**.
 - f. Select the CA type as Local/External depending on your preference.
 - g. Select the appropriate CA from the dropdown and click **Create Token**.
 - h. Copy the value of the **Registration Token** once it is created.
2. Enable **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click on the ellipsis corresponding to the KMIP interface.
 - c. Click **View/Edit**.
 - d. Select the **Auto Registration** checkbox and paste the copied value of the **Registration Token**.
 - e. Select **OU** for the field **Username Location in Certificate**.
 - f. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - g. Scroll down and click **Update** to save your settings.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create a **Client Profile** using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **OU** for the **Username Location in Certificate** field.
 - d. Expand the Certificate Details section.

-
- e. Paste the contents of the CSRs obtained from Nutanix.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a Name Prefix.
 - c. Specify a **Token Lifetime** value along with the **Client Capacity** for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on what your preference.
 - f. Select the appropriate CA from the dropdown menu and click Select Profile.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA which was created using openssl and uploaded on the CipherTrust Manager. For more details, refer to the [CipherTrust Manager Documentation](#).

5. Go to **Registered Clients** and click **Add Client**.
 - Specify client name and paste the **Registration Token** generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save a copy of the **client certificate**.
- Repeat this process for each CSR corresponding to a node on Nutanix AHV.

Sub Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Create a Registration Token in the sub-domain using the following steps:
 - a. Log on to the CipherTrust Manager in your specified subdomain.
 - b. Go to **Access Management > Registration Tokens**.
 - c. Click **Add Registration Token > Begin**.
 - d. Add a **Name Prefix**.
 - e. Specify a value for the **Token Lifetime** along with the **Client Capacity** for the token.
 - f. Click **Select CA**.
 - g. Select the CA type as **Local/External** depending on your requirements.
 - h. Copy the value of the Registration Token once it is created.

Note

By default, Auto Registration is disabled.

1. Enable **Auto Registration** using the following steps:
 - a. Log on to CipherTrust Manager in the root domain.
 - b. Go to **Admin Settings > Interfaces**.
 - c. Click the ellipsis corresponding to the KMIP interface.
 - d. Click **View/Edit**.
 - e. Select the checkbox corresponding to **Auto Registration**.
 - f. Select **OU** for the field **Username Location in Certificate**.
 - g. Paste the copied value of the **Registration Token**.
 - h. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.

-
- i. Scroll down and click **Update** to save your interface configuration.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **OU** in **Username Location in Certificate**.
 - d. Expand the Certificate Details section.
 - e. Paste the content from the CSRs generated by Nutanix.
 - f. Click **Save**.
4. Create a Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a **Token Lifetime** value along with the **Client Capacity** for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as **Local/External** depending on your requirements.
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token and click **Done**.

Note

If you are using External CA then you can select the external CA which was created using openssl and uploaded on the CipherTrust Manager. For more details, refer to the [CipherTrust Manager Documentation](#).

5. Go to Registered Clients and click **Add Client**.
 - a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using External CA then you need to paste the signed client certificate in the Client Certificate field.

- b. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, Auto Registration is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

Local CA for Automatic Server Certificate Generation should be set to **Turn off auto generation from Local CA** in case of External CA.

6. Select the CA according to your preference:
 - If you are using **External CA** then select the CA under **External Trusted CAs**

- If you are using **Local CA** then select the CA under **Local Trusted CAs**
7. If you are using an External CA, expand the **Upload Certificate** section:
 - In the **Certificate** field, paste the contents of Server Certificate, CA, and Server Key file in the same order.
Do not introduce any space or character or symbol between the contents of these files.
 - Select certificate **Format** as **PEM**.
 - **Password** field is optional and can be skipped.
 - Click **Update**.

Sub Domain

1. Switch to **Root Domain**.
2. Go to **Admin Settings > Interfaces**.
3. On the KMIP Interface, click the ellipsis and then click **View/Edit**.
4. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, Auto Registration is disabled.

5. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
6. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

Local CA for Automatic Server Certificate Generation should be set to **Turn off auto generation from Local CA** in case of External CA.

7. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.

- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

External CA

Note

If you are using an External CA in the root Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Go to **Admin Settings > Interfaces**.
- b. Click the **Add** icon to add the External CA.
- c. Click **Update**.
- d. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.
- e. Select the **Certificate Chain** option and click **Build Certificate Chain**.
- f. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
- g. Select certificate Format as **PEM**.
- h. Click on **Upload Certificate**.

Issuing CSR(s) for Each Node of Nutanix AHV

1. Log on to the Nutanix Prism Element, go to **Settings > Data-at-Rest-Encryption**.
2. Click **Edit Configuration**.
3. Under **Select Key Management Server (KMS)**, select **An external KMS** and click **Save KMS Type**.
4. In the **Certificate Signing Request Information** section:

- a. Specify **Email**, **Organization**, **Organizational Unit**, **Country Code**, **City**, and **State**.

Note

The value for the ****OU**** field should be the same as the user you created on CipherTrust Manager in one of the previous steps.

- a. Click **Save CSR Info**.
- b. Click **Download CSRs**.
- c. This will initiate a download of a compressed zip containing all your CSRs. Save this file locally and extract its contents to a folder.

Note

In case of domains, the **Organizational Unit** field for the CSR must be of the format: **domainName||domainUser**.

Certificate Signing Request Information

Enter the following information to generate the Certificate Signing Requests for the cluster.

EMAIL [Input Field]	ORGANIZATION Thales	ORGANIZATIONAL UNIT CiphertrustManager
COUNTRY CODE IN	CITY NO	STATE UP

Save CSR Info Download CSRs

Getting the CSR(s) signed from CipherTrust Manager

Note

In this section, we are using the Local Root CA of CipherTrust Manager to sign the CSRs. However, you

can use any other CA as per your convenience.

Note

This step is applicable only for Auto-Registered KMIP clients.

1. Log on to CipherTrust Manager.
2. Go to **CA > Local**.
3. Click on the ellipsis corresponding to the existing Local CA and click **Download** to save a copy of the CA certificate.
4. Click on the Name of this Local CA and click **Upload CSR**
5. Paste the contents of one of the CSR files, specify a name for the certificate.
6. Select the **Certificate Purpose** as client and click **Issue Certificate**.
7. Repeat the signing process for all the remaining CSRs for each of the nodes.

Configuring CipherTrust Manager as an External KMS on Nutanix AHV

To configure the CipherTrust Manager as an external KMS:

1. On the Nutanix Prism Element UI, go to **Settings > Data at Rest Encryption**.
2. Click **Edit Configuration**.
3. Scroll down to **Key Management Server** and perform the following steps:
 - a. Click **Add New Key Management Server > Add Address**.
 - b. Enter a name for the Key Management Server.
 - c. Enter the IP address and the KMIP port of the CipherTrust Manager.
 - d. Repeat the above steps for each node of CipherTrust Manager if you have a clustered setup.

Add a New Key Management Server

Add Address

Enter a name and at least one address for the Key Management Server.

NAME thales-cm	
ADDRESS 	PORT 5696
ADDRESS 	PORT 5696

◀ Back

Save

4. Scroll down to the **KMS CA Certificates** section and perform the following steps:
 - a. Click **Add New Certificate Authority**.
 - b. Click **Upload CA Certificate**.
 - c. Specify a name for the CA.
 - d. Click **Save**.
5. Scroll down to the **Key Management Server** section and perform the following steps:
 - a. Click **Manage Certificates** for the desired key management server. The **Manage Signed Certificates** screen is displayed.
 - b. Upload the node certificates. Perform either of the following:
 - Click **Upload Files** and upload all the certificates at once.
 - Click the **Upload** link for each node separately.
 - c. Test whether the certificates are correct. Perform either of the following:
 - Click **Test all nodes** to test the certificates for all nodes at once.
 - Click the **Test CS (or Re-Test CS)** link for each node separately.
 - Confirm that the status has changed to **Verified**.

Enabling encryption on Nutanix AHV

- Log on to the Nutanix Prism Element UI.
- Click on the **Settings** icon on the toolbar's right side to open the Settings menu.
- Navigate to **Data-at-rest Encryption** under **Security** in the left pane of the Settings.
- Toggle on the Encryption option.
- In the confirmation pop-up window, type ENCRYPT and then click **Encrypt**.
- This will trigger the encryption of the Nutanix cluster. This step usually takes some time to complete. Once done, the cluster encryption is configured and enabled.

Verifying Your Integration

To verify the successful integration:

1. On the Nutanix VCP UI, go to **Data at Rest Encryption** settings.
2. Select **Enable Encryption**, enter **ENCRYPT** and select **Encrypt**.

Note

This step usually takes a long time, upto few hours. After completion, the key in the settings turns golden and displays the message **Encryption State of Cluster: Software encryption is enabled**.

After the encryption is complete, verify that the keys are created on the Ciphertrust Manager.

3. Test whether the keys are being retrieved from the CipherTrust Manager. To do so:
 - a. Power off the VMs running in the encrypted Nutanix cluster, Nutanix Controller VMs (CVMs), hosts, and CipherTrust Manager VMs.

A Nutanix CVM runs the Nutanix software and serves all the I/O operations for the hypervisor and VMs running on that host.
 - b. Power on the hosts and CVMs.
 - c. Power on the VMs in the encrypted Nutanix cluster.

The power on operation should succeed only when any of the configured CipherTrust Managers are running.

4. Check whether the stored keys are being replaced for the **Rekey** operation. To do so:

- a. In the **Manage Keys** section, select the **Rekey** option.

A message **Encryption State of Cluster: Encryption keys were successfully changed** is displayed.

- b. Test whether the keys can be used for decryption. To do so:
 - a. Power off the VMs running in the encrypted Nutanix cluster, CVMs, hosts, and CipherTrust Manager VMs.
 - b. Power on the hosts and CVMs.
 - c. Power on the VMs in the encrypted Nutanix cluster.
- c. Check the CipherTrust Manager cluster logs to verify whether the keys are exported from the active CipherTrust Manager.

Licensing

This section provides the licensing requirements for KMIP clients and Nutanix clusters.

Condition	License Requirement
Encrypting the Self Encrypting Drives (SEDs)	x1 KMIP client license for each node in a cluster/block
Adding a Node to an existing cluster	x1 KMIP client license for each additional node

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP auto-registration is not enabled	The following error message is displayed in the CipherTrust Manager Records: "errorMessage": "Unregistered client, please register a new client from CLI or API or UI."	Check whether the Auto registration option is selected in Admin Settings > System Interfaces > kmip . If it is not selected, turn on the Auto registration option. Refer to Configuration steps for CipherTrust Manager using the GUI or ksctl.
User corresponding to OU/UID has not been created	The following error messages are displayed in the CipherTrust Manager Records: "errorMessage": "username not found: <username> "errorMessage": "Could not authenticate certificate user, hence client is not auto	Check whether the user corresponding to the UID/OU of the node certificates has been created in Keys & Access Management > Users . If the user does not exist, create a new user with the same name as the OU/UID field of the node certificates. Refer to Configuration steps for

Issue	Error Message	Remediation
	registered in mode tls-cert-pw-opt"	CipherTrust Manager using the GUI or ksctl. After creating the user, add it to the Key Users Group.
User has not been added to the Key Users group	The following error message is displayed in the CipherTrust Manager Records: "errorMessage": "authorization denied: verdict was deny: CreateKey"	Check whether the user corresponding to the OU/UID field of node certificates has been added to the Key Users group in the GUI at Keys & Access Management > Users > > Groups . If the user is not added, add it to the Key Users group. Refer to Configuration steps for CipherTrust Manager using the GUI or ksctl.
The Username Location in Certificate has been set incorrectly	The following error messages are displayed in the CipherTrust Manager Records: "errorMessage": "username not found: <username>" "errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"	Check whether the Username Location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is incorrect, set the correct value for Username Location in the Certificate. Refer to Configuration steps for CipherTrust Manager using the GUI or ksctl.



CipherTrust Integrations

4 February 2026

Contents

Rubrik. 3

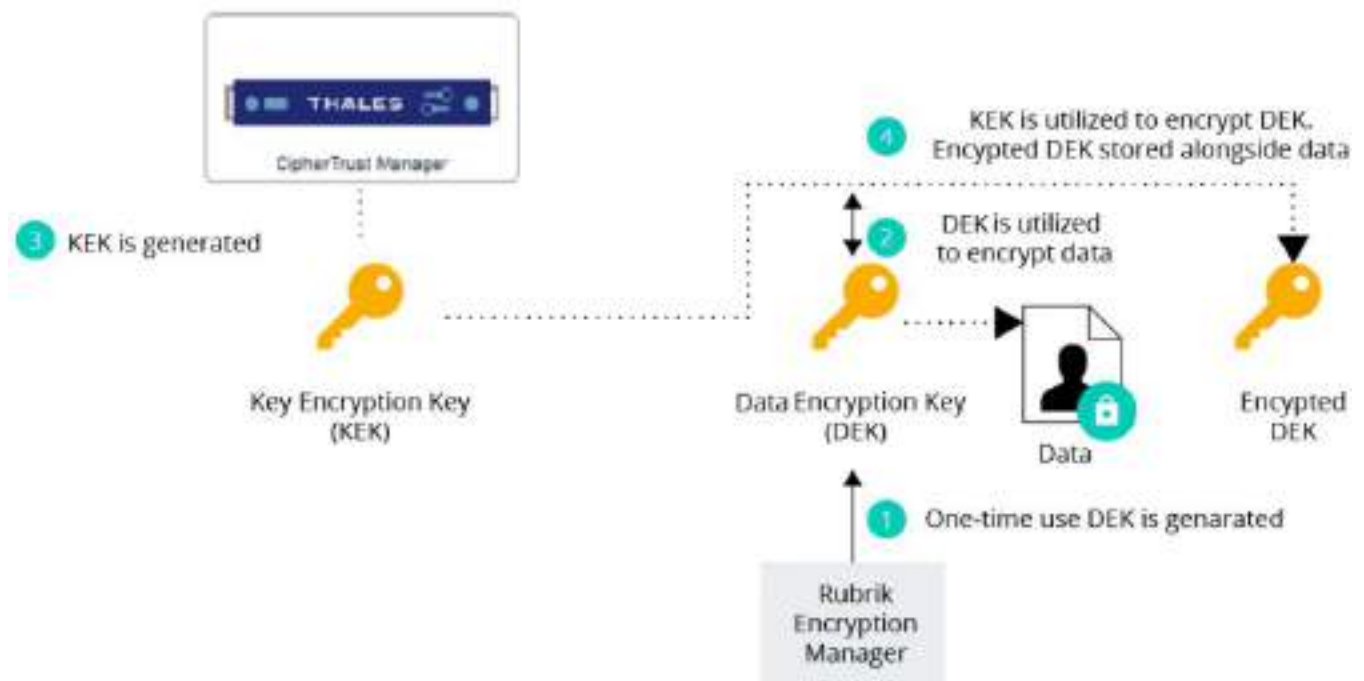
 Integration with CipherTrust Manager 4

 Appendix. 17

Rubrik

Rubrik software encryption for data at rest works on the application level, meaning all the work and encryption is performed through software. As data is ingested, Rubrik generates a unique one-time-only symmetric Data Encryption Key (DEK). DEKs utilize the AES-256 cipher and are utilized to encrypt both the ingested backup data, along with any associated metadata generated by Rubrik. This process ensures that without the associated DEK the data is essentially unreadable, even by the Rubrik platform. This results in the need to store this associated DEK somewhere alongside the data itself.

Therefore when Rubrik needs to read this data, it must pass through two phases of encryption, first decrypting the DEK with a KEK, then decrypting the data itself. To obtain and manage KEK's, customers can leverage an external Key Management Interoperability Protocol (KMIP) compliant key management server such as the CipherTrust Manager.



Supported Product Versions

Note

This integration has been validated in the field by a partner or in a customer environment with the following software versions. It is recommended to test the integration in a non-production environment with desired software versions before deploying it to production. Thales will provide best-effort support.

CipherTrust Manager

- CipherTrust Manager 2.3

Rubrik

-
- [Rubrik 6.0](#)
 - [Rubrik 7.0](#)

Integration with CipherTrust Manager

This section provides the process of integrating Rubrik with the CipherTrust Manager.

- [Prerequisites](#)
- [Configuration on CipherTrust Manager](#)
- [Configuration on Rubrik](#)

Prerequisites

This section provides the prerequisites for integration of Rubrik with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager Documentation](#).
- Rubrik communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager.
- IP address of the CipherTrust Manager and port of the KMIP interface must be accessible from the Rubrik system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the Rubrik cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)
4. [Creating or Adding a CA \(optional\)](#)
5. [Registering a KMIP Client](#)
6. [Configuring the KMIP Interface](#)

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

-
1. Navigate to **Admin Settings > Domains**.
 2. Click **Add Domain**. The **Add Domain** page appears.
 3. Specify the following information:
 - **Name** - Enter the domain name.
 - **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.
 - **Parent CA** - Select parent CA as root CA.
 - **Allow Subdomain User Management** - Select this check box if you want to enable the sub-domain user management through this domain.
 4. Click **Save**.
 5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Sub-Domain

Note

To create a user in sub-domain, you must enable **Allow Subdomain User management**.

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the sub-domain.
2. Open the **Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example, **Key Admins** or **Key Users**.
6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**

- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **View/Edit**.
-

-
- d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on the CA type that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.

-
- h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.

- Specify client name and paste the **Registration Token** that you generated in the above step.

Note

If you are using external CA then you can either upload the client certificate as a file using **Upload Certificate** or paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub-Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:

-
- a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the **Certificate Details** section.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.

For domain, the format to enter the Common Name field of the cert is always:
`domainName | | domainUser`

- f. Click **Save**.
4. Create a Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.
5. If you are using External CA then you can select the external CA which you created earlier. Refer to **External CA** under [Creating or Adding a CA](#).
6. Go to Registered Clients and click **Add Client**.
 - a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- a. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

6. Depending upon the CA that you created [above](#), select an option from the following::
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs**, provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Sub-Domain

1. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

External CA

Note

If you are using an External CA in the Sub-Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.
- i. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.

-
- j. Select the **Certificate Chain** option and click **Build Certificate Chain**.
 - k. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
 - l. Select certificate Format as **PEM**.
 - m. Click on **Upload Certificate**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
 3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

Configuration on Rubrik

1. [Uploading CA Certificate to Rubrik](#)
2. [Configuring KMIP Server on Rubrik](#)

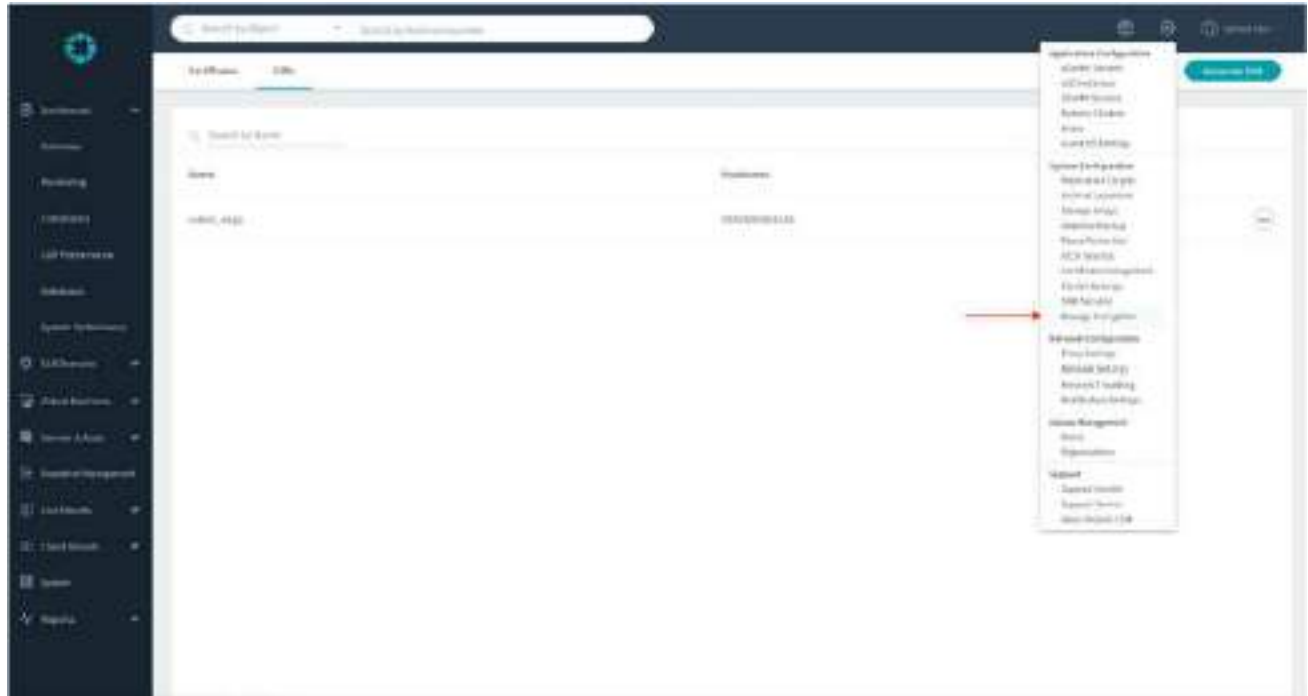
Uploading CA Certificate to Rubrik

1. Copy the local **CA certificate** to clipboard.
2. In Rubrik **Certificate Management Console**, add the CA certificate that you generated from CipherTrust Manager by pasting the certificate from the clipboard to the **Certificate** field. Specify the **Display Name**, and click **Add**.

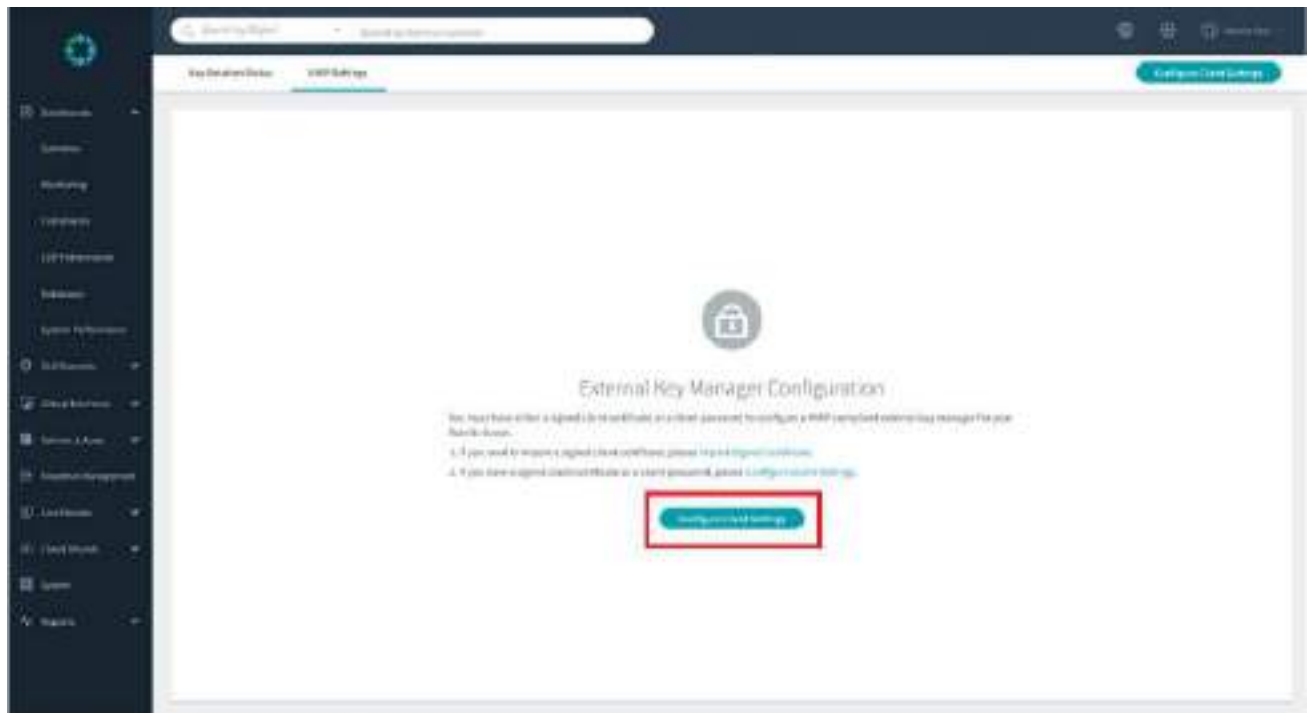
Now you can view both client certificate and CA certificate in the **Certificates** list.

Configuring KMIP Server on Rubrik

1. Select **Manage Encryption** on the System Configuration page.



2. Under the **KMIP Settings** tab, click **Configure Client Settings**.



Configure Client Settings

Client Authentication Mode

☐ Password Only
 ☒ Client Certificate Only
 ☐ Both

Username
rubrik_edge

Select a TLS Certificate. If you have not imported your TLS Certificate, import it from the [Certificate Management page](#).

rubrik_edge X

Select certificate from list or type certificate name

Cancel Update

- On the **Add KMIP Server** page, enter the **KMIP Server Address** and **Port** of the KMIP server. Select a TLS certificate, and click **Add**.

After successful configuration, an AES key is generated that will appear on the list available under **CipherTrust Manager > Keys**.

This completes the integration of Rubrik with the CipherTrust Manager.

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the Autoregistration option is selected in Admin Settings > System > Interfaces > kmip . If it is not selected, turn on the Autoregistration option. Refer to the steps mentioned in Enable KMIP Client Registration .
User corresponding to username location in the certificate (for example, OU/UID) is not created	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence the client is not auto registered in</i>	Check whether the user corresponding to UID/OU of the node certificates is created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the OU/UID field of the node

Issue	Error Message	Remediation
	<code>mode tls-cert-pw-opt"</code>	certificates. After creating the user, add this user to the Key Admins . Refer to the steps mentioned in Create a New User .
User is not added to the Key Admin group	The following error message is displayed in CipherTrust Records: "errorMessage": "authorization denied: verdict was deny: CreateKey"	Check whether the user corresponding to UID/OU of the node certificates is created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The username location in the Certificate is set incorrectly	The following error messages are displayed in CipherTrust Records: "errorMessage": "username not found: " "errorMessage": "Could not authenticate certificate user, hence the client is not auto registered in mode tls-cert-pw-opt"	Check whether the username location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the username location in the Certificate.



CipherTrust Integrations

4 February 2026

Contents

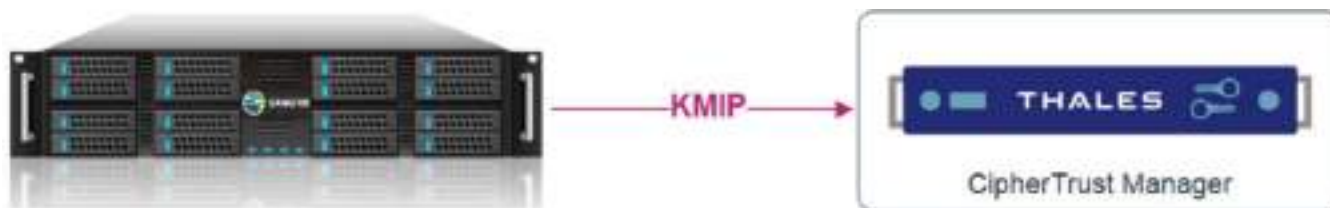
- Sangfor HCI 3
 - Integration of Sangfor HCI 3
 - Verifying Integration 22
 - Appendix..... 23

Sangfor HCI

This document describes how to configure and integrate the CipherTrust Manager with Sangfor HCI.

Sangfor Hyper-Converged Infrastructure (HCI) provides an innovative third generation cloud computing architecture that reduces at least 70% of the Total Cost of Ownership (TCO), simplifies operations, and enhances the network security. It converges compute, storage, networking, and security on a single software stack.

- One-stop software-defined data centre solution
- Easy operation and quick installation for business-critical applications
- Integrates with any commodity servers commercially available in the market
- Build your own private cloud, extend to a public cloud or create your own Hybrid Cloud



Supported Product Versions

Note

This integration has been validated in the field by a partner or in a customer environment with the following software versions. It is recommended to test the integration in a non-production environment with desired software versions before deploying it to production. Thales will provide best-effort support.

This integration is validated on the following software versions:

CipherTrust Manager

- CipherTrust Manager 2.3 or higher

Sangfor

- Sangfor version 9.6 or higher

Integration of Sangfor HCI

This section lists the steps to integrate Sangfor HCI with CipherTrust Manager.

- Prerequisites

-
- Configuration on CipherTrust Manager
 - Configuration on Sangfor HCI

Prerequisites

This section provides the prerequisites for integration of Sangfor HCI with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager Documentation](#).
- Sangfor HCI communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the Sangfor HCI system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the Sangfor HCI cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.
- Ensure that Sangfor HCI has proper license for data at Rest Encryption. To check the license information in Sangfor HCI, navigate to **Settings > Software and Licenses > License Information**.
- Ensure that the required licenses are activated on CipherTrust Manager. For more details, refer to the [CipherTrust Manager documentation](#).

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)
4. [Creating or Adding a CA \(optional\)](#)
5. [Registering a KMIP Client](#)
6. [Configuring the KMIP Interface](#)

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

1. Navigate to **Admin Settings > Domains**.

-
2. Click **Add Domain**. The **Add Domain** page appears.
 3. Specify the following information:
 - **Name** - Enter the domain name.
 - **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.
 - **Parent CA** - Select parent CA as root CA.
 - **Allow Subdomain User Management** - Select this check box if you want to enable the sub-domain user management through this domain.
 4. Click **Save**.
 5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Sub-Domain

Note

To create a user in sub-domain, you must enable **Allow Subdomain User management**.

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the sub-domain.
2. Open the **Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example, **Key Admins** or **Key Users**.
6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**

- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **View/Edit**.
-

-
- d. On the **Interface Detail** window, perform the following:
- Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on the CA type that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.

-
- h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.

- Specify client name and paste the **Registration Token** that you generated in the above step.

Note

If you are using external CA then you can either upload the client certificate as a file using **Upload Certificate** or paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub-Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:

-
- a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the **Certificate Details** section.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.

For domain, the format to enter the Common Name field of the cert is always:
`domainName | | domainUser`

- f. Click **Save**.
4. Create a Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.
5. If you are using External CA then you can select the external CA which you created earlier. Refer to **External CA** under [Creating or Adding a CA](#).
6. Go to Registered Clients and click **Add Client**.
 - a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- a. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

6. Depending upon the CA that you created [above](#), select an option from the following::
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs**, provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Sub-Domain

1. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

External CA

Note

If you are using an External CA in the Sub-Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.
- i. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.

-
- j. Select the **Certificate Chain** option and click **Build Certificate Chain**.
 - k. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
 - l. Select certificate Format as **PEM**.
 - m. Click on **Upload Certificate**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
 3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

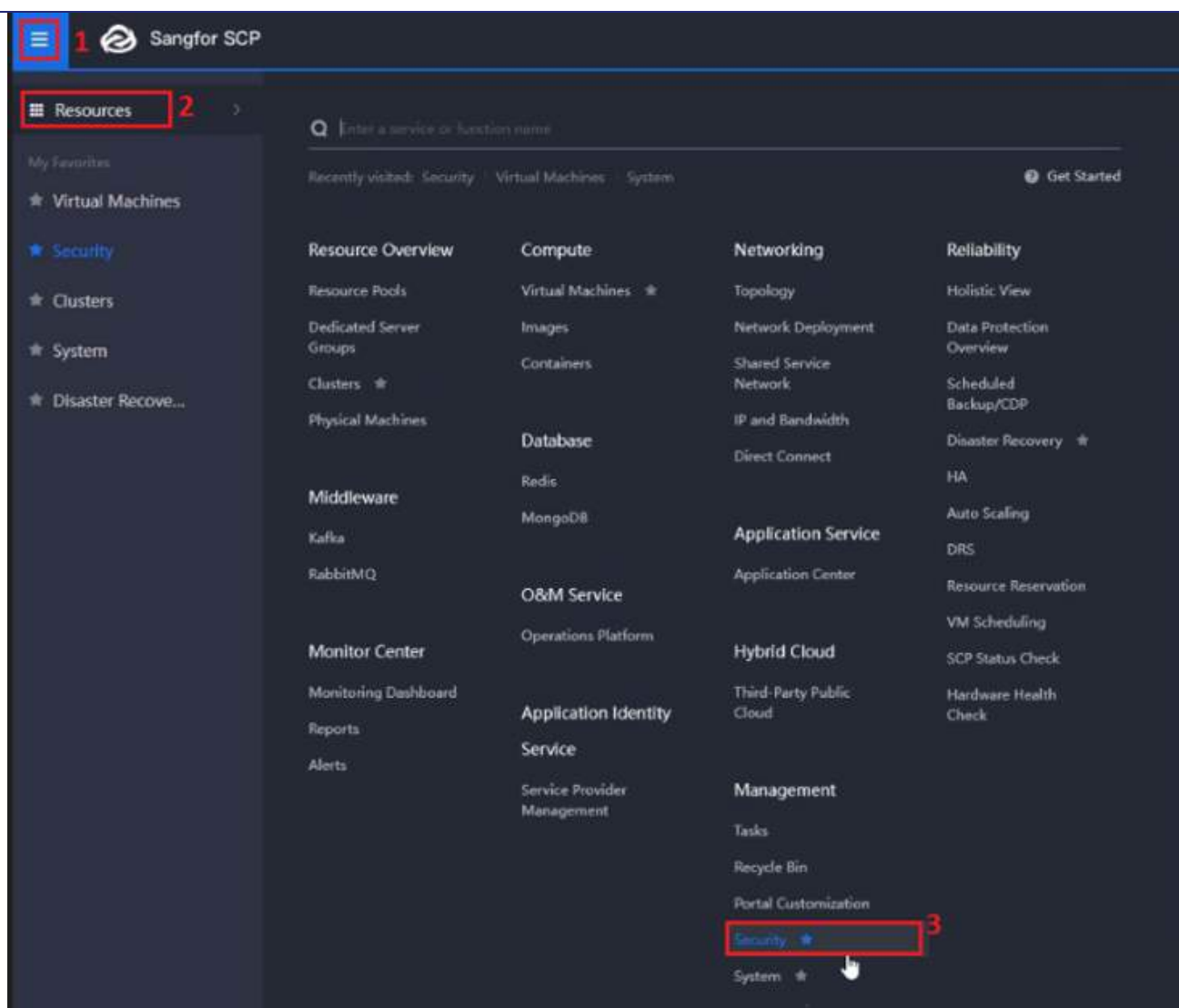
In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

Configuration on Sangfor HCI

Add a KMS Server to the Sangfor HCI Server

To add KMS, perform the following steps:

1. Log in to the **Sangfor Cloud Platform** web console.



2. Navigate to the **Security** page and select **Key Management Service**.
3. On the **Key Management Service** page, select **KMS Server**, and click **Add**.



4. Provide the following KMS Server information:

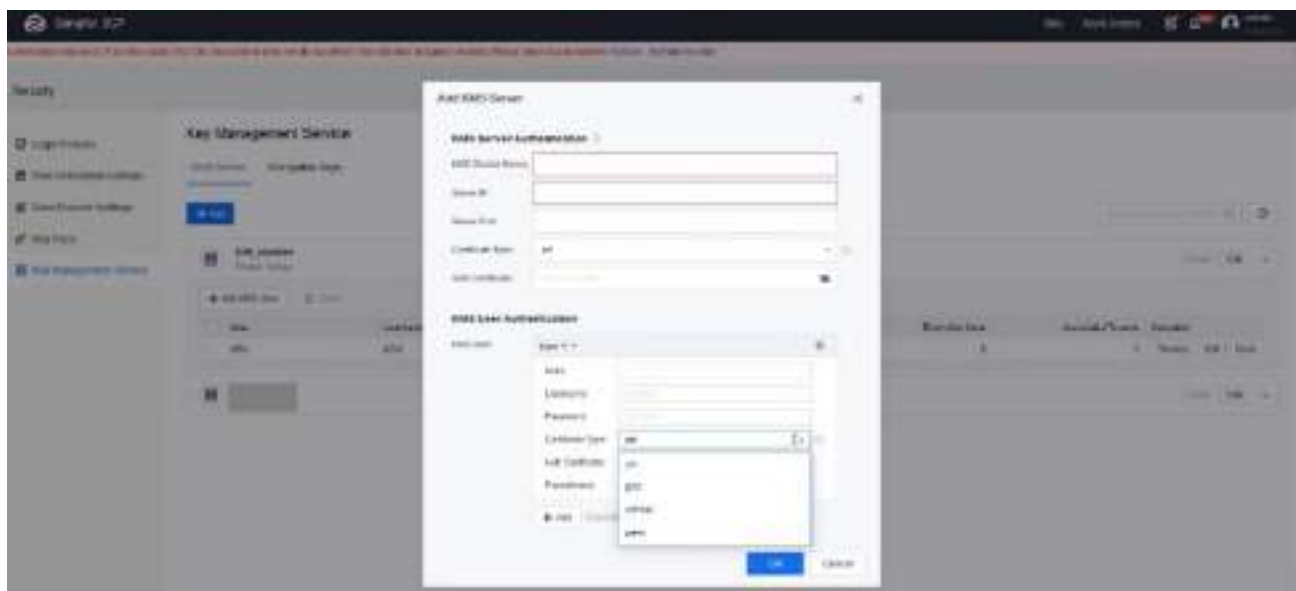
- **KMS Server Authentication**

- KMS Cluster Name: CM_Cluster
- Server IP: 18.167.165.248 (CipherTrust Manager IP address)
- Server Port: 5696 (CipherTrust Manager default KMIP interface)
- Certificate Type: pem
- Auth Certificate: upload "Certificate.pem"

- **KMS User Authentication**

- Alias: sfhci x
- Username: sfhci
- Password: (the one created at CipherTrust Manager for user sfhci)
- Certificate Type: pem
- Auth Certificate: upload "Bundle.pem"

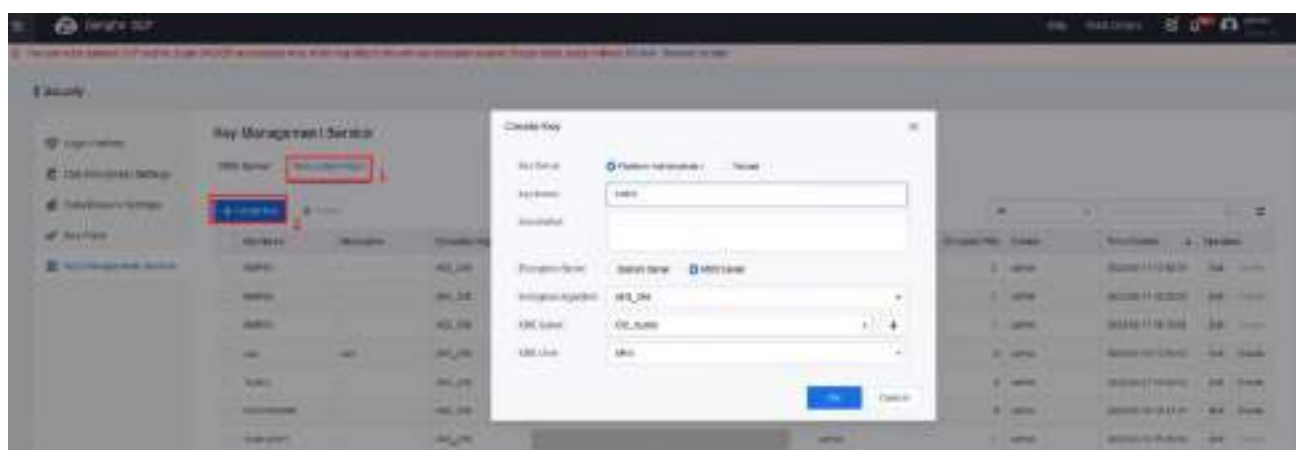
5. Click **OK**



Create an AES encryption key

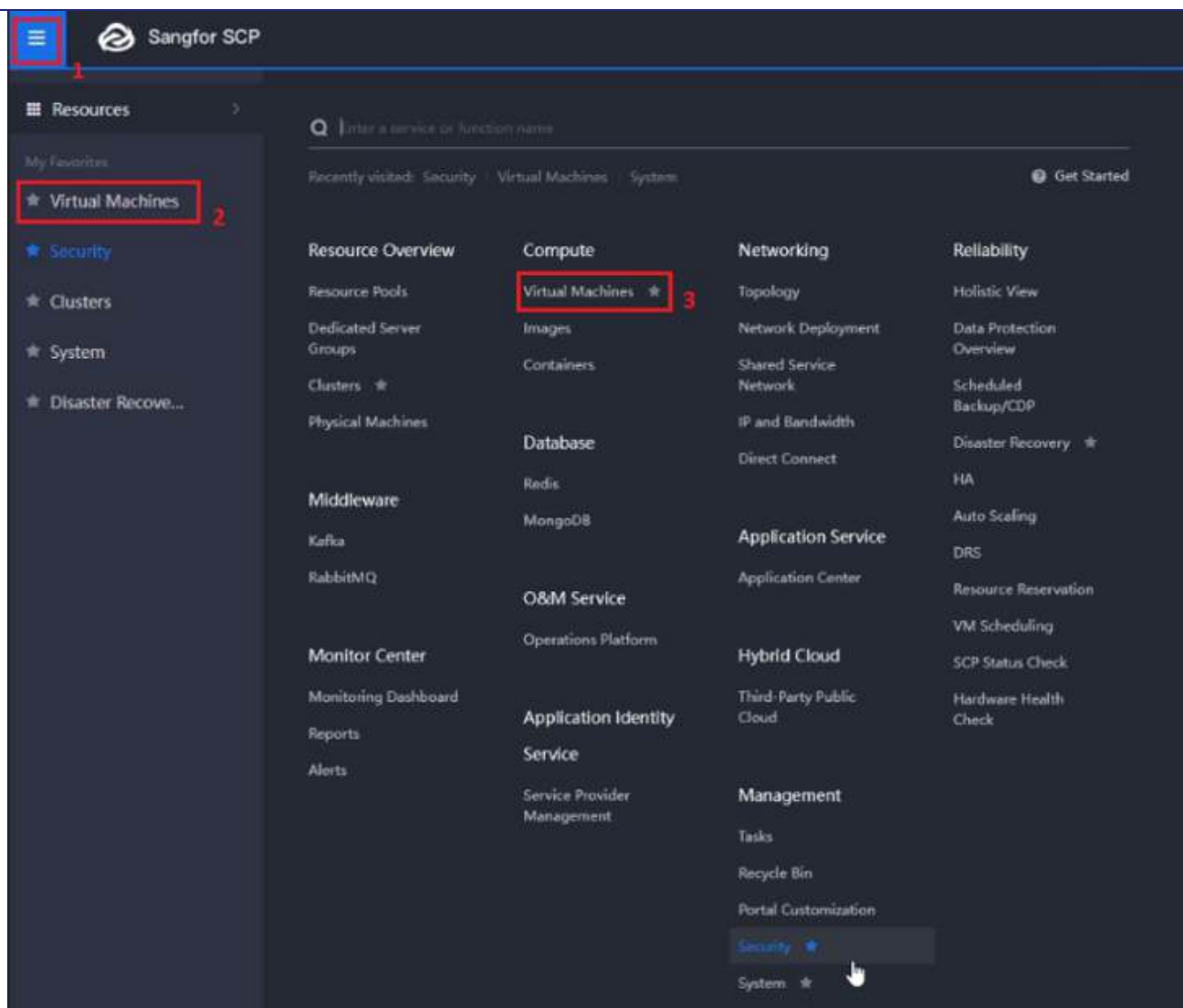
- On the **Key Management Service** page, select **Encryption Keys**, and click **Create Key**.
- Provide the following KMS Server information:
 - Key Name: CM01 (Any name)

- b. Encryption Server: KMS Server
 - c. Encryption Algorithm: AES_256
 - d. KMS Server: CM_cluster
 - e. KMS User: sfhci
3. Click **OK**.

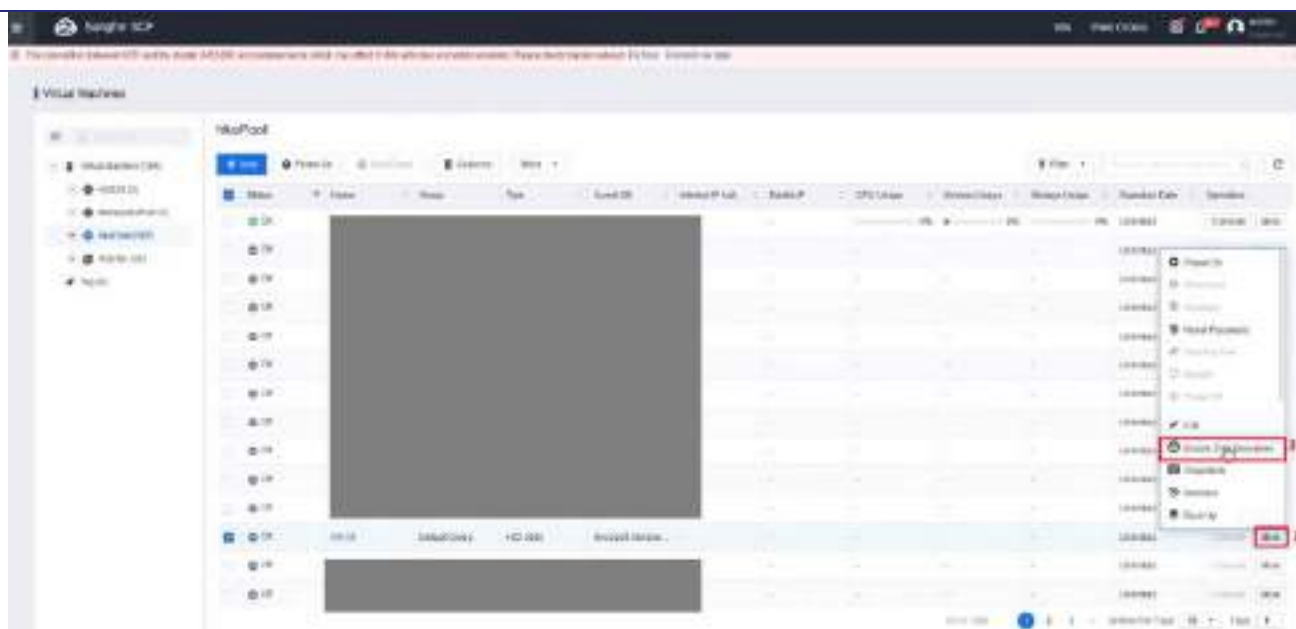


Encrypt the Virtual Machine

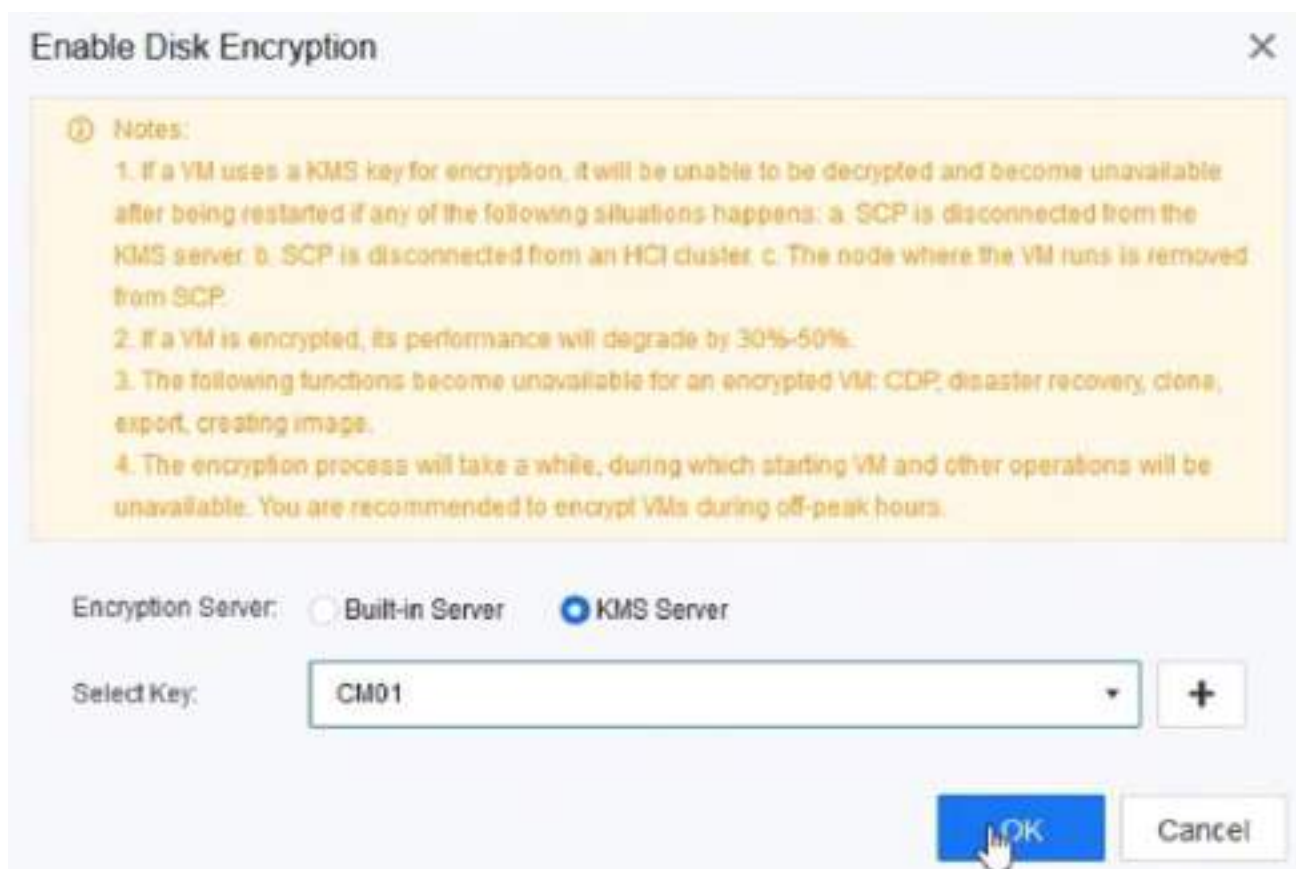
1. Navigate to the **Virtual Machine** page. Ensure the virtual machine is in power off status.



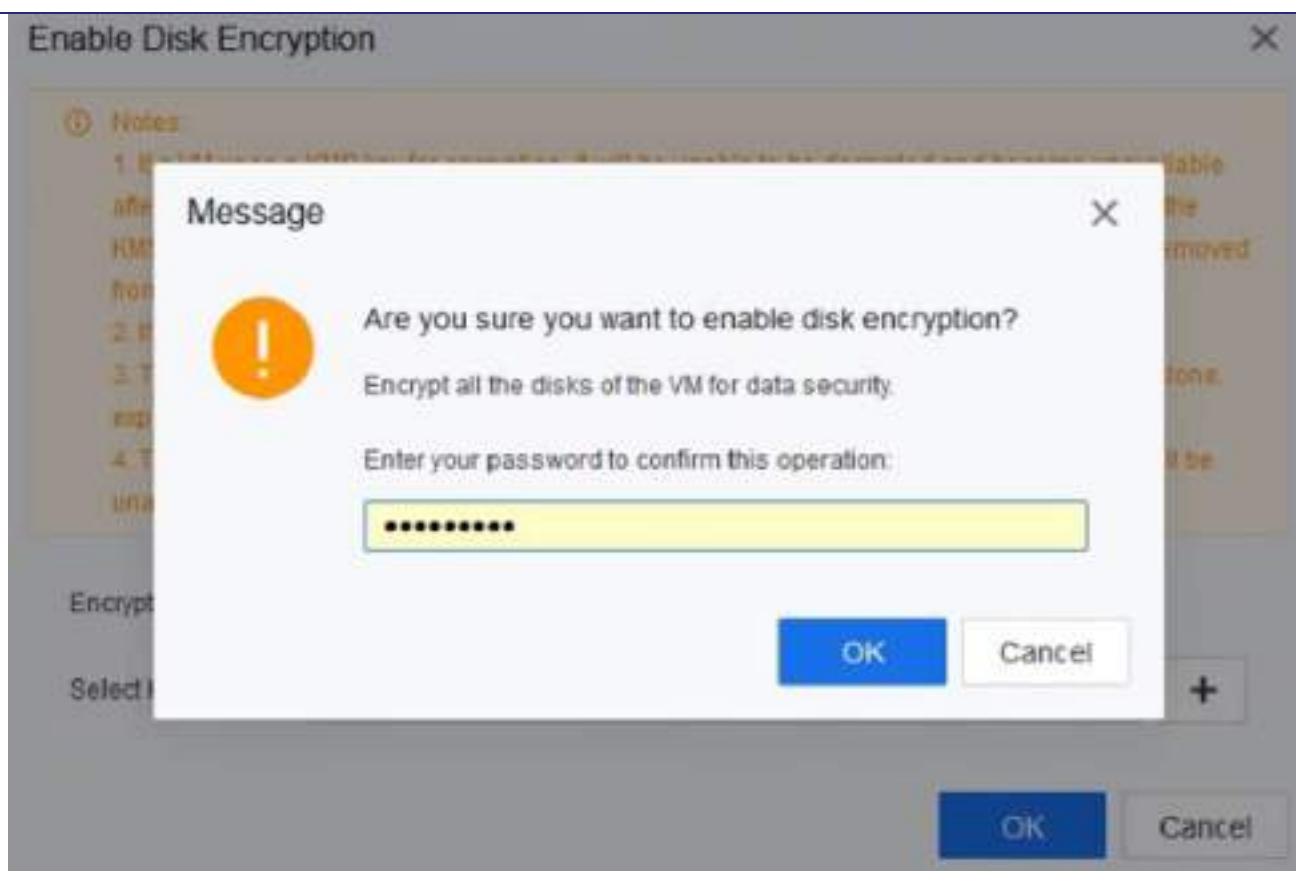
2. Click **More** and select **Enable Disk Encryption**. The **Enable Disk Encryption** window pops up.



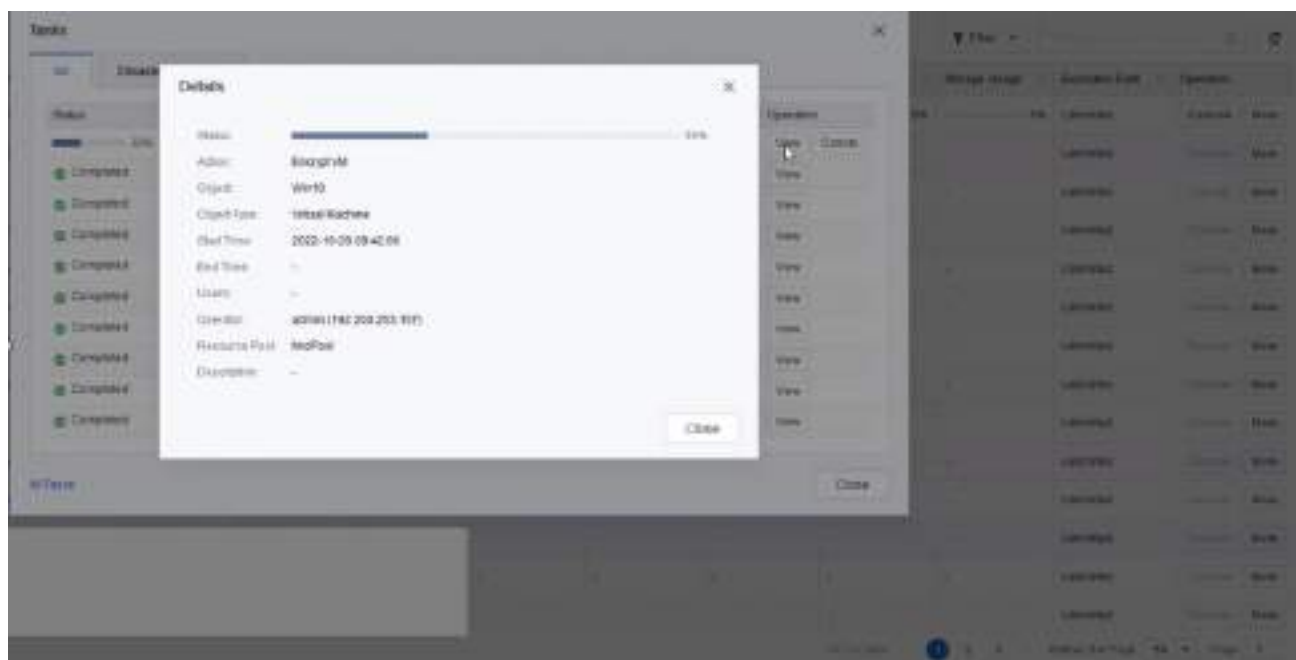
3. Select **KMS Server** as **Encryption Server**.
4. Select the key from the drop down menu. Click **OK**.



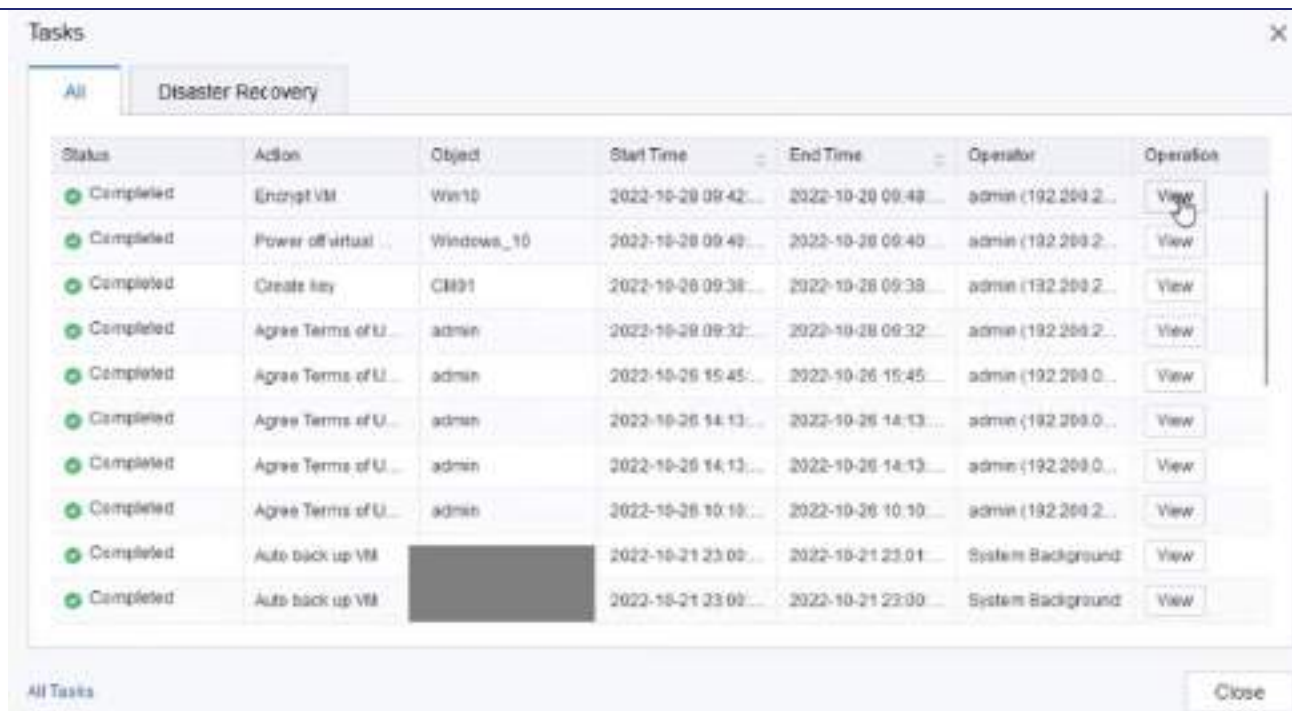
5. Enter the Sangfor Cloud Platform admin password to proceed with the encryption. Click **OK**.



6. Navigate to **Tasks** list to view the encryption progress.



7. Virtual machine successful encrypted result from Sangfor Cloud Platform.



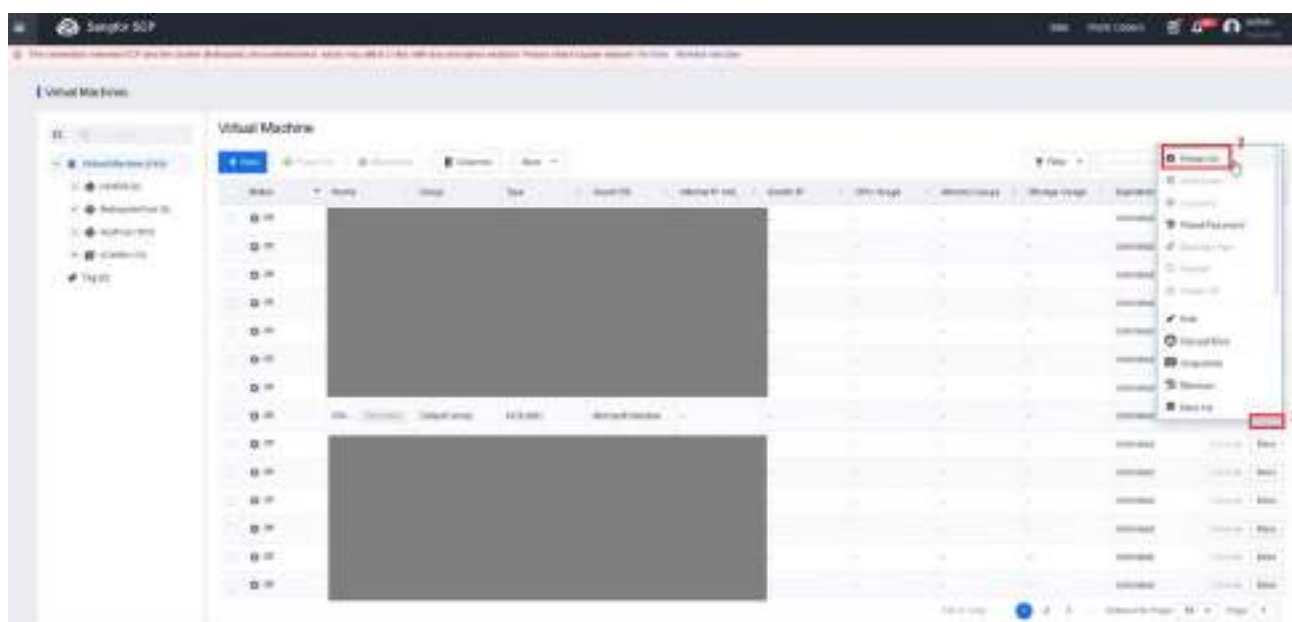
Verifying Integration

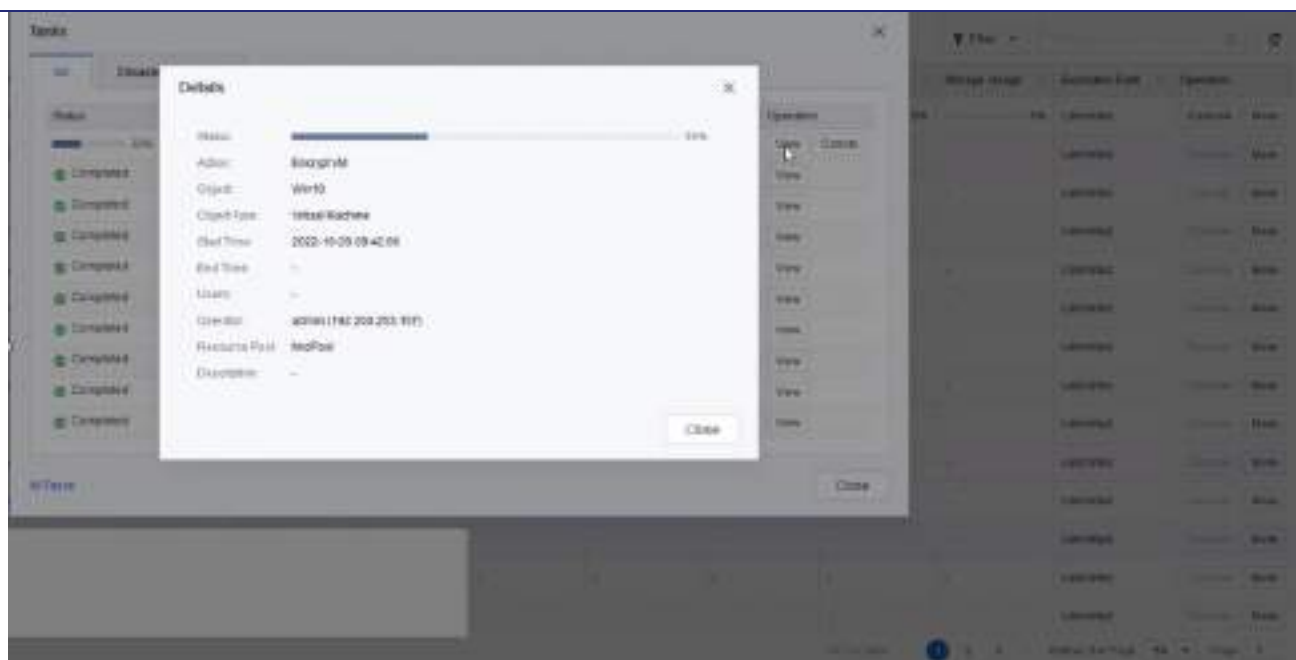
Perform the following steps to ensure the successful integration:

Power On the Encrypted Virtual Machine

After performing the integration steps, **Power ON** the Encrypted Virtual Machine.

1. You can view successful power on result from Sangfor Cloud Platform as shown below:





- Similarly, You can view successful power on result from CipherTrust Manager. Successful power on result from CipherTrust Manager indicates that the integration is successful.

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the Autoregistration option is selected in Admin Settings > System > Interfaces > kmip . If it is not selected, turn on the Autoregistration option. Refer to the steps mentioned in Enable KMIP Client Registration .
User corresponding to username location in certificate (example: OU/UID) has not been created	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"</i>	Check whether the user corresponding to UID/OU of the node certificates has been created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the OU/UID field of the node certificates. After creating the user, add this user to the Key Admins . Refer to steps mentioned in Create a New User .
User has not been added to Key Admin group	The following error message is displayed in CipherTrust Records:	Check whether the user corresponding to UID/OU of the

Issue	Error Message	Remediation
	<i>"errorMessage": "authorization denied: verdict was deny: CreateKey"</i>	node certificates has been created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The Username location in Certificate has been set incorrectly	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"</i>	Check whether the Username Location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the Username Location in the Certificate.



CipherTrust Integrations

4 February 2026

Contents

SANtricity 3

 Integration of SANtricity 4

 Verifying Your Integration 16

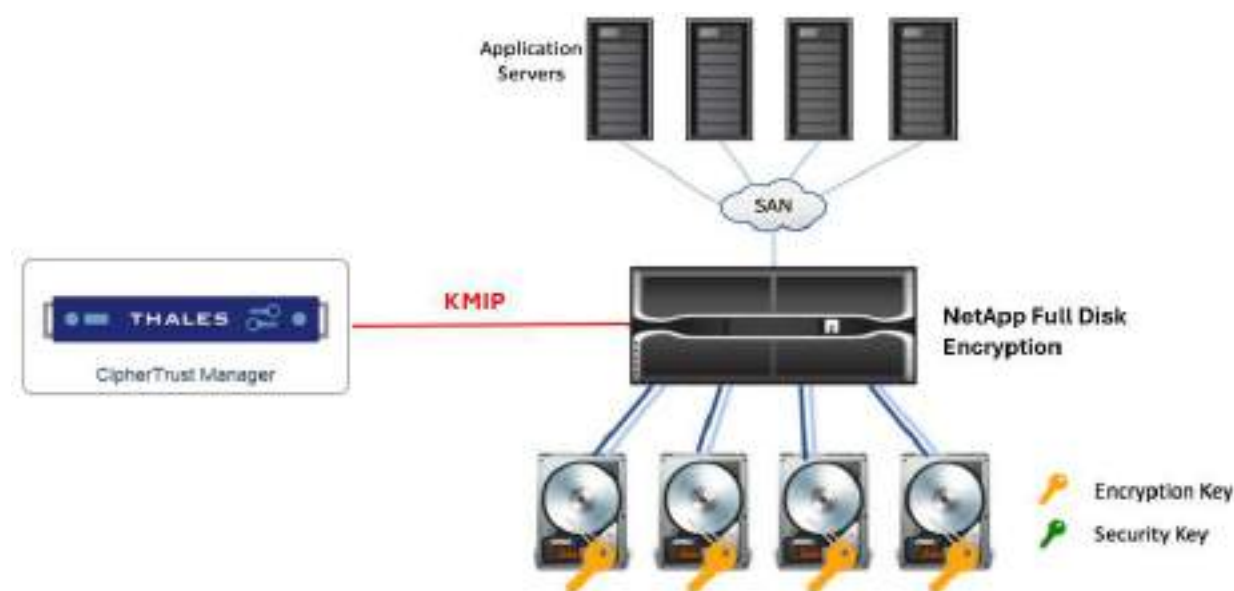
 Appendix 18

SANtricity

SANtricity optimizes the performance and resource use of E-Series and EF-Series storage systems, ensuring reliable, high-speed data handling for demanding tasks like analytics, video surveillance, and backup operations.

Here are the key features:

- **Boost performance:** Boost performance for applications that need fast response times, high data access speeds, or large data transfers.
- **Maximize uptime:** Experience top-tier reliability with automated tools, online configuration, advanced Dynamic Disk Pools technology, and other smart features.
- **Gain flexibility:** Make storage management easier while gaining the flexibility to fine-tune all E-Series storage systems to meet your specific needs.
- **Manage anytime, from anywhere:** Manage your NetApp E-Series system anytime, from anywhere. The built-in, web-based interface simplifies and streamlines your management tasks.



Supported Product Versions

Note

This integration has been validated in the field by a partner or in a customer environment with the following software versions. It is recommended to test the integration in a non-production environment with desired software versions before deploying it to production. Thales will provide best-effort support.

CipherTrust Manager

-
- [CipherTrust Manager 2.16](#)

SANtricity

- [SANtricity 11.8](#)

Integration of SANtricity

This section lists the steps to integrate SANtricity with CipherTrust Manager.

- [Prerequisites](#)
- [Configuration on CipherTrust Manager](#)
- [Configuration on SANtricity](#)

Prerequisites

This section provides the prerequisites for integration of SANtricity with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager Documentation](#).
- SANtricity communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the SANtricity system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the SANtricity cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.
- Ensure that SANtricity has proper license for data at Rest Encryption. To check the license information in SANtricity, navigate to **Settings > Software and Licenses > License Information**.
- Ensure that the required licenses are activated on CipherTrust Manager. For more details, refer to the [CipherTrust Manager documentation](#).

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a User](#)
2. [Assigning User to a Group](#)
3. [Creating or Adding a CA \(optional\)](#)

-
4. Registering a KMIP Client
 5. Configuring the KMIP Interface

Creating a User

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example, **Key Admins** or **Key Users**.
6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the knip interface.
 - c. Click **View/Edit**.
 - d. On the **Interface Detail** window, perform the following:

-
- Select the Interface Mode as **TLS**, **verify client cert**, **user name taken from client cert**, **auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
 2. Go to **Products > KMIP**.
 3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the **Certificate Details** section.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.

For domain, the format to enter the Common Name field of the cert is always:
`domainName | | domainUser`
 - f. Click **Save**.
 4. Create a Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
-

-
- h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.
5. If you are using External CA then you can select the external CA which you created earlier. Refer to **External CA** under [Creating or Adding a CA](#).
 6. Go to Registered Clients and click **Add Client**.
 - a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- b. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn**

off auto generation from Local CA.

6. Depending upon the CA that you created [above](#), select an option from the following::
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs** provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Configuration on SANtricity

1. [Create CSR on SANtricity](#)
2. [Upload the Signed Certificate to SANtricity](#)

Create CSR on SANtricity

1. On SANtricity System Manager, navigate to **Settings > Certificates > Key Management**. Click **Complete CSR. Complete & Download Client Certificate Signing Request** window appears.



2. Enter the required details and click **Download**. A CSR will be downloaded to your local system.

Complete & Download Client Certificate Signing Request ✕

Complete and download a client certificate signing request (CSR) to obtain proper client authentication to access the key management server...

Common name ?

Organization ?

Organizational unit (optional) ?

City/Locality

State/Region (optional) ?

Country ISO code ?

 Generating a new CSR will create a new key pair. This will overwrite any existing key pair, which may cause a loss of access to the key management server.

DownloadCancel

Upload the Signed Certificate to SANtricity

Before uploading the This certificate needs to be signed by the CA and the recieved signed certificate can be uploaded through the following steps:

1. Navigate to **Key Management** and click **Create External Key**.
2. On the **Create External Security Key** window, enter the CipherTrust Manager **IP address** and **Port**.

Create External Security Key

1 Connect to Key Server

2 Create/Backup Key

Connect to the following key management server...

What do I need to know before creating a security key?

Key management server address ?

Key management port number ?

+ Add key server

Select client certificate

Browse...

Select private key file (Optional)

Browse...

Select key server certificate (server, intermediate CA or root CA)

Browse...

Close

Next >

3. Select the SANtricity Certificate and CipherTrust CA certificate. Click **Next**.
4. Enter the **Passphrase**, **Re-enter passphrase** and click **Finish**.

The screenshot shows a window titled "Create External Security Key" with a close button (X) in the top right corner. The window has two steps: "1 Consent to Key Server" and "2 Create Backup Key", with the second step being the active one. Below the steps, the text reads "Create a security key and a backup key (optional) ...". There is a checkbox labeled "Create a backup key" which is checked. Below this, a note states: "Important: When you create a security key, a copy of the key will be saved to your local host. For security purposes, a pass phrase must be provided to encrypt the backup key." This is followed by the instruction "Define a pass phrase" with a help icon. There are two text input fields for the pass phrase, both containing masked characters (dots). At the bottom of the window, there are three buttons: "Back" (disabled), "Close" (disabled), and "Finish" (active).

Create External Security Key

1 Consent to Key Server 2 Create Backup Key

Create a security key and a backup key (optional) ...

☒ Create a backup key

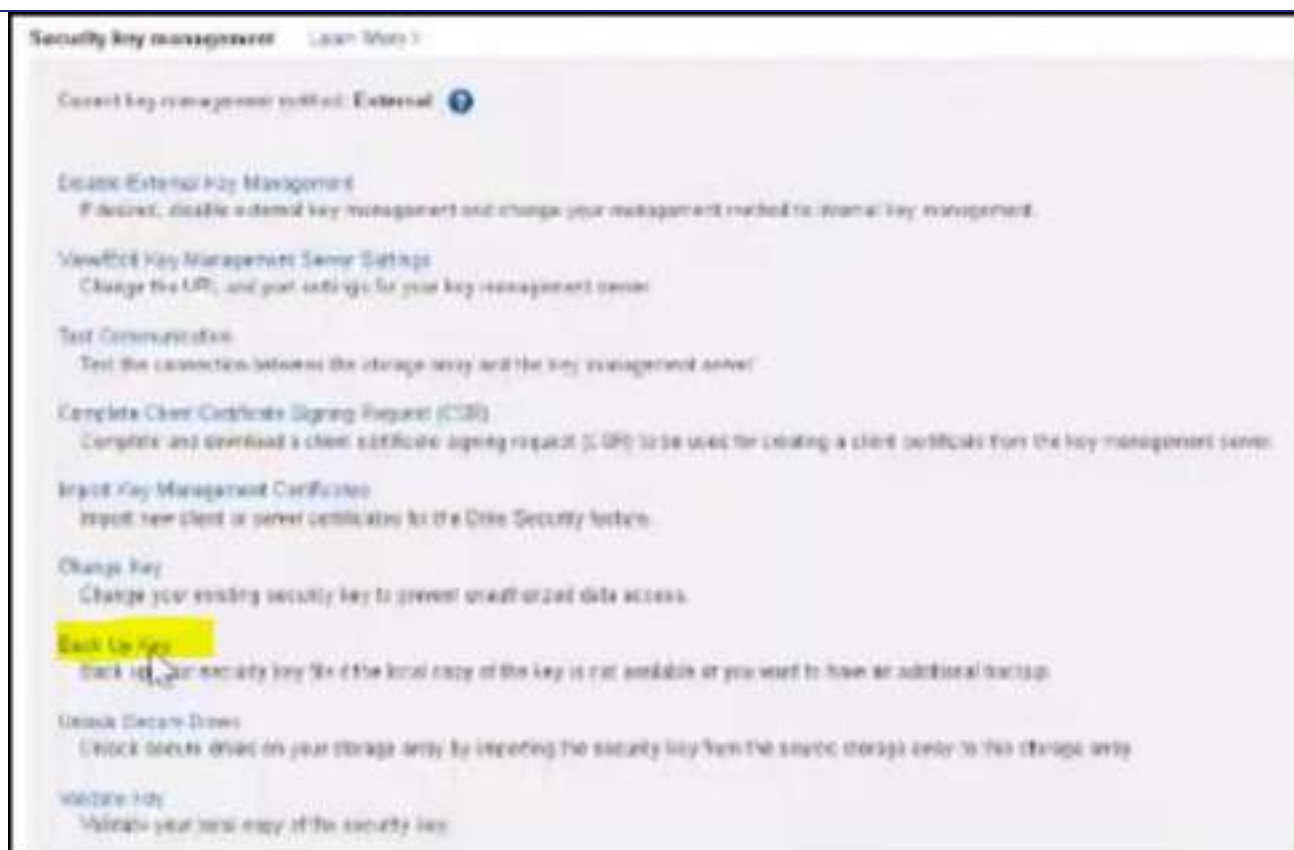
Important: When you create a security key, a copy of the key will be saved to your local host. For security purposes, a pass phrase must be provided to encrypt the backup key.

Define a pass phrase ?

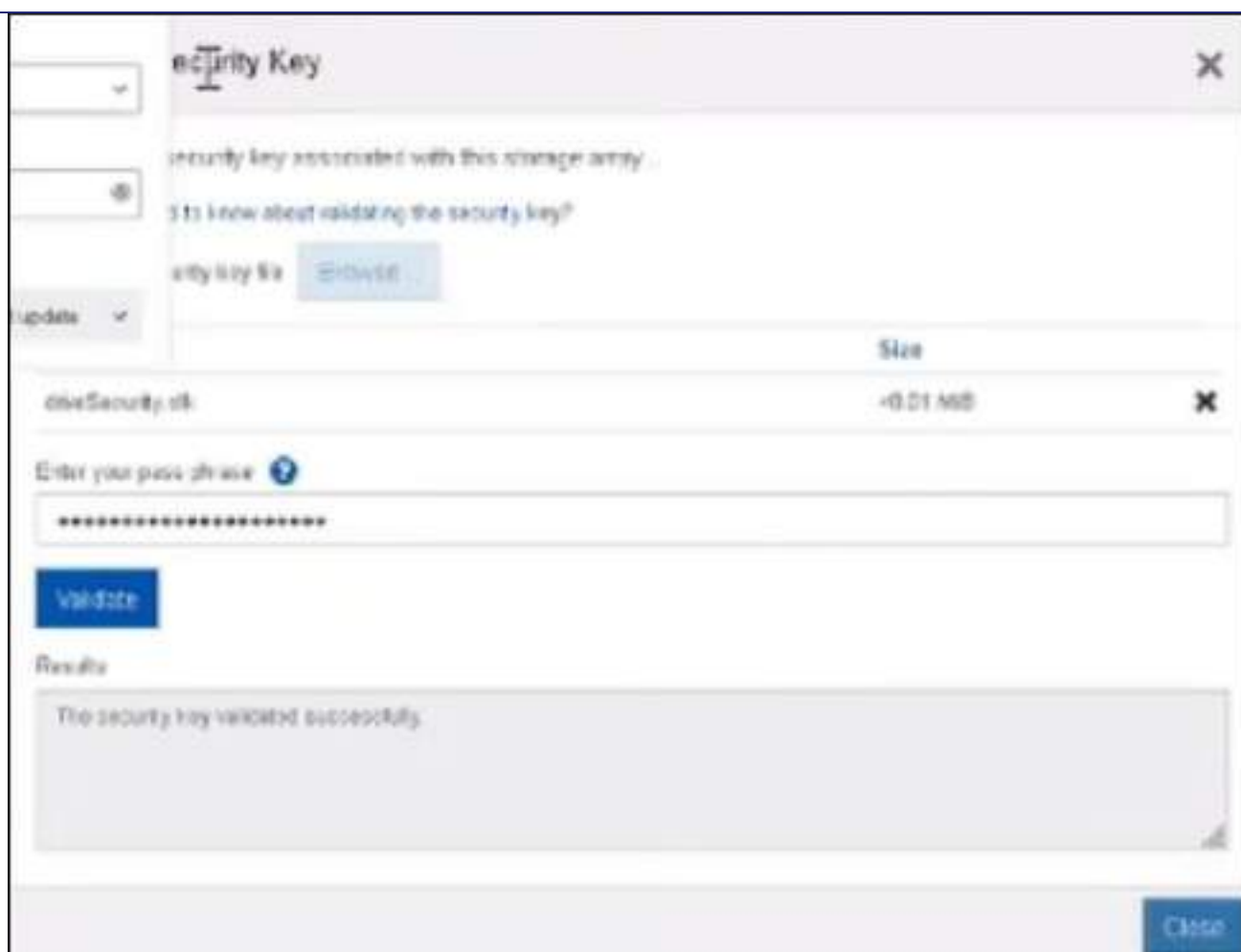
Re-enter the pass phrase

Back Close Finish

5. Click **Back Up Key**. **Back Up Security Key** window appears.



6. Enter the passphrase that you created above and click **Backup**. A backup file will be downloaded to your local system.
7. Click **Validate the backup key**. A pop-up window will appear.
8. Select the **Download backup key**, enter the **passphrase**, and click **Validate**.



9. A security key validated successfully will appear.
10. Under Security Key management, now External should be reflected near certificate key management.



11. Navigate to **CipherTrust Manager > Keys** and verify the keys created on CipherTrust Manager.

Verifying Your Integration

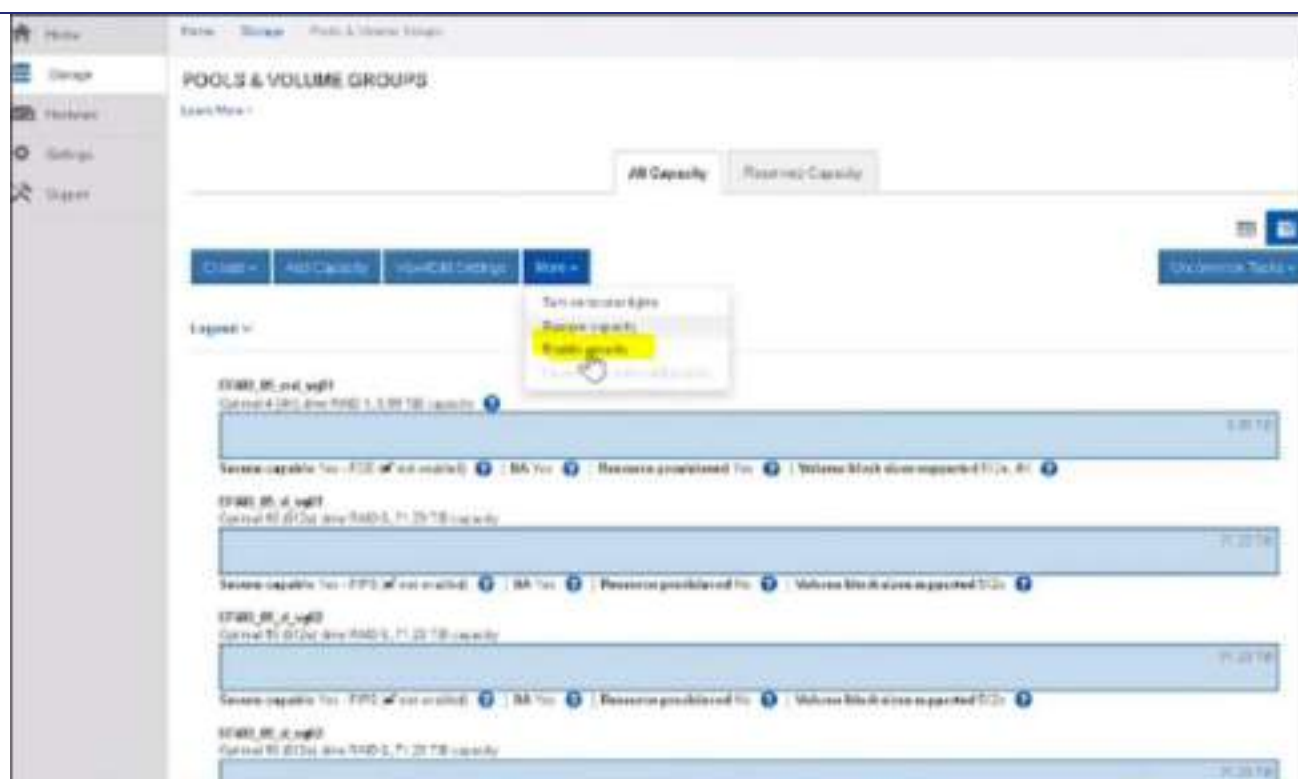
Perform the following steps to ensure the integration is successful:

Encrypt the Volume

1. On SANtricity System Manager, navigate to **Storage > Pools & Volume Group**.

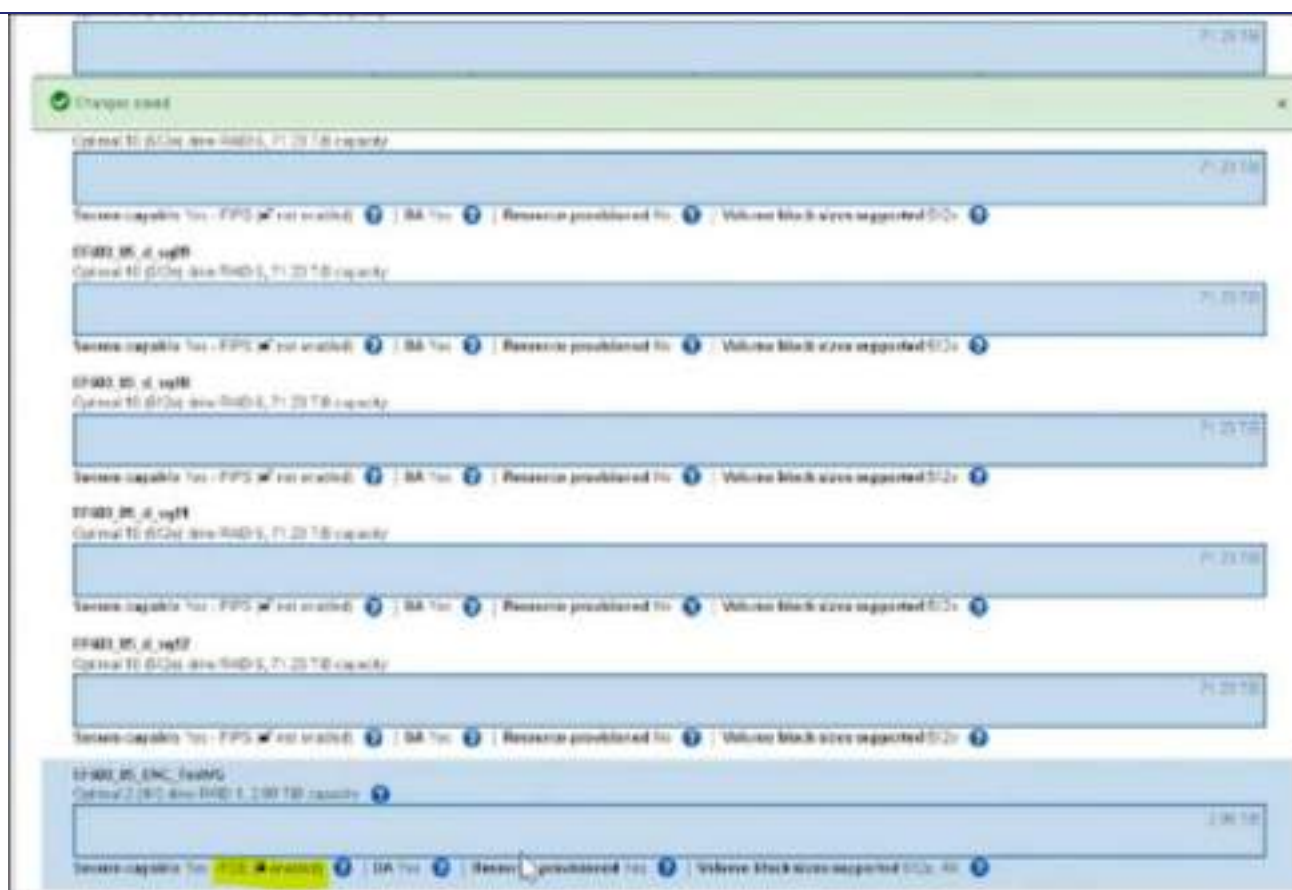


2. Select the volume that needs to be encrypted.
3. Click **More** and select **Enable Security** option.



4. Type **ENABLE** and click **Enable**. The volume gets encrypted and padlock icon appears against the specific volume.





5. Verify the **Records** on CipherTrust Manager where you can locate the **Export Key**.

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the Autoregistration option is selected in Admin Settings > System > Interfaces > kmip . If it is not selected, turn on the Autoregistration option. Refer to the steps mentioned in Enable KMIP Client Registration .
User corresponding to username location in certificate (example: OU/UID) has not been created	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found:"</i> <i>"errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"</i>	Check whether the user corresponding to UID/OU of the node certificates has been created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the OU/UID field of the node certificates. After creating the user,

Issue	Error Message	Remediation
		add this user to the Key Admins . Refer to steps mentioned in Create a New User .
User has not been added to Key Admin group	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "authorization denied: verdict was deny: CreateKey"</i>	Check whether the user corresponding to UID/OU of the node certificates has been created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The Username location in Certificate has been set incorrectly	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"</i>	Check whether the Username Location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the Username Location in the Certificate.



CipherTrust Integrations

4 February 2026

Contents

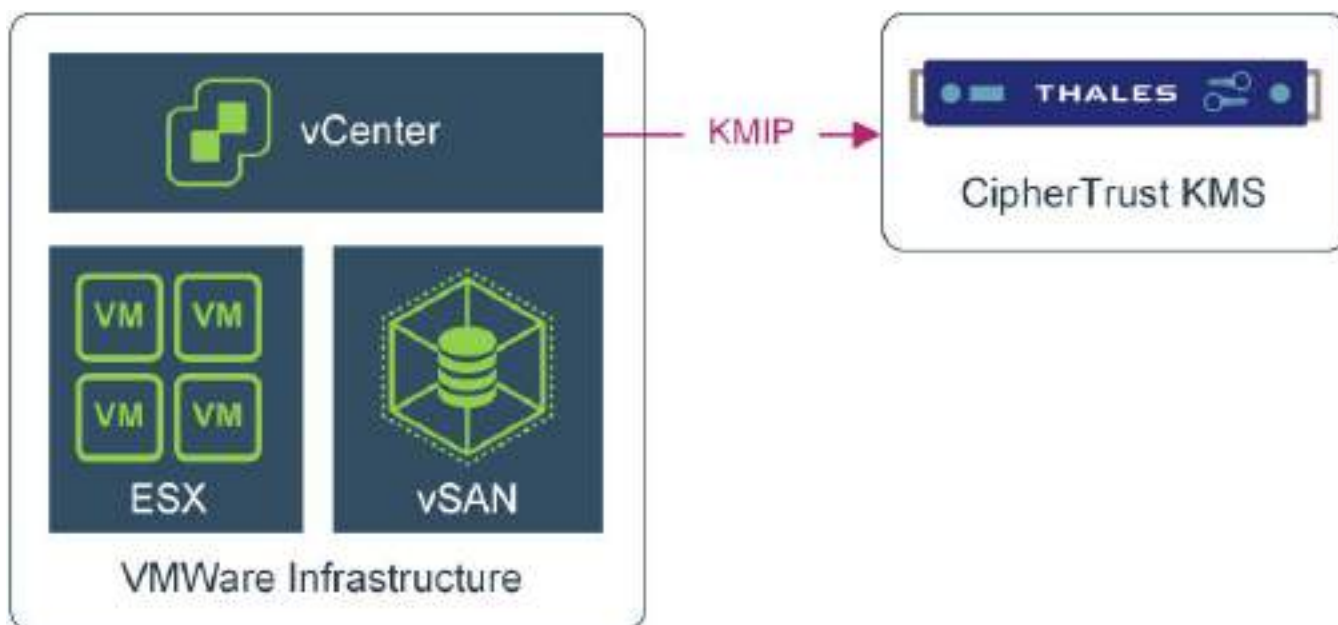
- VMware 3
 - Integration with CipherTrust Manager 4
 - Licensing 30
 - Appendix..... 30

VMware

Note

This integration is applicable to **vSAN**, **vSphere Trust Authority**, and **vmCrypt**.

vSAN is a VMware storage virtualization software that offers a mechanism to encrypt the virtual machine in place and in transit. The CipherTrust Manager is used to provide a key to the virtual machine and vSAN to enable the encryption.



Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

This integration is validated on the following software versions:

CipherTrust Manager

- CipherTrust Manager 1.8 and higher

VMware

-
- vSphere 6.5 (u1, u2)
 - vSphere 6.7 (u1, u2, u3)
 - vSphere 7.0 (u1)
 - vSphere 8.0 (u1, u2, u3)

For version compatability, you can also refer [VMware Compatibility Guide](#).

Key Management Interoperability Protocol (KMIP)

- Upto KMIP 1.4

Integration with CipherTrust Manager

This section lists the steps to integrate VMware with CipherTrust Manager.

- [Prerequisites](#)
- [Configuration on CipherTrust Manager](#)
- [Configuration on VMware](#)

Prerequisites

This section provides the prerequisites for integration of VMware with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager Documentation](#).
- VMware communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager.
- IP address of the CipherTrust Manager and port of the KMIP interface must be accessible from the VMware system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the VMware cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)

-
4. Creating or Adding a CA (optional)
 5. Registering a KMIP Client
 6. Configuring the KMIP Interface

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

1. Navigate to **Admin Settings > Domains**.
2. Click **Add Domain**. The **Add Domain** page appears.
3. Specify the following information:
 - **Name** - Enter the domain name.
 - **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.
 - **Parent CA** - Select parent CA as root CA.
 - **Allow Subdomain User Management** - Select this check box if you want to enable the sub-domain user management through this domain.
4. Click **Save**.
5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**

-
- Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.

6. Click **Add User**. The newly created user will now appear on the **Users** page.

Sub-Domain

Note

To create a user in sub-domain, you must enable **Allow Subdomain User management**.

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the sub-domain.
2. Open the **Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:
 - Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter **Password**. Re-enter the password in the **Password Match** field.
6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example,

Key Admins or Key Users.

6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
-

-
- a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **View/Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.

- e. Select the CA type as Local/External depending on the CA type that you created [above](#).
- f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
- g. Select the Client Profile from the dropdown which you have created in the above step.
- h. Click **Create Token**.
- i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.

- Specify client name and paste the **Registration Token** that you generated in the above step.

Note

If you are using external CA then you can either upload the client certificate as a file using **Upload Certificate** or paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub-Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.

-
- b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
 4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created [above](#) and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the knip interface.
 - c. Click **Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
-

-
- Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the **Certificate Details** section.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.

For domain, the format to enter the Common Name field of the cert is always:

`domainName | | domainUser`

- f. Click **Save**.
4. Create a Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.
 5. If you are using External CA then you can select the external CA which you created earlier. Refer to **External CA** under [Creating or Adding a CA](#).
 6. Go to Registered Clients and click **Add Client**.
-

- a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- a. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

6. Depending upon the CA that you created [above](#), select an option from the following::
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs**, provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating

or uploading CA, refer to [Creating or Adding a CA](#).

Sub-Domain

1. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

External CA

Note

If you are using an External CA in the Sub-Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

- i. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options** > **Upload New Certificate** > **Ok**.
 - j. Select the **Certificate Chain** option and click **Build Certificate Chain**.
 - k. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
 - l. Select certificate Format as **PEM**.
 - m. Click on **Upload Certificate**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
 3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

In case of External CA, **Local CA for Automatic Server Certificate Generation** should be set to **Turn off auto generation from Local CA**.

Further, You need to perform the following configuration on CipherTrust Manager specific to VMware:

1. [Creating a Client Certificate](#)
2. [Configuring NTP Server \(Optional\)](#)

Creating a Client Certificate

This section is applicable to KMIP clients registered using Auto Registration.

1. Log on to the CipherTrust Manager.
2. Navigate to the **CA > Local CA** and click on the name of the CA certificate you need to issue the certificate from.

3. Click on **Issue Certificate**.

4. Enter the **Display Name**, followed by **Common Name**.

The common Name of the certificate should be the same as the name of the user you created on CipherTrust Manager. Refer to [Creating a User](#) section.

For more information on the format of common name of the certificate, refer to [LDAP and Multi-Domain Client Usernames in KMIP Certificates](#).

5. Select the **Algorithm** and **Size**, and click **Issue Certificate**.

6. Save the **Private Key** and the **CSR**.

7. Select the Certificate Purpose as client, specify the validity of the certificate in days, and click **Issue Certificate**.

8. Download a copy of this certificate by clicking the ellipsis next to the certificate name.

Configuring NTP Server (Optional)

Based on your deployment strategy, you may need to configure an NTP (Network Time Protocol) server. Use either of the following commands to add an NTP server:

Command 1:

```
ksctl ntp servers add --host time.nist.gov
```

Command 2:

```
ksctl ntp servers add --host ntp-b.nist.gov --key secret
```

Configuration on VMware

Add KMS to the vCenter Server

To add KMS, perform the following steps on the vCenter UI.

1. Log in to the **vCenter Server**.
2. Browse the inventory list and select the **vCenter Server** instance.
3. Click **Configure** and under **Security**, click **Key Providers**.
4. Click **Add Standard Key Provider**, enter key provider information, and click **Add Key Provider**. You can click **Add KMS** to add more Key Management Servers.

Add Standard Key Provider



Name key-provider

KMS	Address	Port	
<u>CipherTrust-Node1</u>	<u>192.168.100.145</u>	<u>5696</u>	
<u>CipherTrust-Node2</u>	<u>192.168.100.146</u>	<u>5696</u>	

ADD KMS

> Proxy configuration (optional)

> Password protection (optional)

CANCEL **ADD KEY PROVIDER**

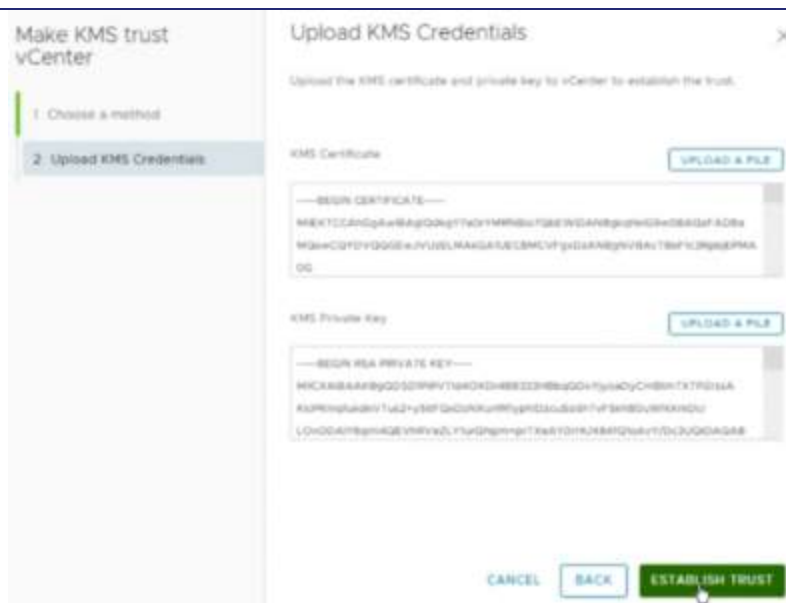
5. Click **Trust**. vCenter Server adds the key provider and displays the status as Connected.

Uploading Client Certificate and Private Key onto vSphere

Perform the following steps to upload Client Certificate and Private Key:

1. Log in to the vSphere Web Client, and select a vCenter Server system.
2. Click **Configure** and select **Key Management Servers**.
3. Select the KMS instance with which you want to establish a trusted connection.
4. Select **Upload a File** and upload copies of the client certificate and private key and click Establish Trust.

Alternatively, you can paste the certificate that you received from the KMS vendor into the top text box or click **Upload File** to upload the certificate file. Paste the key file into the bottom text box or click **Upload File** to upload the key file.



5. Click **OK**.

You can enable the following encryptions:

FCD

First Class Disk (FCD), or Improved Virtual Disk, enables independent lifecycle management of virtual disks separate from VMs. FCDs can be created through FCD APIs or by promoting existing disks.

FCDs are managed via the VSLM API. Newer vSphere versions allow snapshot-based backups of detached FCDs using UUID and datastore ID, unlike earlier versions that required attachment to a dummy VM. Regular I/O still needs attachment. FCDs support encryption via external KMS for enhanced security.

Here are the steps to Configure FCDs:

Steps to Create a New VM Storage Policy with Encryption

1. Go to **Menu > Policies and Profiles > VM Storage Policies**.
2. Click **Create** to create a new storage policy.
3. Enter a name for the policy (for example, **FCD-Encryption-Policy**).
4. Under **Rule Sets**, select **Enable encryption**.
5. Complete the wizard and click **Finish**.

Steps to Create an Encrypted First-Class Disk (FCD)

1. Connect to the vCenter Server using PowerCLI.

```
Connect-VIServer -Server 10.xx.xx.xx -User appuser -Password xxxxxxxx
```

2. Fetch the storage policy and the corresponding datastore.

```
$storagePolicy = Get-SpbmStoragePolicy -Name "TCG-Policy-Thales"
$datastore = Get-Datastore -Name "datastore1"
```

3. Create a new encrypted disk and attach it to your VM.

```
New-HardDisk -CapacityGB 10 -StoragePolicy $storagePolicy -Datastore $datastore  
-Confirm:$false -VM Windows-Test_Machine -KeyProvider $keyprovider
```

VMcrypt

1. Connect to vCenter Server by using the vSphere Client.
2. Right-click the virtual machine that you want to encrypt and select **VM Policies > Edit VM Storage Policies**.
3. Select the storage policy.
4. To encrypt the VM and its hard disks, select an encryption storage policy and click **OK**.
5. To encrypt the VM but not the virtual disks, toggle on **Configure per disk**, select the encryption storage policy for VM Home and other storage policies for the virtual disks, and click **OK**.
6. If you prefer, you can encrypt the virtual machine, or both virtual machine and disks, from the **Edit Settings** menu in the vSphere Client.
7. Right-click the virtual machine and select **Edit Settings**.
8. Select the VM Options tab, and open **Encryption**. Choose an encryption policy. If you deselect all disks, only the VM home is encrypted.
9. Click **OK**.

VSAN

To enable vSAN encryption:

1. Navigate to the KMS cluster created in vCenter.
2. Right-click the cluster and select **Settings**. The Configure tab is displayed.
3. Expand the vSAN section and click **General**.
4. Click **Edit**. Edit vSan Settings window is displayed.
5. Enable Encryption and select the previously created KMS cluster.
6. Click **OK**.

VTA

VMware has added a new feature named vSphere Trust Authority and CipherTrust Manager supports this feature. To configure vSphere Trust Authority, refer to the [VMware documentation](#).

To connect Key Provider Service to KMS, you need to configure the trust setup.

Example:

```
Set-TrustAuthorityKeyProviderClientCertificate -KeyProvider $kp8 -CertificateFile
Path <path/to/certfile.pem> -PrivateKeyFilePath <path/to/privatekey.pem>
```

Configure and Enable Trust Authority State

vSphere Trust Authority is not enabled by default. You must configure your environment for vSphere Trust Authority before you can start using it.

1. Making a vCenter Server cluster into a vSphere Trust Authority Cluster (also called enabling the Trust Authority State) starts the required Trust Authority services on the ESXi hosts in the cluster.
2. Connect as the Trust Authority Administrator user to the vCenter Server of the Trust Authority Cluster.

```
Connect-VIServer -server vsphere-sunnyvale1.gemalto.com -User trustedadmin@vsp
here.local -Password xxxxxxxx
```

3. If you see the following error.

```
Connect-VIServer : 2/14/2022 1:50:42 PM Connect-VIServer Error: Invalid serve
r certificate. Use
Set-PowerCLIConfiguration to set the value for the InvalidCertificateAction op
tion to Prompt if you'd like to connect once or to add a permanent exception f
or this server.
Additional Information: Could not establish trust relationship for the SSL/TLS
secure channel with authority 'xx.xxx.xxx.11'.
```

Run the following command and try again.

```
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore
```

4. Check the current state of the cluster. The output shows either **Disabled** or **Enabled** in the **State** column for each cluster found. **Disabled** means that the Trust Authority services are not running.
5. Enable the Trust Authority Cluster.

```
Set-TrustAuthorityCluster -TrustAuthorityCluster 'Infrastructure' -State Enabl
ed
```

Two services start on the ESXi hosts in the Trust Authority Cluster.

- The Attestation Service
- The Key Provider Service

Collect Information about ESXi Hosts and vCenter Server to be Trusted

To establish trust, the vSphere Trust Authority Cluster requires information about the Trusted Cluster's ESXi hosts and vCenter Server. This information should be exported as files for importing into the Trust Authority Cluster.

Export the ESXi Host Description of Software

1. Disconnect from all connections.

```
Disconnect-VIServer -server * -Confirm:$false
```

2. Connect as the root user to one of the ESXi hosts in the Trusted Cluster.

```
Connect-VIServer -server 10.xxx.xx.xx -User root -Password xxxxxxxxx
Connect-VIServer -server 10.xxx.xx.xx -User root -Password xxxxxxxxx
```

3. Confirm the ESXi host:

```
Get-VMHost
```

4. Assign Get-VMHost to a variable.

```
$vmhost = Get-VMHost
```

5. Export the ESXi host description of software (the ESXi image). By default, this command exports the information to the `image.tgz` file. The `image.tgz` file in the example is named `image-esxi-201.tgz` to indicate what ESXi host this is from. Ensure that the destination directory exists before running this command:

```
Export-VMHostImageDb -VMHost $vmhost -FilePath C:\vta\image-esxi-201.tgz
```

Collect the TPM Endorsement Key

1. Disconnect any current connection.

```
Disconnect-VIServer -server * -Confirm:$false
```

2. Connect as the root user to one of the ESXi hosts in the Trusted Cluster.
-

```
Connect-VIServer -server 10.xxx.xx.xx -User root -Password xxxxxxxx
```

3. Confirm the ESXi host.

```
Get-VMHost
```

4. Assign Get-VMHost to a variable.

```
$vmhost = Get-VMHost
```

5. Export the EK certificate of the ESXi host.

```
Export-Tpm2EndorsementKey -VMHost $vmhost -FilePath -FilePath C:\vta\tpm2ek-201.json
```

The file is created. Repeat the process for the other ESXi hosts you want to trust. Make sure the filename is different for each ESXi host.

Collect Information about the Trusted Cluster's vCenter Server

1. Export the Trusted Cluster's vCenter Server principal information. To do so, first disconnect from the ESXi host.

```
Disconnect-VIServer -server * -Confirm:$false
```

2. Connect to the vCenter Server of the Trusted Cluster using the Trust Authority Administrator user.

```
Connect-VIServer -server 10.xxx.xx.xx -User trustedAdminTC -Password xxxxxxxx
```

3. Export the Trusted Cluster's vCenter Server principal information.

```
Export-TrustedPrincipal -FilePath C:\vta\trustedcluserprincipal-12.json
```

Import the Trusted Host Information to the Trust Authority Cluster

To import a TPM EK certificate into the Trust Authority Cluster, you must change the Trust Authority Cluster's default attestation type to accept EK certificates. The default attestation type accepts TPM Certificate Authority (CA) certificates.

1. Disconnect from the ESXi host.

```
Disconnect-VIServer -server * -Confirm:$false
```


2. Connect as the Trust Authority Administrator user to the vCenter Server of the Trust Authority Cluster.

```
Connect-VIServer -server 10.xxx.xx.xx -User trustedadmin@vsphere.local -Password xxxxxxxx
```

3. Check the current state of the cluster.

```
Get-TrustAuthorityCluster
```

4. Assign the Get-TrustAuthorityCluster cluster information to a variable.

```
$vTA = Get-TrustAuthorityCluster 'infrastructure'
```

5. Assign the Get-TrustAuthorityTpm2AttestationSettings information to a variable.

```
$tpm2Settings = Get-TrustAuthorityTpm2AttestationSettings -TrustAuthorityCluster $vTA
```

6. Run the Set-TrustAuthorityTpm2AttestationSettings cmdlet, specifying RequireEndorsementKey. At the confirmation prompt, press Enter.

```
Set-TrustAuthorityTpm2AttestationSettings -Tpm2AttestationSettings $tpm2Settings -RequireEndorsementKey
```

Output

```
Confirmation
Configure the Tpm2AttestationSettings 'TrustAuthorityTpm2AttestationSettings-domain-c2111'
following parameters:
RequireCertificateValidation: False
RequireEndorsementKey: True
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
```

Name	RequireEndorsementKey	RequireCertificateValidation	Health
TrustAuthorityTpm2AttestationSettings...	True	False	Ok

7. Connect to the VI server.

```
Connect-VIServer -server 10.xxx.xx.xx -User trustedadmin@vsphere.local -Password xxxxxxxx
```

Output

```

Name          Port  User
----          -
10.xxx.xx.xx  443   VSPHERE.LOCAL\trustedadmin

```

8. Store the details of **TrustAuthorityCluster** in a variable **vTA**.

```

$vTA = Get-TrustAuthorityCluster 'infrastructure'
$vTA

```

Output

```

Name          State  Id
----          -
Infrastructure Enabled TrustAuthorityCluster-domain-c40

```

9. Fetch the endorsement keys for the trust authority cluster.

```

New-TrustAuthorityTpm2EndorsementKey -TrustAuthorityCluster $vTA -FilePath
C:\vta\tpm2ek-25.json

```

10. Import the ESXi host base image information.

```

New-TrustAuthorityTpm2EndorsementKey -TrustAuthorityCluster $vTA -FilePath
C:\vta\tpm2ek-27.json

```

Output

```

TrustAuthorityClusterId          Name
-----
TrustAuthorityCluster-domain-c40 34323550-3939-4d33-3144-323631314734

```

Health

Ok

11. Import the ESXi host base image information.

```

New-TrustAuthorityVMHostBaseImage -TrustAuthorityCluster $vTA -FilePath C:\vta\image-esxi-25.tgz

```

Output

```

TrustAuthorityClusterId          VMHostVersion          Health

```

```
-----
TrustAuthorityCluster-domain-c40
```

```
-----
ESXi 8.0.3-0.70.24674464
```

```
-----
Ok
```

The Trust Authority Cluster knows which ESXi hosts it can remotely attest, and so, which hosts it can trust.

Create the Key Provider on the Trust Authority Cluster

1. Disconnect from all connections.

```
Disconnect-VIServer -server * -Confirm:$false
```

2. Connect to the vCenter Server of the Trust Authority Cluster.

```
Connect-VIServer -server vsphere-sunnyvale1.gemalto.com -User trustedadmin@vsp
here.local -Password Asdf@1234
```

3. Assign Get-TrustAuthorityCluster information to a variable called \$VTA.

```
$VTA = Get-TrustAuthorityCluster 'Infrastructure'
```

4. Create the trusted key provider using the New-TrustAuthorityKeyProvider cmdlet.

```
New-TrustAuthorityKeyProvider -TrustAuthorityCluster $VTA -PrimaryKeyId db781d
cbdf42481c910117e19631efc0e9d4f39055074eb1b2eec842fd13a23c -Name KeyControl1
-KmipServerAddress 10.xxx.xx.xx -KmipServerPort xxxx
```

5. Use the ID of the keys created earlier and assign a name. For example, KeyControl1.
6. Specify the IP address and the port of the CipherTrust Manager.
7. Each logical key provider must have a unique name across all vCenter Server systems.
8. To add multiple key servers to the key provider, use the Add-TrustAuthorityKeyProviderServer cmdlet.
9. Establish the trusted connection so that the KeyControl server trusts the trusted key provider. Obtain the trusted key providers in the given Trust Authority Cluster.

```
$kp = Get-TrustAuthorityKeyProvider -TrustAuthorityCluster $VTA | Select-Objec
t -First 1
```

10. Upload the certificate and private key using the Set-TrustAuthorityKeyProviderClientCertificate command.

```
Set-TrustAuthorityKeyProviderClientCertificate -KeyProvider $kp -CertificateFilePath C:\Users\Administrator\Downloads\v-cert.pem -PrivateKeyFilePath C:\Users\Administrator\Downloads\v-key.pem
```

As a result, the trusted key provider has established trust with the key server. Finish the trust setup by uploading a key server certificate so that the trusted key provider trusts the key server.

11. Obtain the trusted key providers in the given Trust Authority Cluster.

```
$kp = Get-TrustAuthorityKeyProvider -TrustAuthorityCluster $VTA | Select-Object -First 1
```

12. Get the key server server certificate. Initially, the certificate is not trusted, so the Trusted state is False. If you have more than one key server configured, a list of certificates is returned. Verify and add each certificate using the following instructions. Before trusting the certificate, assign `Get-TrustAuthorityKeyProviderServerCertificate -KeyProviderServer $kp.KeyProviderServers` information to a variable.

```
$cert = Get-TrustAuthorityKeyProviderServerCertificate -KeyProviderServer $kp.KeyProviderServers
```

13. Run this command and verify the output.

```
$cert.Certificate.ToString()
```

Output

```
[Subject]
CN=kmip.ciphertrustmanager.local, O=Thales, L=Austin, S=TX, C=US
[Issuer]
CN=CipherTrust Root CA, O=Thales, L=Austin, S=TX, C=US
[Serial Number]
39986905AFCCD708983A4CE975BC72D9
[Not Before]
6/10/2025 3:37:25 PM
[Not After]
9/14/2027 3:37:25 PM
[Thumbprint]
CC638805FCFAF3E9E3CB1CB4B737E04078BF444D5
```

14. Add the KMIP server certificate to the trusted key provider.

```
Add-TrustAuthorityKeyProviderServerCertificate -ServerCertificate $cert
```

15. Verify the status of the key provider, by refreshing the key provider status.

```
$kp = Get-TrustAuthorityKeyProvider -TrustAuthorityCluster $VTA
```

16. Get the key provider status.

```
$kp.Status
```

Output

KeyProviderId	Health	HealthDetails	ServerStatus
domain-c40-KeyControl4	Ok	{}	{10.xxx.xx.xx}

Export the Trust Authority Cluster Information

After you have imported the vSphere Trust Authority Cluster information to the Trusted Cluster, the Trusted Hosts start the attestation process with the Trust Authority Cluster.

1. To import the Trust Authority Cluster information, Disconnect from all connections.

```
Disconnect-VIServer -server * -Confirm:$false
```

2. Ensure that you are connected as the Trust Authority Administrator to the vCenter Server of the Trusted Cluster.

```
Connect-VIServer -server vsphere-sunnyvale1.gemalto.com -User trustedadmin@vsphere.local -Password xxxxxxxxx
```

3. Export the Trust Authority Cluster's Attestation Service and Key Provider Information, this command exports the service information to the `clsettings.json` file.

```
$VTA = Get-TrustAuthorityCluster 'Infrastructure'
Export-TrustAuthorityServicesInfo -TrustAuthorityCluster $VTA -FilePath C:\vta\clsettings.json
```

Output

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	6/6/2025 9:42 PM	8352	clsettings.json

The file containing the Trust Authority Cluster information is created.

Import the Trust Authority Cluster Information to the Trusted Hosts

To import the Trust Authority Cluster information.

1. Disconnect from all connections:

```
Disconnect-VIServer -server * -Confirm:$false
```

2. Ensure that you are connected as the Trust Authority Administrator to the vCenter Server of the Trusted Cluster (Workload cluster).

```
Connect-VIServer -server TrustedCluster_VC_ip_address -User trust_admin_user  
-Password 'password'
```

3. Verify that the state of the Trusted Cluster is Disabled.

```
Get-TrustedCluster
```

4. Assign the Get-TrustedCluster information to a variable and verify its value.

```
$TC = Get-TrustedCluster -Name 'KMS-Cluster'
```

5. Import the Trust Authority Cluster information to the vCenter Server. Use the `clsettings.json` file previously exported in "Export the Trust Authority Cluster Information".

```
Import-TrustAuthorityServicesInfo -FilePath C:\vta\clsettings.json
```

6. Enable the Trusted Cluster.

```
Set-TrustedCluster -TrustedCluster $TC -State Enabled
```

Output

```
Confirmation  
Setting TrustedCluster 'KMS-Cluster' with new TrustedState 'Enabled'. Do you want to proceed?  
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
```

Name	State	Id
----	-----	--
KMS-Cluster	Enabled	TrustedCluster-domain-c9

7. Verify that the Attestation Service and the Key Provider Service are configured in the Trusted Cluster.

```
$TC = Get-TrustedCluster -Name 'KMS-Cluster'
$TC.AttestationServiceInfo
```

Output

ServiceAddress	ServicePort	ServiceGroup
-----	-----	-----
10.xxx.xx.xx	443	host-45:xx7f8085-9ca1-446c-...
10.xxx.xx.xx	443	host-159846:xx7f8085-9ca1-4...

The ESXi Trusted Hosts in the Trusted Cluster begin the attestation process with the Trust Authority Cluster.

Configure the Trusted Key Provider for Trusted Hosts

1. Disconnect from all connections.

```
Disconnect-VIServer -server * -Confirm:$false
```

2. Ensure that you are connected as the Trust Authority Administrator to the vCenter Server of the Trusted Cluster (Workload cluster).

```
Connect-VIServer -server TrustedCluster_VC_ip_address -User trust_admin_user
-Password 'password'
```

3. Obtain the trusted key provider.

```
Get-KeyProvider
```

You can use the `-Name` option to specify a single trusted key provider.

4. Assign the Get-KeyProvider trusted key provider information to a variable.

```
$workload_kp = Get-KeyProvider
```

5. If you have multiple trusted key providers, you can use Select-Object to select one of them: (Optional)

```
$workload_kp = Get-KeyProvider | Select-Object -Index 0
```

6. Register the trusted key provider.

```
Register-KeyProvider -KeyProvider $workload_kp
```

Output

Name	DefaultForSystem	ClientCertificateExpiryDate
----	-----	-----
KeyControl1	False	1/1/0001 12:00:00 AM

Licensing

This section provides the licensing requirements for KMIP client, vSAN, and vCenter.

Condition	License Requirement
Encryption at VM Level	x1 KMIP client license for vCenter
Encryption at vSAN Level	Minimum x2 KMIP client license. x1 for vCenter + x1 per vSAN node

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the Autoregistration option is selected in Admin Settings > System > Interfaces > kmip . If it is not selected, turn on the Autoregistration option. Refer to the steps mentioned in Enable KMIP Client Registration .
User corresponding to username location in certificate (example: OU/UID) has not been created	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"</i>	Check whether the user corresponding to UID/OU of the node certificates has been created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the OU/UID field of the node

Issue	Error Message	Remediation
		certificates. After creating the user, add this user to the Key Admins . Refer to steps mentioned in Create a New User .
User has not been added to Key Admin group	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "authorization denied: verdict was deny: CreateKey"</i>	Check whether the user corresponding to UID/OU of the node certificates has been created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The Username location in Certificate has been set incorrectly	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"</i>	Check whether the Username Location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the Username Location in the Certificate.



CipherTrust Manager

Version: v2.22 (latest)

4 February 2026

Contents

KMIP Reference 3

KMIP Reference

Note

For integration guides on the respective KMIP clients, refer to [CipherTrust Integrations](#).

The Key Management Interoperability Protocol (KMIP) is a communication protocol that facilitates key management and cryptographic operations on a key management server. This section provides additional user information and specifications applicable to the KMIP interface of CipherTrust Manager.

This section provides the following information:

- [Setting up a KMIP Client](#)
- [KMIP Licensing](#)
- [KMIP Client Registration](#)
- [Supported Objects, Operations, Attributes, and Key Formats](#)
- [Cryptographic Operations for KMIP](#)
- [Recommended Curve Mappings](#)
- [KMIP Interface Mode](#)

Setting up a KMIP Client

Here are the prerequisites for allowing KMIP clients to connect to CipherTrust Manager:

- The CipherTrust Manager server's KMIP port 5696 should be open from the client.
- The CipherTrust Manager server supports the following TLS versions:
 - TLS version 1.0
 - TLS version 1.1
 - TLS version 1.2 (default minimum)
 - TLS version 1.3

Caution

TLS 1.0 and 1.1 are deprecated and support will be discontinued in a future release.

- Configure the KMIP service to trust the external CAs. Refer to [Trusted CAs \(NAE, KMIP, and WEB\)](#) for details.

KMIP Parameter: `kmip_enable_hard_delete`

This optional parameter enables a hard delete of keys on a "KMIP Destroy" operation. This means that both meta-data and material are removed from CipherTrust Manager for the key being deleted. By default, only key material is removed and meta-data is preserved with the updated key state. This setting applies only to the KMIP interface. Should be set to '1' for enabling this feature or '0' for returning to the default behavior.

KMIP Licensing

The KMIP interface is active for a 90-day trial period after CipherTrust Manager is installed. After the trial period expires, a new purchased KMIP license must be installed to use the KMIP interface. Contact [Thales Customer Support](#) for assistance in obtaining a license.

To activate a license using the GUI

Activating a KMIP license requires a license string for the CipherTrust Manager with which the clients will be registered. This string is generated when a license is activated on the Sentinel EMS License Portal.

Refer to [Activating a Connector License](#) for details.

After the KMIP license is activated, its state becomes **Active** on the Features tab of the Licensing page of the CipherTrust Manager GUI. The license is displayed with the feature name **KMIP**.

KMIP Client Registration

CipherTrust Manager recognizes only registered KMIP clients.

For registering a KMIP client, a certificate is required. The certificate facilitates authentication and communication between CipherTrust Manager and the client.

You can register KMIP clients using following methods:

1. Using Auto Registration (using an existing certificate)

Auto registration is useful when the clients are having certificates from the Classic KeySecure, or from older versions of CipherTrust Manager, or certificates issued by External CAs.

If you are already having one of the above certificate, you can utilize it to register your KMIP client, or to upgrade to a newer version of CipherTrust Manager. For detailed steps, refer to the section [KMIP Client Registration for existing CipherTrust Manager and external CA\(s\)](#).

2. Using Manual Registration

Before proceeding to manual registration, make sure that a client profile is created on the CipherTrust Manager. A client profile contains a set of properties, on basis of which, the KMIP client gets registered. KMIP clients can be registered manually by:

- Creating a new certificate using local CA on the CipherTrust Manager.
- Using a certificate signed by an external CA.

Refer to [To Register a new KMIP Client](#) for details.

Note

- CipherTrust Manager 2.2.0 onward, it is mandatory to register the KMIP clients.
- For KMIP client administration, the user must be a member of `Key Users` and `Client Admins` groups.

Specifying LDAP and Multi-Domain Client Usernames in KMIP Certificates

The connection and/or username can be specified in the following formats:

- `<username>` - A local user in the root domain. For example, `joe`.
- `<ldap-connection>|<username>`

An LDAP user in the root domain. For example, `ldap-connection|joe`.

- `<domain>|<ldap-connection>|<username>`

An LDAP user in a specific domain. For example, `domain|ldap-connection|joe`.

- `<domain>||<username>`

A root domain user assigned in a specific domain. For example, `domain||joe`.

- `<domain>|<auth-domain>||<username>`

A local user created inside a specific domain, named "auth-domain". For example, `domain|auth-domain||joe`.

Note

To authenticate a domain user:

- a. Add the trusted CA of the domain to the root domain's external CAs list.
- b. Add that external CA of the root domain to the KMIP interface's trusted external CAs list.

When browsing the GUI, users can be identified from the source property. The user is displayed in either of the formats listed above. A local user is displayed as `local|<username>`.

KMIP Client Registration for existing CipherTrust Manager and external CA(s)

1. Create a registration token to register KMIP clients.

Request

```
ksctl clientmgmt tokens create --configfile config_example.yaml
```

Response

```
{
  "id": "dd8289f4-adbe-412f-9191-320bf2bac847",
  "uri": "kylo:kylo:munshi:tokens:dd8289f4-adbe-412f-9191-320bf2bac847",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2019-12-19T05:52:36.590956451Z",
  "updatedAt": "2019-12-19T05:52:36.590956451Z",
  "token": "RwIgd6RfXd7kzri12ydzoeFXsA2R9ju2YDxMv2xM9duQbqAhcKcyMQ020pqZesti",
  "valid_until": "0001-01-01T00:00:00Z",
  "max_clients": -1,
  "cert_duration": 730,
  "clients_registered": 0,
  "ca_id": "713172e0-02be-48c2-be60-6aae964ce943",
  "label": null
}
```

2. Update the KMIP Interface for which you want to allow Auto registration of KMIP clients. For auto registration, the client certificate which is to be registered should be signed by the trusted CA for the interface. To update the KMIP interface, run the `ksctl interfaces modify` command with the following parameters.

- `-a (true | false)` is for enabling or disabling auto registration
- `-r` is for providing registration token as input

If `-a` is set to `true`, then providing registration token in `-r` is mandatory.

Note

A new KMIP interface can also be created with the `-a` and `-r` parameters.

Note

If the client is auto registered using the `tls-pw-opt` mode, then that certificate cannot be used to connect via the other two modes, `tls-cert-pw-opt` and `tls-cert-and-pw`.

Request to update interface with auto registration on:

```
ksctl interfaces modify -n kmip -a true -r RwIgd6RfXd7kzri12ydzoeFXsA2R9ju2YDxMv2xM9duQbqAhcKcyMQ020pqZesti --configfile config_example.yaml
```

Response

```
{
  "id": "54523325-62d0-40d4-a45e-f45b9e1679e7",
  "name": "kmip",
  "mode": "tls-cert-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:713172e0-02be-48c2-be60-6aae964ce943",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:713172e0-02be-48c2-be60-6aae964ce943"
    ]
  },
  "createdAt": "2019-12-13T03:27:28.767622Z",
  "updatedAt": "2019-12-19T05:55:26.362558Z",
  "default_connection": "local_account",
  "port": 5696,
  "network_interface": "all",
  "interface_type": "kmip",
  "minimum_tls_version": "tls_1_2",
  "local_auto_gen_attributes": {
    "cn": "kmip.keysecure.local",

```



```

    "email_addresses": [
        "support@gemalto.com"
    ],
    "names": [
        {
            "C": "US",
            "ST": "MD",
            "L": "Belcamp",
            "O": "Gemalto",
            "OU": ""
        }
    ],
    "generated": false
},
"auto_registration": true,
"registration_token": "RwIgd6RfXd7kzri12ydzoeFXsA2R9ju2YDxMv2xM9duQbqAhcKc
yMQ020pqZesti"
}

```

Note

`cert_user_field` specifies how the user name is extracted from the client certificate. The default value is CN. This user must already exist. This parameter must be one of the following:

- CN: common name
- OU: organizational unit
- SN: surname
- E: Email address
- UID: User ID
- E_ND: Email without domain; the data to the left of the @ sign in the email address is taken as the user name

3. Connect to the KMIP interface using the client certificate you want to register. It will automatically register the KMIP client corresponding to that certificate.

Note

If the client is mistakenly registered without a username, then you need to manually delete the registered client, before creating a new client. To delete a registered client, log into the console, navigate to **KMIP > Registered Clients**, and delete the client.

To Register a new KMIP Client

1. Create a json file for specifying properties to create a KMIP client profile. There are two ways to create a client profile:
 - Using csr
 - Using csr parameter

A detailed explanation of input parameters is provided after the examples.

profile.json: This file is a user input to CLI for providing profile parameters.

Note

The profile.json is just a place holder for name of file, user can choose any name for this file.

Note

- To register a KMIP client using a client certificate generated by an external CA, during profile creation, the `cert_user_field` parameter is mandatory; whereas, the `csr` and `csr` parameters are not required.
- To register a KMIP client in a domain, first, authenticate the client in that domain using any of the domain [authentication methods](#).

The authentication of the registered client certificate is based on the Subject DN of the certificate.

Note

Email ID is also part of subject DN of the certificate. An email ID mismatch can result in subject DN mismatch.

Example 1

```
{
  "cert_user_field": "CN",
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIIBCDCBjwIBADAQM...T0KnChXGc
6JazA==\n-----END CERTIFICATE REQUEST-----\n"
}
```

Example 2

```
{
  "cert_user_field": "CN",
  "csr_cn": "user1",
  "csr_org_name": "Thales",
  "csr_org_unit": "Security",
  "csr_email": "abc@example.com",
  "csr_city": "Noida",
  "csr_state": "UP",
  "csr_country": "IN"
}
```

**** Parameter Explanation****

- **cert_user_field**: Specifies how the user name is extracted from the client certificate. The default value is **CN**. This user must already exist. For more information, click [here](#).

Note

After registering the new KMIP client, cert_user_field setting of profile is used instead of the setting present in the interface.

- **csr**: This is certificate signing request. The common name specified in CSR will be used to authenticate KMIP request.

Note

If the user is not present, then profile registration will fail.

Other parameters

If CSR is not provided, then CipherTrust Manager creates CSR using below input parameters from `profile.json`:

- `csr_cn`: Common Name used to create CSR.
 - `csr_org_name`: Organization name to be used in CSR.
 - `csr_uid`: User ID used to create CSR.
 - `csr_org_unit`: Organizational unit to be used in CSR.
 - `csr_email`: Email address to be used in CSR.
 - `csr_city`: City name to be used in CSR.
 - `csr_state`: State name to be used in CSR.
 - `csr_country`: Country name to be used in CSR.
2. (Optional Step) Create a json file (for example, `device_cred.json`) for specifying device credential properties.

Note

This step is optional and should be performed only when it is desired to authenticate the KMIP Client using device credentials.

Example 1

All fields in the example are optional, however, at least one field must be specified for a successful authentication using device credentials.

```
{
  "serial_no": "12345",
  "password": "example_password_123",
  "device_id": "example_device_ID",
```

```

    "network_id": "sample_Network_ID",
    "machine_id": "MAC123456789",
    "media_id": "Media123456789"
  }

```

Parameters

- **serial_no**: Serial number of the device, for example the hardware serial number of the device.
 - **password**: Password for the device.
 - **device_id**: Device Identifier of the device.
 - **network_id**: Network identifier of the device, for example the MAC address for Ethernet connected devices.
 - **machine_id**: Machine identifier of the device, for example the client aggregator identifier, such as a tape library aggregating tape drives.
 - **media_id**: Media identifier of the device, for example the volume identifier used for a tape cartridge.
3. (Optional Step) Specify **do_not_modify_subject_dn** or **subject_dn_field_to_modify**:
- **do_not_modify_subject_dn**: Specifies whether Subject DN in the CSR is allowed to be modified or not. If this flag is set to **true**, then Subject DN must be unique across all the CipherTrust Manager clients, otherwise client registration will not be allowed. The default value is **false**. For details, refer to [Renew Local CA Client Certificates](#).
 - **subject_dn_field_to_modify**: Makes Subject DN unique so that the client can be identified uniquely. The default value is **UID**. For details, refer to [Renew Local CA Client Certificates](#).

Note

Email ID is also part of subject DN of the certificate. Therefore, an email ID mismatch can result in subject DN mismatch.

Example 1

```

ksctl kmip createprofile -p kmipprofile1 -c profile.json --configfile config_example.yaml --do_not_modify_subject_dn true

```

Example 2

```
ksctl kmip createprofile -p kmipprofile1 -c profile.json --configfile config_example.yaml --subject_dn_field_to_modify 'DNQ'
```

4. Create the KMIP profile using the json file created in step 1.

Request

```
ksctl kmip createprofile -p kmipprofile1 -c profile.json --configfile config_example.yaml
```

Note

When you create a client profile through KMIP, it creates a profile with the reserved prefix `kmip` in the generic client profile.

Example

Profile name provided on KMIP:

Profile name = "test"

Internally generates generic client profile with name:

profile name = "kmip-test"

If the same client profile name already exists (for example, "kmip-test") in the generic client profile section, the KMIP profile naming convention issue occurs and the name conflict error is displayed. For details on generic client profile, refer to [Managing Client Profiles](#).

Note

If using device credentials, use the following request instead:

```
ksctl kmip createprofile -p kmipprofile1 -c profile.json -d device_cred.json --configfile config_example.yaml
```

Here, `device_cred.json` is the device credential file created in previous step.

Response

```
{
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "createdAt": "2019-12-18T10:19:32.745104774Z",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
}
```

```
{
  "id": "70d92635-7e9f-4a9b-9161-d7f41b4e0109",
  "name": "kmipprofile1",
  "properties": {
    "csr_cn": "rajesh"
  },
  "updatedAt": "2019-12-18T10:19:32.745104774Z",
  "uri": "kylo:kylo:nae:kmip_profile:kmipprofile1"
}
```

5. Create a registration token using the profile created in the previous step.

If client registration is through local CA:

Request

```
ksctl kmip createkmipregtoken -p kmipprofile1 --configfile config_example.yaml
```

Note

The `profile_name` is required and must be present in the system before creating a registration token.

Response

```
{
  "id": "6b8f5397-b738-4160-8c39-7979bae280aa",
  "uri": "kylo:kylo:munshi:tokens:6b8f5397-b738-4160-8c39-7979bae280aa",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2019-12-18T10:29:26.701363932Z",
  "updatedAt": "2019-12-18T10:29:26.701363932Z",
  "token": "LW0heyZNXHE9SEKmu7B0zVqnwHns42tahMVWJYUKP6I5CI67RQLS0EDkZ96Uioc
I",
  "valid_until": "0001-01-01T00:00:00Z",
  "max_clients": -1,
  "cert_duration": 730,
  "clients_registered": 0,
  "ca_id": "efe1d302-7ff4-46ca-a2a4-c4d9236cc0f8",
  "label": {
    "KmiclientProfile": "kmipprofile1"
  }
}
```

```
}
}
```

If client registration is through external CA:

Request

```
ksctl kmip createkmipregtoken -p kmipprofile1 --configfile <config_example.yaml> --ca_id <CA-Id-of-the-External-CA>
```

Note

To register a client using external CA, make sure to upload the external CA to CipherTrust Manager, and the `ca_id` parameter must be set to the ID of the uploaded external CA.

Response

```
{
  "id": "6b8f5397-b738-4160-8c39-7979bae280aa",
  "uri": "kylo:kylo:munshi:tokens:6b8f5397-b738-4160-8c39-7979bae280aa",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2019-12-18T10:29:26.701363932Z",
  "updatedAt": "2019-12-18T10:29:26.701363932Z",
  "token": "LW0heyZNXHE9sEKmu7B0zVqnwHns42tahMVWJYUkP6I5CI67RQLS0EDkZ96UiocI",
  "valid_until": "0001-01-01T00:00:00Z",
  "max_clients": -1,
  "cert_duration": 730,
  "clients_registered": 0,
  "ca_id": "efe1d302-7ff4-46ca-a2a4-c4d9236cc0f8",
  "label": {
    "KmipClientProfile": "kmipprofile1"
  }
}
```

6. Register the new KMIP client using the registration token generated in the previous step.

If client registration is through local CA:

Request

```
ksctl kmip register -n Kmiclient2 -t LW0heyZNXHE9sEKmu7B0zVqnwHns42tahMVWJYUk
P6I5CI67RQLS0EDkZ96UiocI --kmipCertOutFile ClientCert.pem --kmipKeyOutFile Cli
entKey.pem --configfile config_example.yaml
```

Response

```
{
  "cert": "-----BEGIN CERTIFICATE-----4xq+8sZ6PuwxePopPD2HWB+uxjLjdaKb01wz5A
Ek...-
N3e2aj0ow/ZC2FxtkipCIyYdlTJ7P\n-----END CERTIFICATE-----\n",
  "client_ca": "-----BEGIN CERTIFICATE-----\kXLCgQWnp2Kp445v2ZAYf3CUwWnBBUzL
ZdpDkUkQpaloJaMkNXU2KFVxjyDT7cXM...
v5jwHaYfqXImDs0bC/UZyEo+bnwFlA==\n-----END CERTIFICATE-----\n",
  "client_id": "ac8c7131-7dc4-4b1e-ab42-3529b4c4a497",
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nwSnPtpvJZU9kac0...2LfMJxmbnmA
ug2YM/\n-----END CERTIFICATE REQUEST-----\n",
  "key": "-----BEGIN RSA PRIVATE KEY-----zMTurULUu9S9dc89ajkKJ...TlkMA/Gx0IO
kpH1zMbJLf6NEga0ZbC1tCLjxB6YjQIgHfrTqupc=\n-----END RSA PRIVATE KEY-----\n"
}
```

If client registration is through external CA:

Request

```
ksctl kmip register -n Kmiclient2 -t LW0heyZNXHE9sEKmu7B0zVqnwHns42tahMVWJYUk
P6I5CI67RQLS0EDkZ96UiocI --configfile config_example.yaml --cert <Path-to-Exte
rnal-Client-Certificate>
```

Response

```
{
  "cert": "-----BEGIN CERTIFICATE-----4xq+8sZ6PuwxePopPD2HWB+uxjLjdaKb01wz5A
Ek...-
N3e2aj0ow/ZC2FxtkipCIyYdlTJ7P\n-----END CERTIFICATE-----\n",
  "client_ca": "-----BEGIN CERTIFICATE-----\kXLCgQWnp2Kp445v2ZAYf3CUwWnBBUzL
ZdpDkUkQpaloJaMkNXU2KFVxjyDT7cXM...
v5jwHaYfqXImDs0bC/UZyEo+bnwFlA==\n-----END CERTIFICATE-----\n",
  "client_id": "ac8c7131-7dc4-4b1e-ab42-3529b4c4a497",
  "csr": "",
  "key": ""
}
```

Note

To register a client using an external CA, the `cert` parameter should specify the path of the client certificate.

7. Use **Client Certificate** and **Client Private Key** to connect with the server.

User Impersonation

The CipherTrust Manager allows a client to impersonate a user for a client, meaning, the client can connect to the CipherTrust Manager as a user without providing the user credentials, on NAE and KMIP interfaces. By default, the user impersonation functionality is limited to impersonate users of the clients' domain (where the client is registered) only. Hence, user impersonation applies only to the users created in the clients' domain or assigned users (users created in the clients' domain and assigned to some other domain).

If you set the `ALLOW_USER_IMPERSONATION_ACROSS_DOMAIN` property to `true`, clients belonging to the `root` domain are allowed to impersonate the users created within sub-domains. The property is only applicable to NAE and KMIP interfaces. The default value of this property is `false`.

```
ksctl properties modify --name ALLOW_USER_IMPERSONATION_ACROSS_DOMAIN --value true
```

Note

Make sure to restart the NAE service when you change the value of "ALLOW_USER_IMPERSONATION_ACROSS_DOMAIN" property.

KMIP Client Renewal

The CipherTrust Manager allows you to renew the certificate of a registered KMIP Client.

Note

The KMIP clients should use **Admin's Renew Route** for renewal. For details, refer to [Renewing Client Certificates](#).

Supported Objects, Operations, Attributes, and Key Formats

The following KMIP Objects Types, Attributes, and Operations supported on CipherTrust Manager are in accordance with KMIP Specification Version 1.4.

Supported Object Types

Object	Object	Object
Symmetric Key	Public Key	Private Key
Secret Data	Opaque Object	Certificate

Supported Operations

Operation	Operation	Operation	Operation
Activate	Delete Attribute	Get Attributes List	Register
Add Attribute	Destroy	Locate	Re-key
Check	Discover Versions	MAC	Re-key Key Pair
Create	Encrypt	MACV	Revoke
Create Key Pair	Get	Modify Attribute	Sign
Decrypt	Get Attributes	Query	SignV

Note

CipherTrust Manager supports Wrapping and Unwrapping of Keys for Get and Register operations. To read more, go to [Wrap/Unwrap support in KMIP](#) for details.

Supported Attributes

Attribute	Attribute	Attribute
Unique Identifier	Name	Object Type
Cryptographic Algorithm	Cryptographic Length	Cryptographic Parameters
Cryptographic Domain Parameters	Cryptographic Usage Mask	Digital Signature
State	Initial Date	Activation Date
Process Start Date	Protect Stop Date	Deactivation Date
Destroy Date	Compromise Occurrence Date	Compromise Date
Revocation Reason	Link	Application Specific Information
Contact Information	Object Group	Custom Attribute
Fresh Attribute	Alternate Name	Digest Attribute
Certificate Type	Certificate Length	X.509 Certificate Identifier
X.509 Certificate Subject	X.509 Certificate Issuer	Certificate Identifier
Certificate Subject	Certificate Issuer	Never Extractable

Attribute	Attribute	Attribute
Extractable	Operation Policy Name	

Note

For information on the above listed object Types, Operations, and Attributes, refer to the OASIS KMIP Documentation.

Note

The Operation Policy Name attribute supports only 'default' value.

Note

The Fresh Attribute, Custom Attribute, Application Specific Information, and Digest Attribute are stored in the Key Meta.

Example 1

If a custom attribute (for example, `AttributeName : x-attribute1`, `AttributeValue: value1`) is added to the key `k1`, then the attribute is stored in Key Meta as:

```
{
  "kmip": {
    "custom": [
      {
        "type": "TextString",
        "x-attribute1": "Value1"
      }
    ],
    "ownerId": "local|261d8e87-73bb-4106-aa3d-06ed92e2f010",
    "permissions": {}
  }
}
```

Example 2

If an application specific attribute (for example, `application_data": "data"`, `"application_namespace": "Test"`) is added to a key, then the attribute is stored in Key Meta as:

```
{
  "kmip": {
    "app_specific_info": [
```

```

    {
      "application_data": "data",
      "application_namespace": "TestASi"
    }
  ],
  "ownerId": "local|261d8e87-73bb-4106-aa3d-06ed92e2f010",
  "permissions": {}
}

```

Supported Key Formats

Object Type	Key Formats
Symmetric	• Raw • Opaque
Private Key	• PKCS_1 • PKCS_8 • PKCS_12
Public Key	• PKCS_1 • PKCS_8
Certificate	• PKCS_12 • X.509
Opaque	• Raw
Secret	• Opaque

Wrap/Unwrap support in KMIP

Key wrapping is used for encrypting data keys. This provides an additional layer of security in protecting keys when they are at rest or are being transmitted over insecure networks. A variety of key wrapping algorithms can be used to encrypt the data keys. Similarly, unwrap mechanisms are used to decrypting the data keys.

CipherTrust Manager supports wrapping/unwrapping of keys under Get and Register operations.

Note

Only Symmetric and Private keys can be wrapped/unwrapped in KMIP.

Supported wrapping methods are:

- Encrypt
- MACSign

Note

For Encrypt Wrapping Method:

- Only AES and RSA keys can be used to wrap the symmetric keys.
- Only AES-256 keys can be used to wrap private keys.

Note

For MACSign Wrapping Method:

- Only HMAC keys can be used for MAC request.
- Only RSA keys can be used for Sign request.

Encrypt Wrapping Method

Symmetric Data Keys	Symmetric Data Keys	Private Data Keys
Supported BlockCipher Modes for AES wrap keys	Supported padding methods for RSA wrap keys	Supported BlockCipher Mode for AES wrap keys
• NISTKeyWrap (RFC-3394) • AESKeyWrapPadding (RFC-5649)	• PKCS1V1_5 • OAEP Supported Hash Algorithms for OAEP padding method: • SHA_1 • SHA_256 • SHA_384 • SHA_512	• AESKeyWrapPadding (RFC-5649)

MACSign Wrapping Method for Symmetric and Private Keys

MAC Request	Sign Request	Sign Request
Supported HMAC algorithms for HMAC wrap keys	Supported padding methods for RSA wrap keys	Supported DigitalSignatureAlgorithms for RSA wrap keys
• HMAC_SHA_1 • HMAC_SHA_256 • HMAC_SHA_384 • HMAC_SHA_512	• PKCS1V1_5 (Default) • PSS Supported Hash Algorithms for above padding methods: • SHA_1 • SHA_256 • SHA_384 • SHA_512	• SHA_1WithRSAEncryption • SHA_256WithRSAEncryption • SHA_384WithRSAEncryption • SHA_512WithRSAEncryption

Supported Encodings

- NoEncoding
- TTLVEncoding

For more information, visit [OASIS Key Management Interoperability Protocol Documentation](#).

Opaque Key Format for Symmetric Keys

The CipherTrust Manager supports opaque key format type for symmetric keys in the REST/KMIP interface. The opaque key format type can be used during importing keys when key material is not raw. For more details, refer to [OASIS Key Management Interoperability Protocol Documentation](#).

Example Request

```
<RequestMessage>
  <RequestHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="2"/>
    </ProtocolVersion>
    <BatchCount type="Integer" value="1"/>
  </RequestHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Register"/>
    <RequestPayload>
      <ObjectType type="Enumeration" value="SymmetricKey"/>
      <TemplateAttribute>
        <Attribute>
          <AttributeName type="TextString" value="Name"/>
          <AttributeValue>
            <NameValue type="TextString" value="OpaqueKey_AES128_1"/>
            <NameType type="Enumeration" value="UninterpretedTextString"/>
          </AttributeValue>
        </Attribute>
        <Attribute>
          <AttributeName type="TextString" value="Cryptographic Usage Mask"/>
          <AttributeValue type="Integer" value="Decrypt Encrypt"/>
        </Attribute>
      </TemplateAttribute>
      <SymmetricKey>
        <KeyBlock>
          <KeyFormatType type="Enumeration" value="Opaque"/>
          <KeyValue>
            <KeyMaterial type="ByteString" value="000102030405060708090a0b0c0d0e0f000102030405060708090a0b0c0d0e0f000102030405060708090a0b0c0d0e0f000102030405060708090a0b0c0d0e0f000102030741"/>
          </KeyValue>
          <CryptographicAlgorithm type="Enumeration" value="AES"/>
          <CryptographicLength type="Integer" value="128"/>
        </KeyBlock>
      </SymmetricKey>
    </RequestPayload>
  </BatchItem>
</RequestMessage>
```

Example Response

```
<ResponseMessage>
  <ResponseHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1" />
      <ProtocolVersionMinor type="Integer" value="2" />
    </ProtocolVersion>
    <TimeStamp type="DateTime" value="2022-02-15T07:31:37+00:00" />
    <BatchCount type="Integer" value="1" />
  </ResponseHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Register" />
    <ResultStatus type="Enumeration" value="Success" />
    <ResponsePayload>
      <UniqueIdentifier type="TextString" value="88021e1314ca4e03a6dac2ffd593a
c91f801970257f240599224075c3e3f5d6d" />
    </ResponsePayload>
  </BatchItem>
</ResponseMessage>
```

Cryptographic Operations for KMIP

Using KMIP interface, you can perform following cryptographic operations:

Operations	Algorithms	Modes	Padding
Encrypt/Decrypt	AES	• ECB (default) • CBC • GCM • CTR	• PKCS5 (default) • None
Encrypt/Decrypt	TDES	• ECB (default) • CBC	• PKCS5 (default) • None
Encrypt/Decrypt	RSA	not applicable	• PKCS1V1_5 (default) • OAEP
MAC/MACV	HMAC_SHA1	not applicable	not applicable
MAC/MACV	HMAC_SHA256	not applicable	not applicable
MAC/MACV	HMAC_SHA384	not applicable	not applicable
MAC/MACV	HMAC_SHA512	not applicable	not applicable
MAC/MACV	AES	CMAC	not applicable

Caution

We recommend that you set the **RandomIV** object in the **Cryptographic Parameters** attribute to true for AES CBC, AES GCM, and TDES CBC encryption requests to have CipherTrust Manager generate a unique random IV with each request. The size of a generated random IV is always fixed to the cipher block size. CipherTrust Manager does not use Cryptographic Parameter objects such as **Fixed Field**

Length, Invocation Field Length or Counter Length to construct the random IV.

Otherwise, you can provide your own IV in the Request Payload as an **IV/Counter/Nonce** object. If you or your KMIP vendor are providing IVs, follow [NIST SP800-38A](#) and [NIST SP800-38D](#) to ensure IVs are sufficiently unique and secure.

Recommended Curve Mappings

The recommended Curve mappings along with the respective Curve IDs are listed in the following table.

Curve	Curve ID
P-224	secp224r1
P-256	prime256v1
P-384	secp384r1
P-521	secp521r1

KMIP Interface Mode

The "interface mode" configuration parameter specifies the KMIP Interface mode, and must be one of the following:

- **tls-pw-opt:** TLS, allow anonymous logins.
 - Username is not extracted from the client certificate.
 - If authentication request is sent in KMIP request, Username will be the owner of the key.
 - If authentication request is not sent in KMIP request, the Key Owner will be the 'global' user.
- **tls-cert-pw-opt:** TLS, user name taken from client cert, auth request is optional.
 - Username is extracted from the registered client's Common Name specified during client's registration.
 - If authentication request is sent in KMIP request, then the Username in the authentication request will be the owner of the key.
- **tls-cert-and-pw:** TLS, password is needed, user name specified during client's registration must match user name in authentication request.
 - Username is extracted from the registered client's Common Name specified during client's registration.
 - The inclusion of the authentication request in KMIP request is mandatory.
 - The Username in the authentication request must match with the registered client's Common Name specified during client's registration.

Tip

The interface mode can be changed using the API or the CLI "ksctl interfaces modify" command.

KMIP Interface mode mapping

Following table provides a mapping between the KMIP interfaces mode from SafeNet KeySecure Classic and CipherTrust Manager.

Safenet KeySecure Classic	CipherTrust Manager
Password authentication: Not Used Client cert authentication: SSL only	tls-pw-opt TLS, allow anonymous logins, verify client cert
Password authentication: Optional Client cert authentication: SSL only	tls-pw-opt TLS, allow anonymous logins, verify client cert
Password authentication: Required Client cert authentication: SSL only	tls-pw-req
Password authentication: Not Used Client cert authentication: SSL with CN	tls-cert-pw-opt Verify client cert, user name taken from client cert, auth request is optional
Password authentication: Optional Client cert authentication: SSL with CN	tls-cert-pw-opt Verify client cert, user name taken from client cert, auth request is optional
Password authentication: Required Client cert authentication: SSL with CN	tls-cert-and-pw Verify client cert, password is needed, user name in cert must match user name in authentication request

Managing KMIP Interface

You can add, remove, or modify KMIP interfaces.

To create a new KMIP interface:

Example

```
ksctl interfaces create -o 5697 -y kmip
```

Response

```
{
  "id": "90b3b131-d6d8-4985-abdd-539162c136c3",
  "name": "kmip_all_5697",
  "mode": "tls-cert-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:c729ffe0-f6ad-49ad-8558-2db435b112c7",
}
```

```

    "trusted_cas": {
      "local": [
        "kylo:kylo:naboo:localca:c729ffe0-f6ad-49ad-8558-2db435b112c7"
      ]
    },
    "createdAt": "2019-08-22T10:10:28.05794Z",
    "updatedAt": "2019-08-22T10:10:28.05794Z",
    "default_connection": "local_account",
    "port": 5697,
    "network_interface": "all",
    "interface_type": "kmip"
  }

```

Note

Interface can only be created in Root domain. To use interfaces in other domains, add the domain certificate to the Local Trusted CA of the domain.

Adding domain CA to the interface as external CA

To add domain CA to the interface as external CA:

1. Log on to the CipherTrust Manager GUI.
2. [Switch](#) to the desired domain.

Tip

To create a new domain, click [here](#).

3. Navigate to **CA > Local**. The list of available CAs is displayed.
4. Click the ellipsis icon corresponding to the CA.
5. Click **Copy/Download** the local CA.
6. Switch to **Root** domain.
7. Navigate to **CA > External**.
8. Click **Add External CA**. The Add External Certificate screen appears.
9. Provide **Display name**.
10. Either upload or paste the **Certificate** that you copied/downloaded in the step 5.

11. Click **Add External CA**. The newly created external CA appears on the External Certificate Authorities page.
12. Navigate to **Admin Settings > Interfaces**.
13. Click **kmip** and select the external CA created in step 11 from the **External Trusted CAs** drop-down list.
14. Click **Update**.

To modify TLS version of an interface:

```
ksctl interfaces modify -n <interface name> -e <minimum tls version>
```

Here:

- `<interface name>` is KMIP interface name.
- `<minimum tls version>` can be `tls_1_0`, `tls_1_1`, or `tls_1_2`. `tls_1_1` and `tls_1_0` are deprecated and will be discontinued in a future release.

Example

```
ksctl interfaces modify -n kmip_all_5697 -e tls_1_2
```

Response

```
{
  "id": "6199c8d1-f59a-4bd1-8dfd-cf4d43900264",
  "name": "kmip_all_5697",
  "mode": "tls-cert-and-pw",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:afdebde8-a5ff-4cbd-9bb2-6dc5186e4c03",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:0a4bf97b-1294-41d2-93cd-d19e6290cfa3",
      "kylo:kylo:naboo:localca:b7860db1-b7d1-4163-996e-22b14090d55a",
      "kylo:kylo:naboo:localca:afdebde8-a5ff-4cbd-9bb2-6dc5186e4c03"
    ]
  },
  "createdAt": "2019-08-28T05:21:05.36768Z",
  "updatedAt": "2019-08-28T07:41:38.62511Z",
  "default_connection": "local_account",
  "port": 5697,
  "network_interface": "all",
  "interface_type": "kmip",
  "kmip_enable_hard_delete": 0,
  "minimum_tls_version": "tls_1_2"
}
```

Viewing and Setting Log Level

You can view or change the log level of KMIP service using the "ksctl" CLI tool.

There are two subcommands in the logs command:

- `get level`: used for viewing the current log level of a service
- `set level`: used for setting log level of a service

Using getlevel

To view the current log level for KMIP service, use the following request in CLI:

Request

```
ksctl logs getlevel --service kmip
```

Response

```
{
  "level": "INF",
  "service": "kmip"
}
```

Using setlevel

Note

Debug-level logging can generate a significant amount of detailed information that can impact the application's performance. Therefore, it is recommended **not to** configure the log level to "debug" mode.

To set current log level to debug for KMIP service, use the following request in CLI:

Request

```
ksctl logs setlevel --service kmip --level debug
```

Response

```
{
  "level": "DBG",
  "service": "kmip"
}
```

Supported levels are:

- `error` or `ERR`
- `info` or `INF`

- `debug` or `DBG`

Note

- To view the logs, go to `/opt/keysecure/logs` directory on appliance.
- To download the logs, use `ksctl logs download` command.
- Only users with administrator privileges can change or set the log level.
- When the log level is set to `debug`, sensitive data from `debug` logs is masked. To unmask the sensitive data, contact **Customer Support**.

Note

Any `ssl` connection related error message can be filtered from the log file based on `ERR` and `tls` tags.

Thales Key Management

Simplify data security by centrally managing keys and policies across the enterprise



Contents

3	Executive Summary
3	Data Security Challenges
3	Looking Back at Encryption and Key Management Techniques
4	Data Encryption Across Disparate Systems – Security Silos
4	The Business Problem - Too Many Silos
5	Essentials of Enterprise Key Management
5	Facilitate Governance and Reporting
6	Introducing CipherTrust Manager
6	The Foundation of Thales Key Management Solutions
7	Key Benefits
7	Thales Enterprise Key Management Solutions
8	Thales Hardware Security Module Solutions
8	Summary
8	About Thales

Executive Summary



Today, every IT organization is striving to protect valuable digital assets of any enterprise from accidental exposure or intentional misuse by cyber criminals. Many organizations have deployed a variety of point encryption solutions as a primary method of protecting sensitive data to meet various digital privacy regulations and compliance mandates. Unfortunately, the majority of these disparate encryption solutions have fallen short in their ability to address their enterprise key management challenges.

This white paper looks back at the evolution of encryption and key management systems, and examines the key challenges faced by IT teams around encryption systems, including regulation and compliance, complexity, lack of proper management tools. This is followed by a review of the recent industry initiatives and compliance regulations that are shaping the future of key management.

The paper concludes with an introduction to CipherTrust Manager, the next generation enterprise key management offering from Thales, which provides a powerful integrated solution that enables organizations to centrally manage encryption key lifecycle and policies for Thales Data Protection Connectors and third-party KMIP compliant products in the enterprise.

Data Security Challenges

Many organizations are adopting digital transformation and migrating to the cloud to drive operational efficiency into their data-intensive processes. According to the 2020 Thales Data Threat Report, which is based on a global-IDC survey of 1,723 executives who have responsibility or influence over IT and data security, we are at an inflection point with the cloud as half of all data is now stored in cloud environments, and 48% of that data is sensitive. Additionally, most organizations rely on multi-cloud environments to deploy their applications, in order to avoid vendor lock-in.

Data migration to third-party hosted environments and multiple cloud service providers is creating new attack surfaces for cybercriminals to exploit. Hence, data breaches continue to threaten the IT landscape at an increasing rate. All of this adds up to today's data environments becoming even more complex; operation complexity is a top barrier to data security.

" We are at an inflection point with the cloud as half of all data is now stored in cloud environments, and 48% of that data is sensitive."

– 2020 Thales Data Threat Report, IDC Survey

Looking Back at Encryption and Key Management Techniques

The Internet has been the most significant driving force in the evolution of encryption and key management technologies. Providing easy access to a company's data to those who need it- employees, partners, and customers has taken priority, with data security being implemented as an afterthought. This connection of public and private resources over the internet to any company's sensitive data has given rise to the increase in data breaches.

In response, a variety of perimeter and endpoint security controls and security policies have been adopted to defend against these threats. In addition, data privacy regulations and a myriad of compliance requirements have been developed to battle these forces. However, they are not enough to stop cyber theft.

As attacks become more sophisticated, the industry has been improving these early encryption systems, eliminating the difficulties of disparate native key repositories that were then scattered across the various information systems in the enterprise.

Data Encryption Across Disparate Systems – Security Silos

The increased adoption of encryption solutions has improved security for enterprises, but it has made life much more challenging for the IT security team, now tasked with managing a variety of cryptographic keys. Nearly all offline data storage devices and many database management systems (DBMS) include the option of embedded native encryption capability. A challenge with these islands of encryption is that keys and key management software from each provider don't usually interoperate well with one another.

The resulting silos of security, where system administrators and database administrators (DBAs) have to become the managers of the encryption keys for a particular system, distracts them from their primary tasks of IT and database administration. Along with the resource inefficiency of such a methodology, it also puts an enterprise's overall security posture at risk.

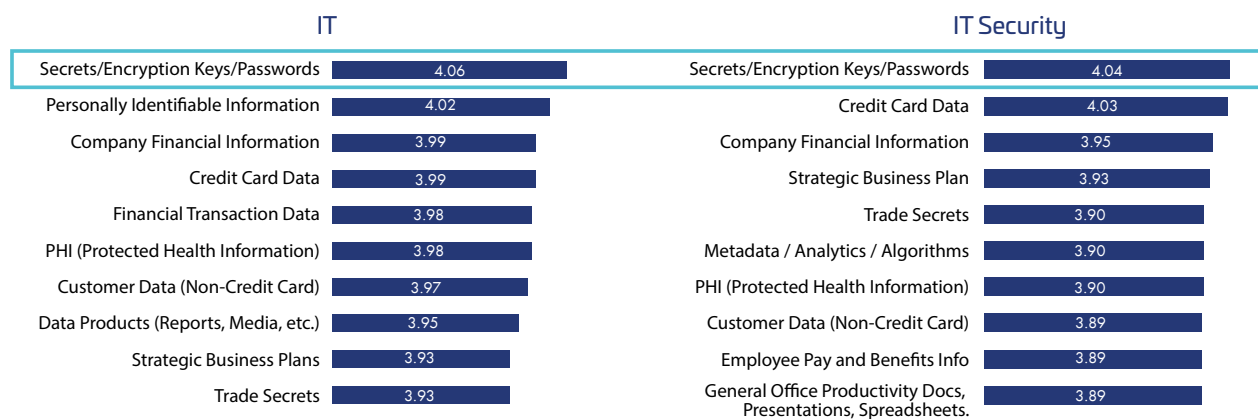


Figure 1: 2020 IDC Data Security Survey - Most Critical Assets that Require Protection

According to the 2020 IDC Data Security Survey (shown in Figure 1), both IT and IT Security practitioners rated Secrets, Encryption Keys and Passwords as the most critical assets that requires protection.

The Business Problem - Too Many Silos

Without a centralized encryption key management solution, security administrators are faced with a costly, inefficient and often impossible task managing disparate encryption keys for many different databases accumulated over time from separate vendors. This heterogeneous world means that an enterprise looking to secure databases, such as Oracle and Microsoft SQL Server using native TDE, has to factor in the increased costs and administrative resources required for managing multiple, incompatible encryption solutions. It is costly and complex to manage separate encryption systems required for each data store, without consistent security policies, repeatable processes and error prone, as shown in Figure 2 below.

When each system administrator separately controls encryption keys for each data repository they manage, the keys are generally stored in the same location as encrypted data, it leaves room for security compromises – the electronic equivalent of taping the key to the front door. Manual systems to store and transmit the encryption keys, lack of password control and the failure to revoke keys when an employee leaves the company can result in data breaches waiting to happen. And strict adherence to compliance requirements is nearly impossible in this situation.

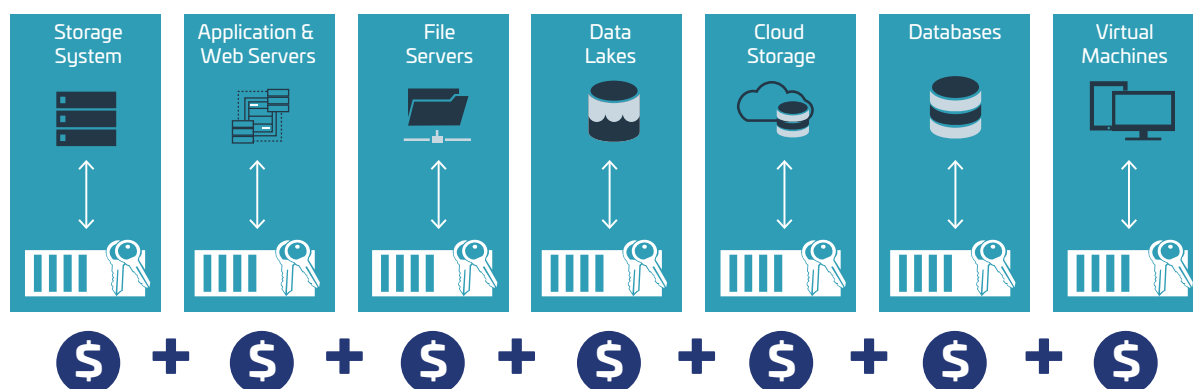


Figure 2: Costly and Complex to Manage Too Many Silos

Essentials of Enterprise Key Management

As organizations deploy an ever-increasing number of encryption solutions, they should look for a centralized key management solution that enables them to securely store and backup/restore the encryption keys, define consistent access control policies, audit all key management operations and separate encryption tasks from key management tasks.

Here are the essential elements of a robust enterprise key management solution that can help address the challenges listed in the previous section.

Secure Key Storage

Secure key storage is the foundation for any enterprise key management system. The use of Hardware Security Modules (HSMs) is a well-established approach for protecting encryption keys. Mandated in government and certain financial/payment markets HSMs protect cryptographic keys and perform various cryptographic functions in a secure tamper-resistant environment. Enterprise key management solutions should provide options to support built-in HSMs, external network attached HSM or a cloud-based HSM-as-a-service, based on the level of assurance your company needs

Centralized Key Lifecycle Management

The key management system should be able to centrally manage keys across their entire lifecycle, including secure key generation, backup/restore, clustering, deactivation, and deletion. It should also provide policy-based access control to keys, support various authentication providers including Active Directory and LDAP, and robust auditing of all key management operations.

Enabling Scalability and Flexibility

As the complexity of an organization's IT infrastructure grows from a single data-center to external hosted environments to multiple cloud service providers, so should the ability of enterprise key management solution to adopt to changing requirements.

A flexible key management solution should be able to support both on-premises infrastructure and be deployable as a virtual appliance in public cloud environments, wherever your sensitive data resides, such as AWS, Azure, Google Cloud and more.

Guaranteed Availability

Organizations cannot function without the availability of business critical data, which is most likely encrypted to reduce business risks due to unauthorized exposure. The key management system should support high-availability clustering options, using two or more physical and/or virtual appliances to reduce security lapses.

Interoperability with Third-party Systems

Any enterprise key management solution should support the following three major interoperability standards that enable you to work with multiple server, storage and device vendors, who utilize the keys for authentication, digitally signing or encrypting data.

- **PKCS#11** – Public Key Cryptographic Standard #11 specifies an API for devices to interoperate with hardware security modules (HSM) and smart cards, which hold cryptographic tokens. It is also used to access signing keys from Certification Authorities (CAs) or to enroll user certificates for digital signing and encryption using asymmetric keys. As an example, PKCS#11 is used by Oracle TDE.
- **EKM/MSCAPI** – Extensible Key Management (EKM) using the Microsoft Cryptographic APIs (MSCAPI), enables MS SQL Server to communicate with third-party key management servers. The keys must be exported from a provider before they are stored in the database. This approach enables key management that includes an encryption key hierarchy and key backup for Microsoft SQL Server Transparent Data Encryption.
- **OASIS KMIP** – Key Management Interoperability Protocol (KMIP), maintained by the Organization for Advancing Open Standards for the Information Society (OASIS), defines the standard protocol for any key management server to communicate with clients (e.g. storage devices, databases) that utilize the keys for embedded encryption. KMIP improves interoperability for key lifecycle management between encryption systems and enterprise applications.

Facilitate Governance and Reporting

The most important aspect of governance is a discipline for managing, controlling, and protecting the security and privacy of data. A policy-driven key management system forces the adherence to procedures for separation of duties and user authorization, and it automates all the security processes involved in the key lifecycle. Some of the more notable industry standards and compliance requirements affecting key management today include:

- **General Data Protection Regulations (GDPR)**

Perhaps the most comprehensive data privacy standard to date, GDPR affects any organization that processes the personal data of EU citizens - regardless of where the organization is headquartered. GDPR is designed to improve personal data protections and increase organizational accountability for data breaches. With potential fines of up to four percent of global revenues or 20 million EUR (whichever is higher), the regulation has impact on many businesses.

- **UIDAI's Aadhaar Number Regulation Compliance**

Unique Identification Authority of India (UIDAI) was established under the provisions of India's 2016 Aadhaar Act. UIDAI is responsible for issuing unique identification numbers (UIDs), called Aadhaar, and providing Aadhaar cards to all residents of India. The 12-digit UIDs are generated after the UIDAI verifies the uniqueness of enrollees' demographic and biometric information; UIDAI must protect individuals' identity information and authentication records.

- **Payment Card Industry Data Security Standard (PCI DSS)**

Any organization that plays a role in processing credit and debit card payments must comply with the strict PCI DSS compliance requirements for the processing, storage and transmission of account data. The PCI DSS standard involves assessment against over 200 tests that fall into 12 general security areas representing six core principles. These PCI DSS tests span a wide variety of common security practices along with technologies such as encryption, key management, and other data protection techniques.

- **Gramm Leach Bliley Act (GLBA)**

The United States requires that firms acknowledge publicly when a disclosure event occurs, taking whatever damages to their reputation or market position that such a statement would entail. Led by California's Database Security Breach Notification Act in 2003, more than half of all states have passed additional rules beyond the general notification requirements of GLBA to require firms to notify disclosure victims of the event, with higher associated costs to the business than GLBA exacted.

- **The U.S. Health Information Technology for Economic and Clinical Health (HITECH) Act**

HITECH does not require breach notification if the Protected Health Information (PHI) being exposed in the event of a data breach is already encrypted. For "unsecured PHI data", notification of the breach to every individual affected must be made. With the increasing cost of recovering from a data breach, it is important to invest in a strong encryption strategy for PHI data, and centralized key management system as its underpinning.

Introducing CipherTrust Manager

The Foundation of Thales Key Management Solutions

CipherTrust Manager enables organizations to centrally manage encryption key lifecycles and policies, independent of where the data resides, and help them meet their data privacy and compliance requirements. It simplifies key lifecycle management tasks, including secure key generation, backup/restore, clustering, deactivation, and deletion. It provides role-based access control to keys and policies, multi-tenancy support with ultimate separation of duties, and robust auditing and reporting of all key management operations.

The unified management console in CipherTrust Manager makes it easy to discover and classify data, and protect sensitive data using a comprehensive set of data protection connectors from Thales.

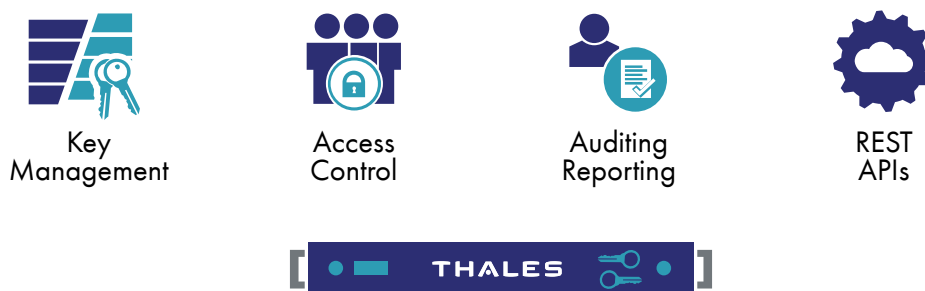


Figure 3: CipherTrust Manager

Key Benefits

Simplified Management

CipherTrust Manager provides a unified management console that enables you to discover and classify sensitive data, and protect data using integrated set of Thales Data Protection connectors across on-premises data stores and multi-cloud deployments. It offers advanced self-service licensing for improved visibility and control of licenses.

Cloud Friendly Deployment

CipherTrust Manager offers users with additional hosting options, and can run as a native virtual machine on AWS, Microsoft Azure, Google Cloud, VMware, Microsoft HyperV, and more. Additionally, native support for CipherTrust Cloud Key Manager is available on CipherTrust Manager, to streamline key management across multiple cloud infrastructures and SaaS applications.

Flexible Form Factors

CipherTrust Manager is available in both virtual and physical form factors and FIPS 140-2 levels. Flexible deployment options can easily scale to provide key management at remote facilities or in cloud infrastructures.

Thales Enterprise Key Management Solutions

Thales Enterprise Key Management solutions are built on CipherTrust Manager. They deliver a robust, standards-based platform for managing encryption keys from disparate sources across the enterprise. It simplifies the administrative challenges around encryption key management to ensure that keys are secure and always provisioned to authorized encryption services.

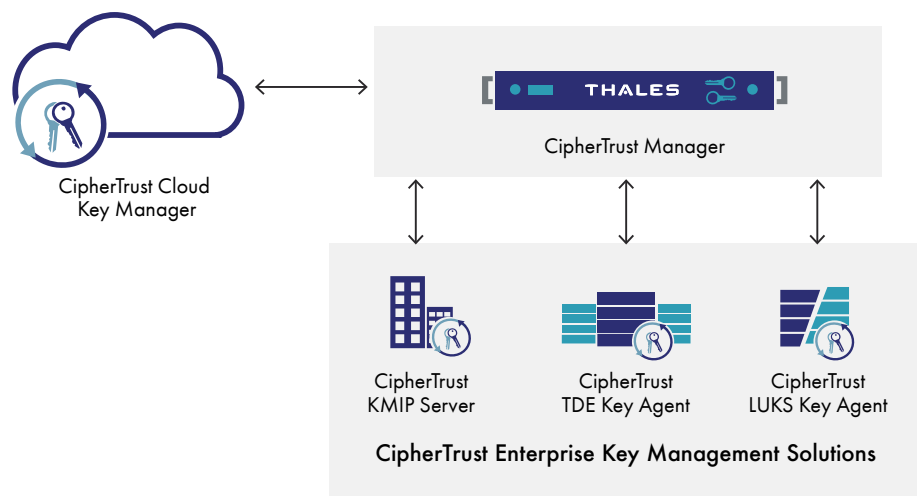


Figure 4: CipherTrust Enterprise Key Management Solutions

CipherTrust Enterprise Key Management solutions enable organizations to centrally manage and store cryptographic keys and policies for the following third-party products as shown in Figure 4. The following three elements comprise the CipherTrust Enterprise Key Management solution.

- **KMIP Server:** provides a simple interface to the key management functions on the CipherTrust Manager, delivering powerful, centralized key management capabilities for KMIP-compliant applications/systems such as SAN and NAS storage arrays, self-encrypting drives, and hyper-converged infrastructure solutions.
- **TDE Key Agents:** CipherTrust Manager centralizes and simplifies key management across Oracle TDE and Microsoft SQL Server TDE using TDE Key Agents, and keeps the TDE keys separate from your database servers.
- **LUKS Key Agent:** Linux Unified Key Setup (LUKS) provides transparent disk encryption for Linux. The LUKS Key Agents enables you to centrally manage encryption keys for Linux disk partitions.

In addition, CipherTrust Manager integrates with CipherTrust Cloud Key Manager to manage keys for multi-cloud environments, such as AWS, Azure, Salesforce and more.

Thales Hardware Security Module Solutions

Thales offers a variety of hardware security modules (HSMs), which are the highest performing, most secure and easiest to integrate in the market today. They act as trust anchors to protect the master keys that encrypt your data and digital identities in a high assurance FIPS 140-2 Level 3-certified, tamper-resistant appliance. Thales offers the following types of purpose-built HSMs:

- General Purpose HSMs: Luna HSMs come in several form-factors — a network attached appliance, an embedded PCI module, and a portable USB appliance. They can be easily integrated with a wide-range of applications to accelerate general cryptographic operations. The CipherTrust Manager model k570, includes a built-in Luna HSM card.
- Cloud HSMs: Data Protection On Demand (DPoD) is a cloud-based platform that provides a wide range of Cloud HSM and key management services through a simple on-line marketplace.

Summary

Data is only as secure as the system that manages the encryption keys protecting that data. A centralized enterprise key management solution is critical to ensuring all sensitive enterprise data is secure and available. CipherTrust Key Management from Thales help organizations maximize IT efficiency through a centralized, extensible platform based solution, and support the burdens of encryption key management across the enterprise and into the cloud – without disrupting existing application or database environments.

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.



Americas

Arboretum Plaza II, 9442 Capital of Texas Highway North,
Suite 100, Austin, TX 78759 USA
Tel: +1 888 343 5773 or +1 512 257 3900
Fax: +1 954 888 6211 | E-mail: sales@thalessec.com

Asia Pacific – Thales Transport & Security (HK) Ltd

Unit 4101-3, 41/F, Sunlight Tower, 248 Queen's Road East
Wanchai, Hong Kong | Tel: +852 2815 8633
Fax: +852 2815 8141 | E-mail: apacsales.cpl@thalesgroup.com

Europe, Middle East, Africa

350 Longwater Ave, Green Park,
Reading, Berkshire, UK RG2 6GF
Tel: +44 (0)1844 201800 | Fax: +44 (0)1844 208550
E-mail: emea.sales@thales-esecurity.com

> cpl.thalesgroup.com <



Enterprise Key Management for Storage Infrastructure



Security conscious organizations are adopting Key Management Interoperability Protocol (KMIP) compliant integrations of their existing storage infrastructure with Enterprise Key Management (EKM) systems. KMIP simplifies the requirement of separating keys from encrypted data, and allows those keys to be centrally managed with a common set of policies from a corporate perspective. Thales offers CipherTrust Manager as the central enterprise key management solution for an expansive ecosystem of storage and archive infrastructure vendors.

Overview

Encryption is fundamental to any defense-in-depth strategy whether the goal is compliance or securing sensitive data. Purpose built or software defined storage solutions are an effective way to deploy encryption (using either self encrypting drives or controller based) in large-scale storage environments. However, as the number of drives or storage end points increase, so does the complexity with increasing number of encryption keys, key stores, and access policies needing management. To cost-effectively support such an environment and bring it into regulatory compliance, enterprise key management must be part of a secure solution.

Solution

Thales' CipherTrust Manager integrates with a multitude of storage products to provide robust, enterprise-scale key management, ensuring that access keys are managed throughout their lifecycle and properly secured with FIPS 140-2 certified internal or external Hardware Security Module (HSM). CipherTrust Manager is also available as a virtual appliance providing organizations with a scalable less expensive alternative to using a physical appliance.

CipherTrust Manager Integrates with Various Storage Solutions

CipherTrust Manager is pre-integrated with a variety of storage and archive solutions via the Key Management Interoperability Protocol (KMIP) to centrally manage key life-cycle for encrypting/decrypting data. Consolidating the policy and key management of servers, databases, and file storage streamlines security administration. Centralized key management improves security by making key creation, rotation, and deletion easier, while also separating duties so that no single administrator is responsible for the entire environment. Additionally, unifying and centralizing policy management, logging, and auditing makes information more readily accessible which, in turn, makes demonstrating compliance with data governance requirements simple.

Virtual CipherTrust Manager allows organizations to utilize a secure virtual appliance to centrally manage encryption keys and enforce access control across cloud infrastructures. It also ensures that organizations maintain ownership of their encryption keys at all times by hardening the appliance OS and encrypting the entire virtual appliance for enhanced key security and protection against snapshot attacks. CipherTrust Manager provides scalable key management and secure encryption at remote facilities or cloud infrastructures.

Summary of Benefits:

Thales' CipherTrust Manager provides customers with complete control by securing the keys needed to access the storage system. Together, Thales and their storage partners offer a secure and efficient means of protecting data at rest with encryption, and help customers meet data security compliance mandates such as FIPS 140-2, PCIDSS, HIPAA, and GDPR.



Centralized Management of Access Keys

Centralize and simplify key life-cycle management (e.g. key generation, rotation, etc.) for various storage platforms and KMIP-compatible encryption solutions, while improving compliance and auditability.

High-Availability Configurations

Cluster multiple CipherTrust Manager appliances to maintain encrypted data availability, even in geographically dispersed data centers.

Hardware or Software Appliance

CipherTrust Manager is available in both physical and virtual form factors. By utilizing CipherTrust Manager, organizations benefit from its flexible options for secure and centralized key management – deployed in physical, virtualized infrastructure, and public cloud environments.

Separation of Duties

CipherTrust Manager supports segmented key ownership and management based on individuals or group owners. This approach is perfect for dividing large enterprises into multiple administrative domains based on organizational structure.

Ecosystem Support

When there are dispersed data repositories with varying security concerns among lines of businesses consuming these repositories, CipherTrust Manager enables businesses to bring all these disparate efforts under one umbrella from a policy management viewpoint. The CipherTrust Manager EKM solutions including the KMIP technology standard help in providing a consistent interface to applications and appliances interfacing with the external key management system.

Conclusion—Secure the Breach

Encrypting data in the storage environment is critical to ensuring that data is safe in the event of a data breach. Thales has partnered with a number of storage vendors to offer organizations the ability to secure data through encryption without making the management of the necessary encryption keys and policies unwieldy or difficult.

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.

Encryption key management solutions for Microsoft® SQL Server® and Oracle® Database



Simplified and protected management for database encryption keys

- Streamlined operations through centralized key management
- Stronger security by separating keys from databases
- Comprehensive key security based on FIPS 140-2 compliant hardware and software solutions

Microsoft SQL and Oracle Database key management challenges

Microsoft SQL Server and Oracle Database solutions provide native transparent database encryption (TDE) that protects the data stored in their customers' enterprise and cloud-hosted databases. And, as with any encryption-based security scheme, securing and managing the encryption keys is critically important for robust data security.

Managing encryption keys presents challenges such as isolating them from the assets they protect and storing them securely—not only a best practice for key management, but a common industry data protection mandate. Encryption key management challenges multiply as organizations use multiple databases for different purposes, each requiring dedicated key management to ensure that keys are securely stored, backed up and available when needed.

Thales key management solutions for transparent database encryption

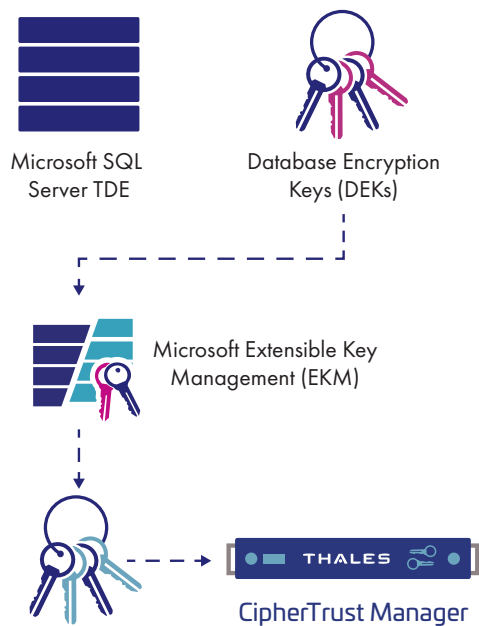
Thales key management solutions centralize key management for your enterprise and cloud-hosted Microsoft SQL Server and Oracle Database, giving you greater command over the keys while increasing your data security.

CipherTrust key management using the CipherTrust Manager

The CipherTrust Manager centrally manages keys for Microsoft SQL and Oracle TDE keys and Thales Data Protection portfolio.

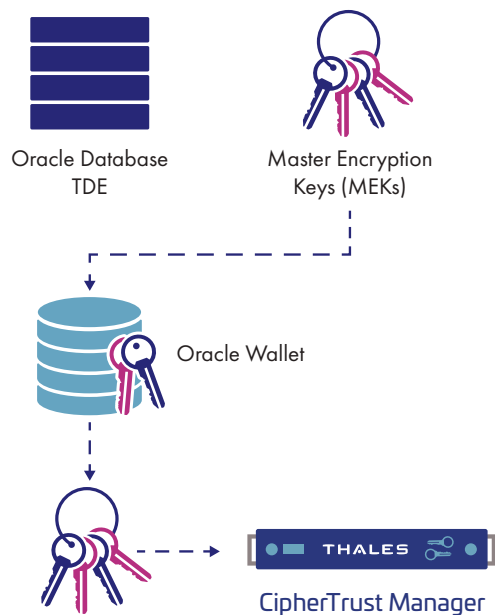
Microsoft SQL Transparent Data Encryption

Thales key management solutions complement Microsoft native TDE by providing secure storage and management of the keys used in Microsoft's database encryption scheme. Microsoft TDE encrypts the sensitive data in the SQL database using a database encryption key (DEK), and Thales interfaces with Microsoft Extensible Key Management (EKM) to store and manage the DEKs in the FIPS 140-2 compliant CipherTrust Manager.



Oracle Database Transparent Data Encryption

The CipherTrust Manager complements Oracle Database native TDE by centrally storing and managing Oracle Database encryption keys. As a part of the Oracle Advanced Security TDE two-tier key architecture, Oracle Database uses master encryption key (MEKs) to encrypt the database encryption keys (DEKs), which are used to encrypt columns and tablespaces within the databases. Thales key management solutions interface with the Oracle Wallet to protect and manage these MEKs within a secure FIPS-certified boundary.



Enterprise and Cloud Support

Whether your database resides on-premises or in the cloud, Thales solutions provide secure and efficient management of your database encryption keys.

CipherTrust Manager: the foundation for Thales key management solutions

The CipherTrust Manager is a high-availability appliance that centralizes encryption key management for Oracle Database and Microsoft SQL Server TDE as well as a variety of additional Thales and third-party encryption solutions. CipherTrust Manager helps direct key life-cycle tasks including generation, rotation, destruction, import and export as well as provide abilities to manage certificates and secrets.

The CipherTrust Manager is available as either hardware or a virtual appliance:

Either appliance is hardened to ensure protection of its contents and prevents any modification to the underlying operating system. In addition, non-HSM enabled appliances can utilize an external HSM as a root of trust for better security as well as utilize its random number generator for better key entropy.

CipherTrust key management solutions

Centrally managing your keys independently from your database applications will help you streamline operations, fulfill compliance mandates and better protect your sensitive data. And as your needs expand, Thales solutions can grow with you. Below is a summary of key management offerings and related products.

- TDE key management
- Database encryption, file encryption, comprehensive data encryption
- KMIP-compliant applications
- Tokenization
- Cloud key management (CipherTrust Cloud Key Manager)
- Key management through APIs
- Key storage

Learn more

Visit us at www.thalescpl.com to learn more about how Thales can help you simplify and strengthen key management for your enterprise database encryption solution.

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.

> cpl.thalesgroup.com <    

Americas – Arboretum Plaza II, 9442 Capital of Texas Highway North, Suite 100, Austin, TX 78759 USA • Tel: +1 888 343 5773 or +1 512 257 3900 • Fax: +1 954 888 6211 • E-mail: sales@thalessec.com
Asia Pacific – Thales Transport & Security (HK) Lt, Unit 4101-3, 41/F, Sunlight Tower, 248 Queen's Road East, Wanchai, Hong Kong • Tel: +852 2815 8633 • Fax: +852 2815 8141 • E-mail: apacsales.cpl@thalesgroup.com
Europe, Middle East, Africa – 350 Longwater Ave, Green Park, Reading, Berkshire, UK RG2 6GF • Tel: +44 (0)1844 201800 • Fax: +44 (0)1844 208550 • E-mail: emea.sales@thales-esecurity.com

Enterprise key management solutions



Challenges

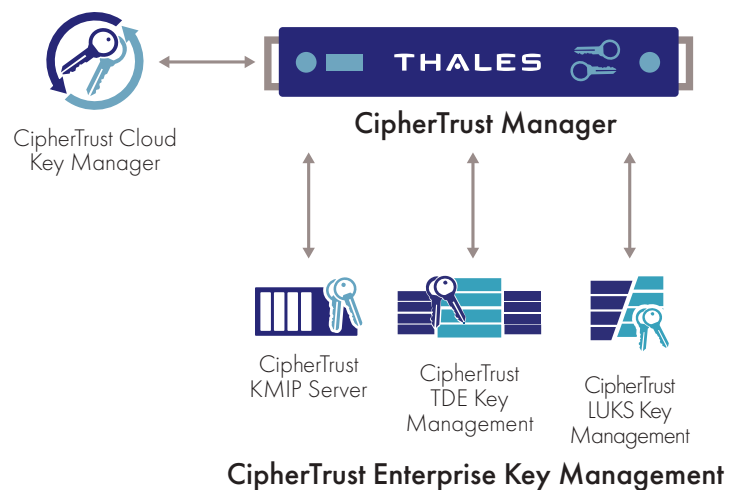
Managing encryption keys as a centralized service is essential to the integrity of any enterprise encryption system. Many organizations operate multiple, independent encryption systems resulting in silos, which complicates key management and erodes security. Managing keys for these silos consumes time, resources and budget. It also risks compromising data, if keys are kept local to the encryption systems they protect.

Solution: Thales key management solutions

Thales CipherTrust key management products centralize key management for internal as well as commercial off the shelf (COTS) applications. This gives customers greater command over your keys while increasing encryption data security. Thales CipherTrust key management products connect with your applications through standard interfaces and deliver access to robust key management and encryption functions.

Highly secure and efficient centralized key management

- Streamlined operations through centralized key management
- Improved compliance by separating keys from applications
- Comprehensive key services delivered by dedicated, FIPS-compliant hardware and software solutions



CipherTrust Manager: the foundation for Thales key management solutions

CipherTrust Manager is a high-availability appliance that centralizes encryption key management for the Thales Data Security Portfolio and third-party encryption solutions. CipherTrust Manager helps direct key life-cycle tasks including generation, rotation, destruction, import and export as well as provide abilities to manage certificates and secrets.

CipherTrust Manager enhances key management by delivering strong separation of duties for increased security. Additionally, CipherTrust Manager offers multi-tenancy, or domains, allowing customers and service providers to generate unique key management environments. This provides additional security and ultimate separation of duties, where no single administrator has access to all domains.

CipherTrust Manager is available as either hardware or a virtual appliance. Either appliance is hardened to ensure protection of its contents and prevents any modification to the underlying operating system. In addition, non-HSM enabled appliances can utilize an external HSM as a root of trust for better security as well as utilize its random number generator for better key entropy.

Thales key management solutions

Thales key management solutions support a variety of applications, including:

Key Management Interoperability Protocol (KMIP)

KMIP is an industry-standard protocol for encryption key exchange between clients (appliances and applications) and a server (key store). Standardization facilitates external key management for storage solutions including SAN and NAS storage arrays, self-encrypting drives and hyper-converged infrastructure solutions. KMIP simplifies the requirement of separating keys from the data being encrypted, and allowing those keys to be managed with a common set of policies from a corporate perspective. CipherTrust Manager has already been certified with a number of 3rd party applications utilizing KMIP (as well as other API's).

Database and Linux Key Management

Thales centralized key management solutions for databases and Linux can provide high security while providing enhanced IT efficiency. For both Transparent Database Encryption (TDE) key management and Linux Unified Key Setup (LUKS), a Thales agent on the database or Linux server requests keys from CipherTrust Manager and serves them to TDE or LUKS.

Cloud Key Management using CipherTrust Cloud Key Manager

CipherTrust Cloud Key Manager delivers efficient, secure control over the full lifecycle of software as a service (SaaS) and infrastructure as a service (IaaS) encryption keys, including key creation, uploading, updating, storing, revocation and reporting. It is available as a service in the cloud and as an on-premises deployment using the CipherTrust Manager to securely manage keys.

CipherTrust Cloud Key Manager supported clouds include Microsoft Azure and Office 365, Amazon Web Services, Salesforce.com and IBM Cloud.



CipherTrust Key Management Solutions for Proprietary Applications

For the most convenient integrations into applications that perform encryption but require centralized, key management, the CipherTrust Manager offers PKCS#11, REST, and NAE-XML API's that can be leveraged by any application environment supporting these API's. In addition, for the most performance-sensitive applications, CipherTrust Manager provides application-layer libraries implementing Java, C, C++, .NET and .NET CORE that can be installed on supported application server environments.

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.

The Importance of KMIP Standard for
Centralized Key Management

How Thales Trusted Cyber Technologies Can Help



Executive Summary

Today, protecting sensitive data is a fundamental requirement in virtually every business. What's also nearly universal is this: operations and security teams continue to struggle with managing the proliferation of encryption keys that enable in protection of sensitive data.. The Key Management Interoperability Protocol (KMIP) was developed to enable more unified, efficient management of keys from multiple encryption technologies and vendors. This white paper offers a look at the KMIP standard, and it shows how Thales solutions help you maximize the advantages of this standard.

The promise of KMIP

Organizations deploy specific encryption solutions based on their unique use cases, technologies, businesses, security threats and more. To address all their requirements, organizations employ many different solutions from different vendors. Historically these different technologies lacked common standards for key management.

KMIP changed that. KMIP advances interoperability standards for enterprise encryption key management in order to help organizations centrally manage keys from a number of vendors. KMIP represents a comprehensive protocol for communication between a client and a server. In most scenarios, customers will need to integrate a KMIP Client with a KMIP Server. The KMIP Server within the Key Management System (KMS) provides life-cycle management of keys sent by the KMIP Client, as well as perform cryptographic operations whose results can be returned to the client.

The KMIP standard is governed by the Organization for the Advancement of Structured Information Standards, or OASIS, a nonprofit organization that promotes the development, convergence and adoption of open standards. The KMIP standard has been developed with the support of top security experts and leading vendors, including Thales. In fact, Thales was one of the four pioneering vendors responsible for original development of the protocol, and the company's representatives have served in leadership capacities on the OASIS KMIP technical committee since the standard's inception in 2007.

How Thales enables customers to capitalize on the KMIP Standard

With its advanced solutions, deep expertise and extensive partnerships, Thales is uniquely equipped to help organizations quickly, effectively and broadly harness the advantages of the KMIP standard. Through its support of KMIP and other standards, Thales solutions enable more efficient, centralized management of the entire data protection ecosystem.

Thales has a track record of successful KMIP integrations and deployments, and continues to work with technology partners and customers to expand its ecosystem. Our solutions have been proven in organizations with the most stringent security requirements, including financial services institutions, health care organizations, retailers and government agencies.

Solution: CipherTrust Manager

With the CipherTrust Manager from Thales, organizations can maximize the advantages of the KMIP standard. In addition, CipherTrust Manager delivers centralized controls that enable consistent and repeatable management of encryption keys, access policies and improved security intelligence.

CipherTrust Manager makes it easy and efficient to manage data-at-rest security across the entire organization by offering a diverse portfolio of encryption solutions to protect data wherever it resides in the organization.. As a result, security teams can address data security policies, compliance mandates and best practices, while reducing administration effort and total cost of ownership. CipherTrust Manager can be deployed with a physical HSM Root of Trust. Optionally, the virtual appliance can use an external HSM. Both solutions provide improved key security and entropy.

The CipherTrust Manager is uniquely equipped to enable organizations to centralize and streamline key management across an organization. Here's why:

- CipherTrust Manager enables security teams to manage keys and policies for a broad range of data protection solutions from Thales, which feature broad capabilities for protecting and controlling access to databases, files and containers—and securing assets residing in cloud, virtual, big data and physical environments.
- CipherTrust Manager can manage keys and policies for a number of third-party platforms, including Transparent Data Encryption (TDE) products from Microsoft, Oracle and IBM.
- CipherTrust Manager offers comprehensive support for the KMIP standard. Effectively, any device or client software that is KMIP-enabled can communicate with CipherTrust Manager to facilitate management of encryption keys. Examples of KMIP clients include management systems in hyperconverged environments (Nutanix), self-encrypting drives in storage systems (Netapp, Dell, IBM, HPE) and native encryption in next-generation databases and virtualized environments.

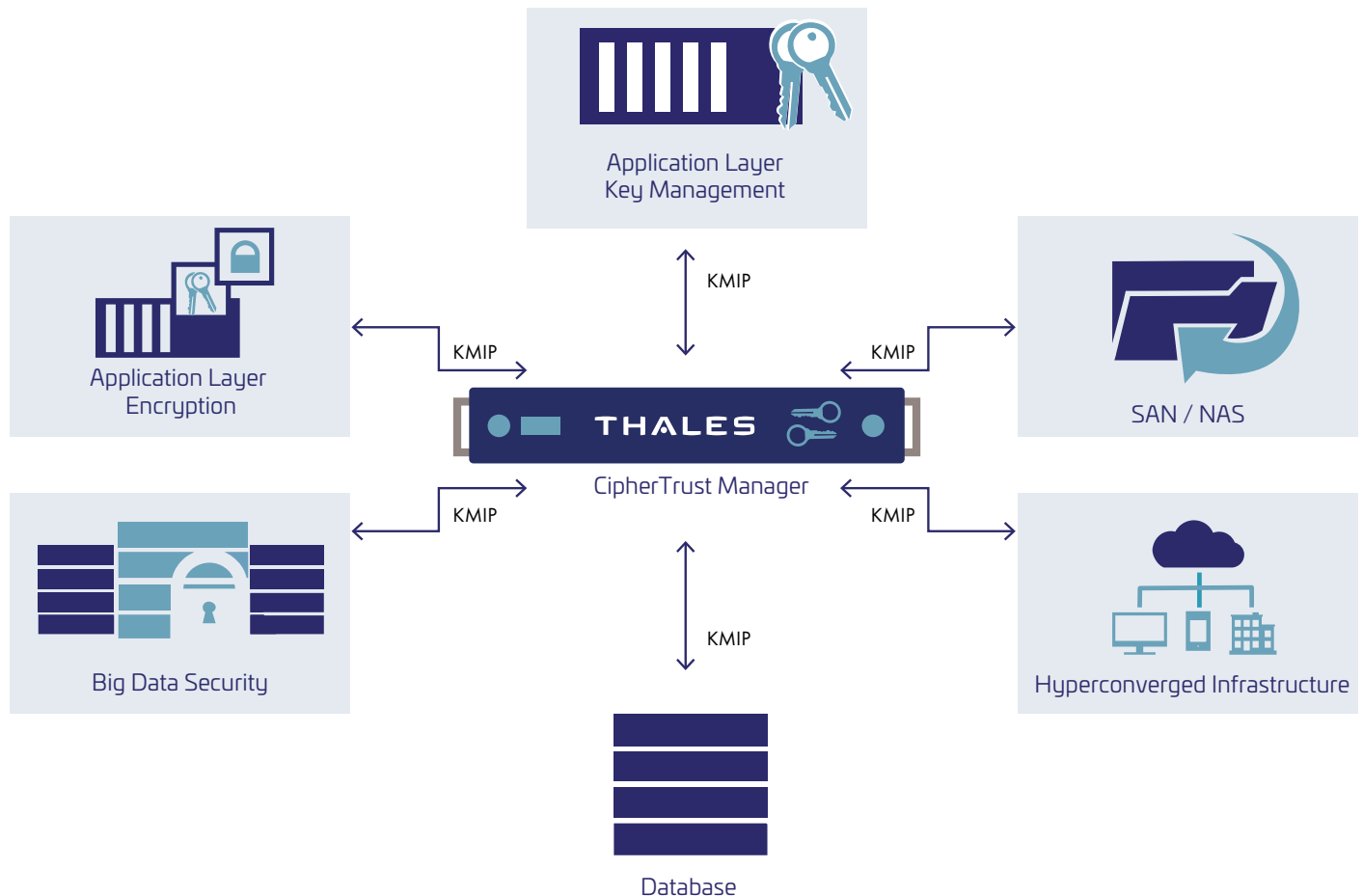


Figure 1: Thales CipherTrust Manager serves a variety of applications and client types using KMIP.

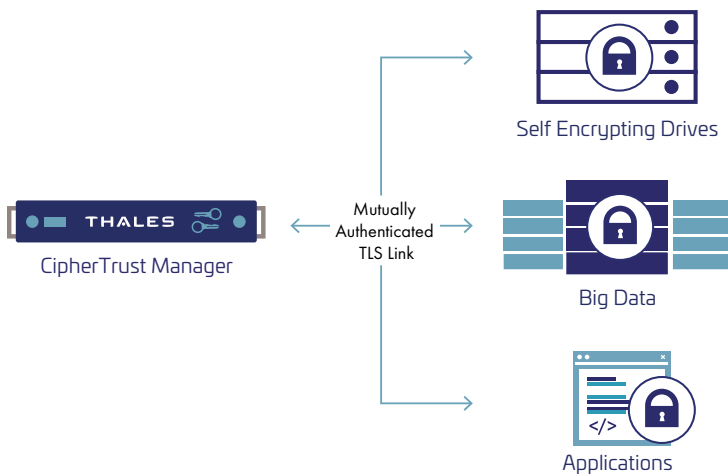


Figure 2. KMIP Client-Server Communications

1. Purchase and install KMIP Client licenses on CipherTrust Manager
2. Generate token to register each KMIP Client
3. Add token and restart KMIP service on CipherTrust Manager
4. Sign the KMIP Client Certificate Signing Request (CSR)
5. Upload CipherTrust Manager signed CSR to the KMIP Client

KMIP-based communications: How it works

Following is a high-level overview of how the process works:

- CipherTrust Manager needs to be prepared for KMIP support, which entails adding a KMIP client to the KMIP client list.
- A KMIP client is registered with the CipherTrust Manager
- Trust between the CipherTrust Manager and the KMIP client needs to be established. The KMIP protocol requires a mutually authenticated TLS connection between clients and servers. Once the client is registered with the CipherTrust Manager for secure communications, a certificate for the KMIP client will be created. This certificate will be used by the KMIP client for TLS authentication.

How Thales KMIP support benefits your organization

By leveraging Thales solutions, expertise and partnerships, organizations can realize a number of benefits:

- **Extend and maximize the value of your investments.** Thales support for KMIP allows organizations to more fully take advantage of their existing investments. CipherTrust Manager enables organizations to centrally manage the keys associated with their applications. As the encryption deployments grow, organizations can continue to employ CipherTrust Manager to manage keys for more technologies.
- **Streamline and enhance operations.** When organizations centralize key management on CipherTrust Manager, they can significantly simplify their operations. For example, administrators can centrally establish and enforce controls and policies for a number of systems. In addition, the solution enables centralized reporting for more effective and efficient compliance auditing, monitoring and reporting.
- **Enhance agility.** CipherTrust Manager helps organizations realize simplified operations and greater integration flexibility. As a result, the solution helps organizations adapt more quickly to evolving technologies and objectives. With the solution, organizations can more quickly and broadly leverage the advantages of innovative technologies like hyperconverged infrastructures and next-generation databases.

Technology support: featured KMIP partners and technologies

Today, many leading technology vendors offer encryption within their solutions, including hyperconverged infrastructures, databases, storage systems and more. This technology-centric encryption represents a vital line of defense. However, if keys are managed in a distributed, insecure fashion, it can erode or even eradicate any of the security benefits that can be gained through encryption.

That's why security experts and even the technology vendors themselves strongly recommend the use of purpose-built, centralized key management platforms, which is critically important for effectively safeguarding the keys generated by encryption. In fact, some vendors don't even allow the use of their encryption capabilities in production without the use of a third-party key management platform.

By leveraging CipherTrust Manager and the KMIP standard, customers can employ the encryption capabilities of their preferred technology vendors, while benefitting from a best-of-breed system for managing and securing keys. The CipherTrust Manager KMIP server enables seamless integration with a number of products across several categories. The sections below highlight some of the most important technologies that are currently supported.

Hyperconverged infrastructure

Hyperconverged infrastructures provide packaged, fully integrated technology stacks, including computing, storage, virtualization and networking. By leveraging these solutions, customers can significantly improve the efficiency of up-front deployment and ongoing administration, more quickly adapt to new business and technology requirements and more. The Thales KMIP server offers proven integration with a number of hyperconverged infrastructure technologies, including the following:

- **VMware vSAN and vSphere.** VMware vSAN powers leading hyperconverged infrastructure solutions. vSAN delivers secure, flash-optimized storage for workloads running on vSphere, VMware's virtualization platform. The vSAN solution delivers native encryption of flash devices. In addition vSphere offers encryption capabilities, securing the inputs and outputs from virtual machines before they get stored in disks. Customers can use CipherTrust Manager to manage keys generated by vSAN and vSphere encryption, enabling secure key management and strong role separation.

Storage

Enterprises rely on massive and rapidly growing storage infrastructures to manage data, and these repositories now invariably contain the organization's most sensitive, highly regulated assets. Encryption of these environments therefore represents a bedrock requirement. In response, many storage system vendors have added system-level encryption into their products. By leveraging CipherTrust Manager and the KMIP standard, customers can employ the encryption capabilities of their storage systems, while centralizing key management. Through its KMIP support, CipherTrust Manager enables seamless integration with a number of storage platforms, including the following:

- **NetApp ONTAP.** NetApp ONTAP enables administrators to assign, promote and retire storage resources without disrupting applications or business operations. ONTAP also unifies customers' control of their entire storage infrastructure, offering robust certificate-based authentication for access control and built-in encryption for data confidentiality. CipherTrust Manager and KMIP enables customers to consolidate key manager for ONTAP data encryption.

Next-generation data management

Database management systems very often house some of an organization's most sensitive data assets, and this is particularly true for next-generation database management platforms like MongoDB. Whether organizations are looking to comply with external regulations or internal policies, encryption of data at rest in these platforms is now an absolute mandate. Through its support for KMIP, CipherTrust Manager enables central, efficient and highly secure management of the keys generated by encryption in the following next-generation data management platforms:

- **MongoDB.** MongoDB is a NoSQL database that equips organizations with a number of significant advantages, including agility, scalability and high availability. In the MongoDB use case, the CipherTrust Manager provides high availability, standards-based enterprise encryption key management for MongoDB TDE using the KMIP standard, and offers vaulting and inventory of certificates. Consolidating enterprise encryption key management delivers consistent policy implementation between systems while reducing training and maintenance costs.

Conclusion

As enterprises expand the use of encryption to protect their sensitive data and assets, so grows the need to unify the management of encryption keys throughout their lifecycle. With the CipherTrust Manager from Thales, organizations can centrally manage keys and policies for Thales data protection solutions. Through its leading KMIP support, CipherTrust Manager can manage keys associated with encryption technologies in hyperconverged infrastructures, storage systems and next-generation data management platforms. As a result, organizations can address their most urgent security objectives, while significantly improving the efficiency of their operations.

About Thales Trusted Cyber Technologies

Thales Trusted Cyber Technologies, a business area of Thales Defense & Security, Inc., protects the most vital data from the core to the cloud to the field. We serve as a trusted, U.S. based source for cyber security solutions for the U.S. Federal Government. Our solutions enable agencies to deploy a holistic data protection ecosystem where data and cryptographic keys are secured and managed, and access and distribution are controlled. For more information, visit www.thalestct.com

THALES

CipherTrust Manager

KMIP REFERENCE GUIDE



Key Management Interoperability Protocol

The Key Management Interoperability Protocol (KMIP) is a communication protocol that facilitates key management and cryptographic operations on a key management server. This section provides additional user information and specifications applicable to the KMIP interface of CipherTrust Manager.

Setting up a KMIP Client

Here are the prerequisites for allowing KMIP clients to connect to CipherTrust Manager:

- > The CipherTrust Manager server's KMIP port 5696 should be open from the client.
- > The CipherTrust Manager server supports the following TLS versions:
 - TLS version 1.0
 - TLS version 1.1
 - TLS version 1.2 (default)

KMIP Parameter: `kmip_enable_hard_delete`

This optional parameter enables a hard delete of keys on a "KMIP Destroy" operation. This means that both meta-data and material are removed from CipherTrust Manager for the key being deleted. By default, only key material is removed and meta-data is preserved with the updated key state. This setting applies only to the KMIP interface. Should be set to '1' for enabling this feature or '0' for returning to the default behavior.

KMIP Licensing

The KMIP interface is active for a 90-day trial period after CipherTrust Manager is installed. After the trial period expires, a new purchased KMIP license must be installed to use the KMIP interface. Contact Gemalto Customer Support for assistance in obtaining a license.

To activate a license using the GUI

To install a KMIP license:

1. Log on to the CipherTrust Manager console as administrator.
2. Click **Admin Settings** to open the application.
3. Click **Licensing**, if needed. The **Server Licenses** section is displayed on the right.
4. Copy the **Connector Lock Code**. Every CipherTrust Manager appliance has a unique connector lock code. This lock code is needed to activate a license for the CipherTrust Manager.

NOTE In a CipherTrust Manager cluster, the connector lock code is applicable to all nodes of the cluster to enforce connector licenses across the cluster nodes. When a license is activated for one CipherTrust Manager appliance, the license is replicated on all the nodes of the cluster.

5. Activate the license.

When you purchase a CipherTrust Manager and connectors, you receive an email after the Purchase Order (PO) is processed. Accompanying this email is a PDF attachment that contains the Entitlement ID (EID). This ID is needed to activate your license.

To activate the KMIP license:

- a. Log on to the [Gemalto License Portal](#). Use your EID when logging on for the first time.
- b. Register your account, if not already done. After registration, you can find the details about your EID and your available licenses.
- c. On the **Entitlements** tab, select the KMIP license.
- d. Click **Activate**.
- e. In the **Quantity** field, specify the number of licenses to activate.
- f. In the **License Lock Code** field, paste the [Connector Lock Code](#) copied earlier.
- g. Click **Activate**.

NOTE

- > Based on the requirements, a KMIP license is activated and a license string is generated.
- > The license string generated for one CipherTrust Manager cannot be used on any other CipherTrust Manager.

- h. Copy the **License String**. This license string is needed to install the KMIP license on the CipherTrust Manager.
6. Return to the CipherTrust Manager console.
7. Under the **Add A New License** section, in the **License String** field, paste the copied [License String](#).
8. Click **Add**. The license is now installed. The license appears under the **Installed Licenses** section.

KMIP Client Registration

CipherTrust Manager recognizes only registered KMIP clients.

For registering a KMIP client, a certificate is required. The certificate facilitates authentication and communication between CipherTrust Manager and the client.

You can register KMIP clients using following methods:

1. Using Auto Registration (using an existing certificate)

Auto registration is useful when the clients are having certificates from the Classic KeySecure, or from older versions of CipherTrust Manager, or certificates issued by External CAs.

If you are already having one of the above certificate, you can utilize it to register your KMIP client, or to upgrade to a newer version of CipherTrust Manager. For detailed steps, see ["KMIP Client Registration for existing CipherTrust Manager and external CA\(s\)" below](#).

2. Generating a new certificate

You can create a new certificate using CipherTrust Manager, and use it to register a new KMIP client. For detailed steps, see ["To register a new KMIP Client" on page 6](#).

Specifying LDAP and Multi-Domain Client Usernames in KMIP Certificates

The connection and/or username can be specified in following formats:

> <username> - A local user in the root domain. For example, **joe**.

> <ldap-connection>|<username>

An LDAP user in the root domain. For example, **ldap-connection|joe**.

> <domain-id>|<ldap-connection>|<username>

An LDAP user in a specific domain. For example, **domain-id|ldap-connection|joe**.

> <domain-id>||<username>

A local user in a specific domain. For example, **domain-id||joe**.

When browsing the GUI, users can be identified from the source property. The user is displayed in either of the formats listed above. A local user is displayed as **local|<username>**.

KMIP Client Registration for existing CipherTrust Manager and external CA(s)

1. Create a registration token to register KMIP clients.

Request:

```
ksctl clientmgmt tokens create --configfile config_example.yaml
```

Response:

```
{
  "id": "dd8289f4-adbe-412f-9191-320bf2bac847",
  "uri": "kylo:kylo:munshi:tokens:dd8289f4-adbe-412f-9191-320bf2bac847",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2019-12-19T05:52:36.590956451Z",
  "updatedAt": "2019-12-19T05:52:36.590956451Z",
}
```

```

    "token": "RwIgd6RfXd7kzri12ydzoefXsA2R9ju2YDxMv2xM9duQbqAhcKcyMQO2OpqZesti",
    "valid_until": "0001-01-01T00:00:00Z",
    "max_clients": -1,
    "cert_duration": 730,
    "clients_registered": 0,
    "ca_id": "713172e0-02be-48c2-be60-6aae964ce943",
    "label": null
}

```

2. Update the KMIP Interface for which you want to allow Auto registration of KMIP clients. For auto registration, the client certificate which is to be registered should be signed by the trusted CA for the interface. To update the KMIP interface, run the `ksctl interfaces modify` command with the following parameters.

- `-a` (true | false) is for enabling or disabling auto registration
- `-r` is for providing registration token as input

If `-a` is set to true, then providing registration token in `-r` is mandatory

NOTE A new KMIP interface can also be created with the `-a` and `-r` parameters.

NOTE If the client is auto registered using the **tls-pw-opt** mode, then that certificate cannot be used to connect via the other two modes, **tls-cert-pw-opt** and **tls-cert-and-pw**.

Request to update interface with auto registration on:

```

ksctl interfaces modify -n kmip -a true -r
RwIgd6RfXd7kzri12ydzoefXsA2R9ju2YDxMv2xM9duQbqAhcKcyMQO2OpqZesti --configfile config_
example.yaml

```

Response:

```

{
  "id": "54523325-62d0-40d4-a45e-f45b9e1679e7",
  "name": "kmip",
  "mode": "tls-cert-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:713172e0-02be-48c2-be60-6aae964ce943",
  "trusted_cas":
    {
      "local": ["kylo:kylo:naboo:localca:713172e0-02be-48c2-be60-6aae964ce943"]
    },
  "createdAt": "2019-12-13T03:27:28.767622Z",
  "updatedAt": "2019-12-19T05:55:26.362558Z",
  "default_connection": "local_account",
  "port": 5696,
  "network_interface": "all",
  "interface_type": "kmip",
  "minimum_tls_version": "tls_1_2",
  "local_auto_gen_attributes":
    {
      "cn": "kmip.keysecure.local",
      "email_addresses": [
        "support@gemalto.com"
      ],
      "names": [
        {
          "C": "US",

```

```

    "ST": "MD",
    "L": "Belcamp",
    "O": "Gemalto",
    "OU": ""
  }
],
  "generated": false
},
"auto_registration": true,
"registration_token": "RwIgd6RfXd7kzril2ydzoefXsA2R9ju2YDxMv2xM9duQbqAhcKcyMQO2OpqZesti"
}

```

NOTE `cert_user_field` specifies how the user name is extracted from the client certificate. The default value is `CN`. This user must already exist. This parameter must be one of the following:

- **CN**: common name
- **OU**: organizational unit
- **SN**: surname
- **E**: Email address
- **UID**: User ID
- **E_ND**: Email without domain; the data to the left of the @ sign in the email address is taken as the user name

3. Connect to the KMIP interface using the client certificate you want to register. It will automatically register the KMIP client corresponding to that certificate.

NOTE If the client is mistakenly registered without a username, then you need to manually delete the registered client, before creating a new client. To delete a registered client, log into the console, navigate to **KMIP > Registered Clients**, and delete the client.

To register a new KMIP Client

1. Create a json file for specifying properties to create a KMIP Client Profile. A KMIP client profile contains a set of properties, on basis of which, the KMIP client gets registered. There are two ways to create a profile:

- Using `csr`
- Using `csr` parameter

A detailed explanation of input parameters is provided after the examples.

profile.json: This file is a user input to CLI for providing profile parameters.

NOTE `profile.json` is just a place holder for name of file, user can choose any name for this file.

Example 1:

```

{
  "cert_user_field": "CN",
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIIBCDChXGc6JazA==\n-----END\nCERTIFICATE REQUEST-----\n"
}

```



```
}
```

Example 2:

```
{
  "cert_user_field": "CN",
  "csr_cn": "user1",
  "csr_org_name": "Thales",
  "csr_org_unit": "Security",
  "csr_email": "abc@example.com",
  "csr_city": "Noida",
  "csr_state": "UP",
  "csr_country": "IN"
}
```

Parameter Explanation

- **cert_user_field**: Specifies how the user name is extracted from the client certificate. The default value is CN. This user must already exist. For more information, click [here](#).

NOTE After registering the new KMIP client, `cert_user_field` setting of profile is used instead of the setting present in the interface.

- **csr**: This is certificate signing request. The common name specified in CSR will be used to authenticate KMIP request.

NOTE If the user is not present, then profile registration will fail.

Other parameters

If CSR is not provided, then CipherTrust Manager creates CSR using below input parameters from `profile.json` :

- **csr_cn**: Common Name used to create CSR.
 - **csr_org_name**: Organization name to be used in CSR.
 - **csr_uid**: User ID used to create CSR.
 - **csr_org_unit**: Organizational unit to be used in CSR.
 - **csr_email**: Email address to be used in CSR.
 - **csr_city**: City name to be used in CSR.
 - **csr_state**: State name to be used in CSR.
 - **csr_country**: Country name to be used in CSR.
2. **(Optional Step)** Create a json file (for example, *device_cred.json*) for specifying device credential properties.

NOTE This step is optional and should be performed only when it is desired to authenticate the KMIP Client using device credentials.

Example 1:

All fields in the example are optional, however, at least one field must be specified for a successful authentication using device credentials.

```
{
  "serial_no": "12345",
  "password": "example_password_123",
  "device_id": "example_device_ID",
  "network_id": "sample_Network_ID",
  "machine_id": "MAC123456789",
  "media_id": "Media123456789"
}
```

Parameters

- **serial_no**: Serial number of the device, for example the hardware serial number of the device.
 - **password**: Password for the device.
 - **device_id**: Device Identifier of the device.
 - **network_id**: Network identifier of the device, for example the MAC address for Ethernet connected devices.
 - **machine_id**: Machine identifier of the device, for example the client aggregator identifier, such as a tape library aggregating tape drives.
 - **media_id**: Media identifier of the device, for example the volume identifier used for a tape cartridge.
3. Create the KMIP profile using the json file created in [step 1](#).

Request:

```
ksctl kmip createprofile -p kmipprofile1 -c profile.json --configfile config_example.yaml
```

NOTE If using [device credentials](#), use the following request instead:

```
ksctl kmip createprofile -p kmipprofile1 -c profile.json -d device_cred.json
--configfile config_example.yaml
Here device_cred.json is the device credential file created in previous step.
```

Response:

```
{
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "createdAt": "2019-12-18T10:19:32.745104774Z",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "id": "70d92635-7e9f-4a9b-9161-d7f41b4e0109",
  "name": "kmipprofile1",
  "properties": {
    "csr_cn": "rajesh"
  },
  "updatedAt": "2019-12-18T10:19:32.745104774Z",
  "uri": "kylo:kylo:nae:kmip_profile:kmipprofile1"
}
```

4. Create a registration token using the profile created in the previous step.

Request:

```
ksctl kmip createkmipregtoken -p kmipprofile1 --configfile config_example.yaml
```

NOTE `profile_name` is required and must be present in the system before creating a registration token.

Response:

```
{
  "id": "6b8f5397-b738-4160-8c39-7979bae280aa",
  "uri": "kylo:kylo:munshi:tokens:6b8f5397-b738-4160-8c39-7979bae280aa",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2019-12-18T10:29:26.701363932Z",
  "updatedAt": "2019-12-18T10:29:26.701363932Z",
  "token": "LW0heyZNXHE9sEKmu7B0zVqnwHns42tahMVWJYUkP6I5CI67RQLS0EDkZ96UiocI",
  "valid_until": "0001-01-01T00:00:00Z",
  "max_clients": -1,
  "cert_duration": 730,
  "clients_registered": 0,
  "ca_id": "efeld302-7ff4-46ca-a2a4-c4d9236cc0f8",
  "label": {
    "KmiclientProfile": "kmipprofile1"
  }
}
```

5. Register the new KMIP client using the registration token generated in the previous step.

Request:

```
ksctl kmip register -n Kmiclient2 -t
LW0heyZNXHE9sEKmu7B0zVqnwHns42tahMVWJYUkP6I5CI67RQLS0EDkZ96UiocI --kmipCertOutFile
ClientCert.pem --kmipKeyOutFile ClientKey.pem --configfile config_example.yaml
```

Response:

```
{
  "cert": "-----BEGIN CERTIFICATE-----4xq+8sZ6PuwxePopPD2HwB+uxjLjdaKb01wz5AEk...-
N3e2ajOow/ZC2FxtkipCIyYdlTJ7P\n-----END CERTIFICATE-----\n",
  "client_ca": "-----BEGIN CERTIFICATE-----
\kXLCgQWnp2Kp445v2ZAYf3CUwWnBBUzLZdpDkUkQpaloJaMkNXU2KFVxjyDT7cXM...
v5jwHaYfqXImDs0bC/UZyEo+bnwFlA==\n-----END CERTIFICATE-----\n",
  "client_id": "ac8c7131-7dc4-4b1e-ab42-3529b4c4a497",
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nwSnPtpvJZU9kac0...2LfMJxmbnmAug2YM/\n-----END
CERTIFICATE REQUEST-----\n",
  "key": "-----BEGIN RSA PRIVATE KEY-----
zMtUrU1Uu9S9dc89ajkKJ...TlkMA/Gx0IOkpH1zMbJL6NEgaOZbC1tCLjxB6YjQIgHfrTqupc=\n-----END RSA
PRIVATE KEY-----\n"
}
```

6. Use Client Certificate and Client Private Key to connect with the server.

Supported Objects, Operations, and Attributes

The following KMIP Objects Types, Attributes, and Operations supported on CipherTrust Manager are in accordance with KMIP Specification Version 1.4.

Supported Object Types

Symmetric Key	Public Key	Private Key
Secret Data	Opaque	

Supported Operations

Activate	Delete Attribute	Get Attributes List	Register
Add Attribute	Destroy	Locate	Re-key
Check	Discover Versions	MAC	Re-key Key Pair
Create	Encrypt	MACV	Revoke
Create Key Pair	Get	Modify Attribute	
Decrypt	Get Attributes	Query	

NOTE CipherTrust Manager supports Wrapping and Unwrapping of Keys for Get and Register operations. To read more, go to ["Wrap/Unwrap support in KMIP" on the next page](#)

Supported Attributes

Unique Identifier	Name	Object Type
Cryptographic Algorithm	Cryptographic Length	Cryptographic Parameters
Cryptographic Domain Parameters	Cryptographic Usage Mask	Usage Limits
State	Initial Date	Activation Date
Process Start Date	Protect Stop Date	Deactivation Date
Destroy Date	Compromise Occurrence Date	Compromise Date
Revocation Reason	Link	Application Specific Information
Contact Information	Last Change Date	Custom Attribute
Fresh Attribute	Alternate Name	Digest Attribute

NOTE For information on the above listed object Types, Operations, and Attributes, refer to the [OASIS KMIP Documentation](#).

NOTE Fresh Attribute, Custom Attribute, Application Specific Information, and Digest Attribute are stored in the Key Meta.

Example 1:

If a custom attribute (for example, `AttributeName : x-attribute1`, `AttributeValue: value1`) is added to the key `k1`, then the attribute is stored in Key Meta as:

```
{
  "kmip": {
    "custom": [
      {
        "type": "TextString",
        "x-attribute1": "Value1"
      }
    ],
    "ownerId": "local|261d8e87-73bb-4106-aa3d-06ed92e2f010",
    "permissions": {}
  }
}
```

Example 2:

If an application specific attribute (for example, `application_data": "data"`, `application_namespace": "Test"`) is added to a key, then the attribute is stored in Key Meta as:

```
{
  "kmip": {
    "app_specific_info": {
      "application_data": "data",
      "application_namespace": "TestASi"
    },
    "ownerId": "local|261d8e87-73bb-4106-aa3d-06ed92e2f010",
    "permissions": {}
  }
}
```

Wrap/Unwrap support in KMIP

Key wrapping is used for encrypting data keys. This provides an additional layer of security in protecting keys when they are at rest or are being transmitted over insecure networks. A variety of key wrapping algorithms can be used to encrypt the data keys. Similarly, unwrap mechanisms are used to decrypting the data keys.

CipherTrust Manager supports wrapping/unwrapping of keys under **Get** and **Register** operations.

NOTE Only *Symmetric* and *Private* keys can be wrapped/unwrapped in KMIP

Supported Wrapping Methods

- > Encrypt
- > MACSign

NOTE

- > For Encrypt Wrapping Method:
 - Only AES and RSA keys can be used to wrap the symmetric keys.
 - Only AES-256 keys can be used to wrap private keys.
- > For MACSign Wrapping Method:
 - Only HMAC keys can be used for MAC request.
 - Only RSA keys can be used for Sign request.

Encrypt Wrapping Method		
Symmetric Data Keys		Private Data Keys
Supported BlockCipher Modes for AES wrap keys	Supported padding methods for RSA wrap keys	Supported BlockCipher Mode for AES wrap keys
> NISTKeyWrap (RFC-3394) > AESKeyWrapPadding (RFC-5649)	> PKCS1V1_5 > OAEP Supported Hash Algorithms for OAEP padding method: • SHA_1 • SHA_256 • SHA_384 • SHA_512	> AESKeyWrapPadding (RFC-5649)

MACSign Wrapping Method for Symmetric and Private Keys		
MAC Request	Sign Request	
Supported HMAC algorithms for HMAC wrap keys	Supported padding methods for RSA wrap keys	Supported DigitalSignatureAlgorithms for RSA wrap keys
> HMAC_SHA_1 > HMAC_SHA_256 > HMAC_SHA_384 > HMAC_SHA_512	> PKCS1V1_5 (Default) > PSS Supported Hash Algorithms for above padding methods: • SHA_1 • SHA_256 • SHA_384 • SHA_512	> SHA_1WithRSAEncryption > SHA_256WithRSAEncryption > SHA_384WithRSAEncryption > SHA_512WithRSAEncryption

Supported Encodings

- > NoEncoding
- > TTLVEncoding

For more information, visit [OASIS Key Management Interoperability Protocol Documentation](#).

Cryptographic Operations for KMIP

Using KMIP interface, you can perform following cryptographic operations:

Operations	Algorithms	Modes	Padding
Encrypt/Decrypt	AES	> ECB (default) > CBC > GCM	> PKCS5 (default) > None
	TDES	> ECB (default) > CBC	> PKCS5 (default) > None
	RSA	not applicable	> PKCS1V1_5 (default) > OAEP
MAC/MACV	HMAC_SHA1	not applicable	not applicable
	HMAC_SHA256	not applicable	not applicable
	HMAC_SHA384	not applicable	not applicable
	HMAC_SHA512	not applicable	not applicable

KMIP Interface Mode

The "interface mode" configuration parameter specifies the KMIP Interface mode, and must be one of the following:

- > **tls-pw-opt**: TLS, allow anonymous logins.
 - Username is not extracted from the client certificate.
 - If authentication request is sent in KMIP request, Username will be the owner of the key.
 - If authentication request is not sent in KMIP request, the Key Owner will be the 'global' user.
- > **tls-cert-pw-opt**: TLS, user name taken from client cert, auth request is optional.
 - Username is extracted from the registered client's Common Name specified during client's registration.
 - If authentication request is sent in KMIP request, then the Username in the authentication request will be the owner of the key.
- > **tls-cert-and-pw**: TLS, password is needed, user name specified during client's registration must match user name in authentication request.
 - Username is extracted from the registered client's Common Name specified during client's registration.
 - The inclusion of the authentication request in KMIP request is mandatory.

- The Username in the authentication request must match with the registered client's Common Name specified during client's registration.

TIP The interface mode can be changed using the API or the CLI "ksctl interfaces modify" command.

KMIP Interface mode mapping

Following table provides a mapping between the KMIP interfaces mode from SafeNet KeySecure Classic and CipherTrust Manager.

Safenet KeySecure Classic	CipherTrust Manager
Password authentication: Not Used Client cert authentication: SSL only	tls-pw-opt TLS, allow anonymous logins, verify client cert
Password authentication: Optional Client cert authentication: SSL only	
Password authentication: Required Client cert authentication: SSL only	tls-pw-req
Password authentication: Not Used Client cert authentication: SSL with CN	tls-cert-pw-opt Verify client cert, user name taken from client cert, auth request is optional
Password authentication: Optional Client cert authentication: SSL with CN	
Password authentication: Required Client cert authentication: SSL with CN	tls-cert-and-pw Verify client cert, password is needed, user name in cert must match user name in authentication request

Managing KMIP Interface

You can add, remove, or modify KMIP interfaces.

To create a new KMIP interface:

Example:

```
$ ksctl interfaces create -o 5697 -y kmip
```

Response:

```
{
  "id": "90b3b131-d6d8-4985-abdd-539162c136c3",
  "name": "kmip_all_5697",
  "mode": "tls-cert-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:c729ffe0-f6ad-49ad-8558-2db435b112c7",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:c729ffe0-f6ad-49ad-8558-2db435b112c7"
    ]
  }
}
```

```

    ],
    "createdAt": "2019-08-22T10:10:28.05794Z",
    "updatedAt": "2019-08-22T10:10:28.05794Z",
    "default_connection": "local_account",
    "port": 5697,
    "network_interface": "all",
    "interface_type": "kmip"
}

```

NOTE Interface can only be created in Root domain. To use interfaces in other domains, add the domain certificate to the Local Trusted CA of the domain.

To modify TLS version of an interface:

```
$ ksctl interfaces modify -n <interface name> -e <minimum tls version>
```

Here:

- <interface name> is KMIP interface name.
- <minimum tls version> can be `tls_1_0`, `tls_1_1`, or `tls_1_2`.

Example:

```
$ ksctl interfaces modify -n kmip_all_5697 -e tls_1_0
```

Response:

```

{
  "id": "6199c8d1-f59a-4bd1-8dfd-cf4d43900264",
  "name": "kmip_all_5697",
  "mode": "tls-cert-and-pw",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:afdebde8-a5ff-4cbd-9bb2-6dc5186e4c03",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:0a4bf97b-1294-41d2-93cd-d19e6290cfa3",
      "kylo:kylo:naboo:localca:b7860db1-b7d1-4163-996e-22b14090d55a",
      "kylo:kylo:naboo:localca:afdebde8-a5ff-4cbd-9bb2-6dc5186e4c03"
    ]
  },
  "createdAt": "2019-08-28T05:21:05.36768Z",
  "updatedAt": "2019-08-28T07:41:38.62511Z",
  "default_connection": "local_account",
  "port": 5697,
  "network_interface": "all",
  "interface_type": "kmip",
  "kmip_enable_hard_delete": 0,
  "minimum_tls_version": "tls_1_0"
}

```

Viewing and setting log level

You can view or change the log level of KMIP service using the "ksctl" CLI tool.

There are two subcommands in the `logs` command:

- > `getlevel`: used for viewing the current log level of a service
- > `setlevel`: used for setting log level of a service

Using `getlevel`

To view the current log level for KMIP service, use the following request in CLI:

Request:

```
ksctl logs getlevel --service kmip
```

Response:

```
{
  "level": "INF",
  "service": "kmip"
}
```

Using `setlevel`

To set current log level to debug for KMIP service, use the following request in CLI:

Request:

```
ksctl logs setlevel --service kmip --level debug
```

Response:

```
{
  "level": "DBG",
  "service": "kmip"
}
```

Supported levels are:

- > `error` or `ERR`
- > `info` or `INF`
- > `debug` or `DBG`

NOTE

- > To view the logs, go to `/opt/keysecure/logs` directory on appliance.
- > To download the logs, use `ksctl logs download` command.
- > Only users with administrator privileges can change or set the log level.
- > When log level is set to debug, sensitive data from debug logs is masked.

Support Contacts

If you encounter a problem while installing, registering, or operating this product, please refer to the documentation before contacting support. If you cannot resolve the issue, contact your supplier or Thales Customer Support.

Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at <https://supportportal.thalesgroup.com>, is where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable database of support resources, including:

- > Software and firmware downloads
- > Latest product documentation
- > Latest release notes listing known problems and workarounds
- > A knowledge base
- > FAQs
- > Technical notes, and more

You can also use the portal to create and manage support cases.

NOTE You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

Telephone Support

If you have an urgent problem, or cannot access the Customer Support Portal, you can contact Thales Customer Support by telephone at [+1 410-931-7520](tel:+14109317520). Additional local telephone support numbers are listed on the support portal.

Email Support

You can also contact technical support by email at technical.support.dis@thalesgroup.com.

Troubleshooting Issues in Conjunction with Customer Support

When making a support call, the CipherTrust Manager System Administrator (ksadmin) may be asked to log in using ssh to retrieve CipherTrust Manager Logs. CipherTrust Manager Logs are located in this directory:

`/opt/keysecure/logs`

NOTE Any ssl connection related error message can be filtered from the log file based on “ERR” and “tls” tags.



CipherTrust Integrations

4 February 2026

Contents

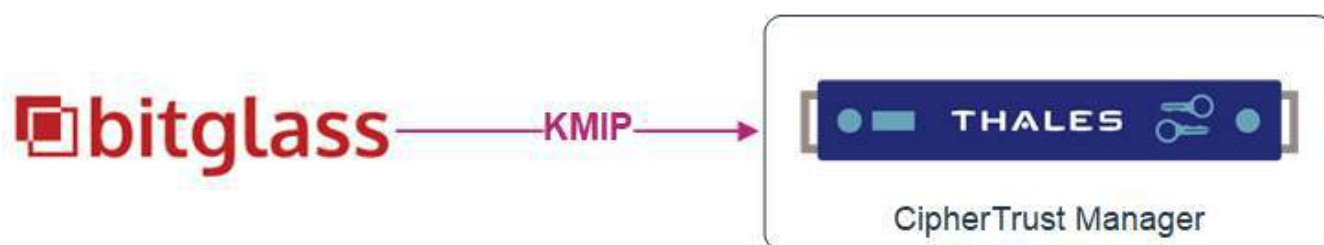
- Bitglass 3
 - Integration with CipherTrust Manager 3
 - Verifying Your Integration 12
 - Appendix. 13

Bitglass

Bitglass enterprise edition allows users to encrypt data at rest in cloud applications (for example, Salesforce, Box, and so on). Keys can be managed in the CipherTrust Manager or the users can add their own KMIP key store to utilize existing key management appliance/servers.

Data is encrypted using 256-bit derived keys which are encrypted by master keys stored in Key Vaults. Master Key is AES 256-bit symmetric key.

If a KMIP CipherTrust Manager becomes inaccessible, Bitglass cannot decrypt any data encrypted by master keys stored in the CipherTrust Manager. It is critical that external KMS infrastructure is deployed in a redundant high-availability architecture.



Supported Product Versions

Note

This integration has been validated in the field by a partner or in a customer environment with the following software versions. It is recommended to test the integration in a non-production environment with desired software versions before deploying it to production. Thales will provide best-effort support.

CipherTrust Manager

- CipherTrust Manager 2.0

Bitglass

- Bitglass

Integration with CipherTrust Manager

This section lists the steps to integrate Bitglass with CipherTrust Manager.

- [Prerequisites](#)
- [Configuration on CipherTrust Manager](#)
- [Configuration on Bitglass](#)

Prerequisites

This section provides the prerequisites for integration of Bitglass with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager Documentation](#).
- Bitglass communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the Bitglass system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the Bitglass cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.
- Ensure that Bitglass has proper license for data at Rest Encryption. To check the license information in Bitglass, navigate to **Settings > Software and Licenses > License Information**.
- Ensure that the required licenses are activated on CipherTrust Manager. For more details, refer to the [CipherTrust Manager documentation](#).

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a User](#)
2. [Assigning User to a Group](#)
3. [Creating or Adding a CA \(optional\)](#)
4. [Registering a KMIP Client](#)
5. [Configuring the KMIP Interface](#)

Creating a User

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Access Management** application.
3. On the left navigation pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Add User**.
5. On the **Add User** page, provide the following details:

-
- Select the required **Connection Type**
 - Select the checkbox against the required **Allowed Client Type**.
 - Enter **Full Name**
 - Enter **Username**
 - Enter the desired **Password**. Re-enter the password in the **Password Match** field.

6. Click **Add User**. The newly created user will now appear on the **Users** page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Navigate to the **Users** page.
2. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
3. Click **Edit/View**.
4. Click **Group Memberships > Add Group**.
5. In the search bar, enter the desired Group name and select the check box corresponding to it. For example, **Key Admins** or **Key Users**.
6. Click **Add Group**.

Creating or Adding a CA (Optional)

Note

Creating a CA is an optional step in this integration. However, it is recommended to select the CA carefully, as it will be used throughout the process.

Select the **Local CA** tab if you want to create a Local CA **OR** select the **External CA** tab if you wish to add the External CA.

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under **Pending CAs** section.

-
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
 4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. The **External Certificate Authorities** page appears. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate.

4. Click **Add External CA**.

Registering a KMIP Client

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Navigate to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Click **Save**. The new profile is created.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.

-
- d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created and uploaded on the CipherTrust Manager.

5. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **View/Edit**.
 - d. On the **Interface Detail** window, perform the following:
 - Select the Interface Mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - Select **Auto Registration** checkbox.
 - Paste the **Registration Token**.
 - Add the CA (Local or External) to the list of Trusted CAs. The CA will reflect under the dropdown menu of Local/External Trusted CAs depending upon CA that you created [above](#).
 - Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:

-
- a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. You can either paste the content of a generated `client.csr` or you can create one, by filling in the required details.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on the CA type that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.
 - Specify client name and paste the **Registration Token** that you generated in the above step.

Note

If you are using external CA then you can either upload the file using **Upload Certificate** option or select the **Text** option to paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **Client Certificate**.

Configuring the KMIP Interface

The KMIP interface can be configured through:

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the **Auto Registration** is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

Local CA for Automatic Server Certificate Generation should be set to **Turn off auto generation from Local CA** in case of External CA.

6. Depending upon the CA that you created [above](#), select an option from the following:
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
 - If you are using **External CA** then select the CA under **External Trusted CAs** provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Further, you need to configure the following:

1. Creating Key (using API Playground)
2. Update the Key Attributes

Creating Key (using API Playground)

1. At the top right corner of the main toolbar, click **API** to go to the API playground.
2. Click **Authenticate** on the top right corner. Specify username and password, then click **Post**.

Note

The session will be valid for 300 seconds. You can also authenticate later if you are unable to perform the following steps within this time frame.

3. Go to **Keys** section in the left sidebar. Click **Create - Post**.
4. Specify the following content in the body, and click **POST**. Specify a value equal to or less than 60 characters.

```
{
  "idSize": 60
}
```

Here **idSize:60** will be set to the **ID** of the key, which will be further used while configuring the policy file of the Bitglass.

Caution

Specifying a value greater than 60 leads to integration failure. To view the list of possible errors, go to the [Troubleshooting](#) section.

Updating the Key Attributes

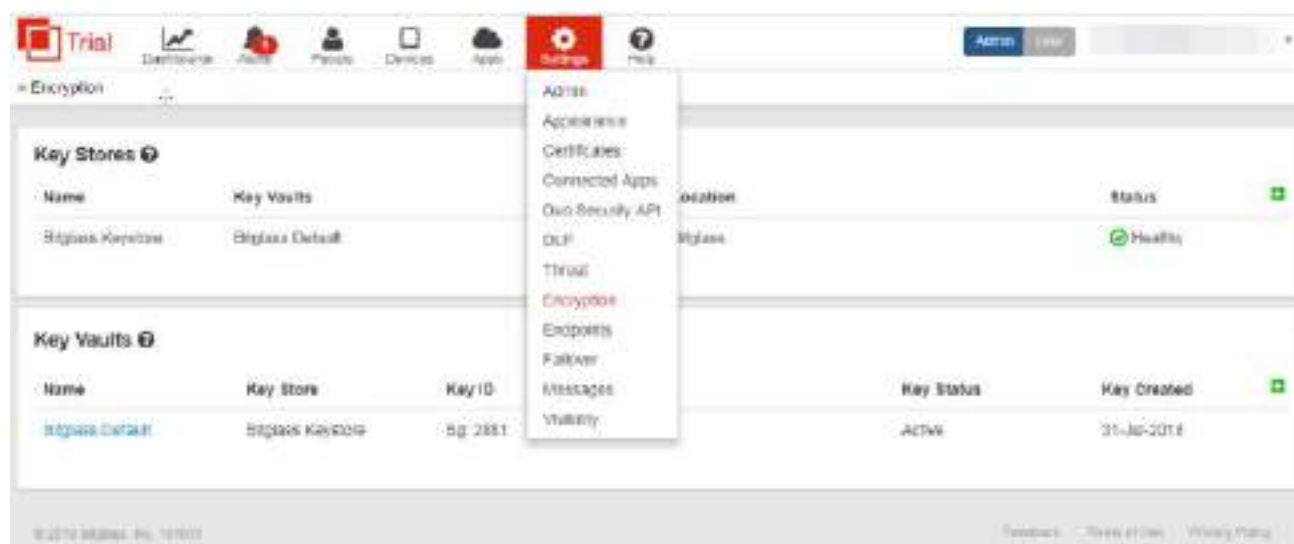
1. In the CipherTrust Manager UI, go to **Keys** and click the Key you have created using API.
2. Ensure that **Exportable** option is checked.
3. Select the user [created above](#) as the **Key Owner**.
4. Click **Update**.

Configuration on Bitglass

- Creating a Key Store
- Creating a Key

Creating a Key Store

1. In the Bitglass UI, go to **Settings > Encryption**.
2. Click the Add (+) button and specify the following:
 - Connection name
 - IP Address of the CipherTrust Manager
 - Port of the CipherTrust Manager
 - Username of the CipherTrust Manager. For Example, bitglass
 - Password of the CipherTrust Manager
 - Upload the downloaded CA Certificate



3. Click **Create** to create a key store.

After performing the above steps, you can check the logs under the CipherTrust Manager. A new connection should be generated under logs.

Creating a Key

1. In the Bitglass UI, go to **Settings > Encryption**.
2. Click the Add (+) button and specify the name.

3. Select the Key Store that is created in the [Creating a Key Store](#) section from the Key Store dropdown.
4. Click **Create** to create a key vault.

After creating a key, use the following steps to perform the Rekey operation:

1. From the Key Vaults, choose the key that is created in the [Creating a Key](#) section.
2. Click **Generate New Key** and click **Continue**. A new key will be created.

The status of old key will be **Decrypt only** and the new key is **Active**.

On the CipherTrust Manager, both the keys (new key and old key) are active. Moreover, you can also check the new key in the logs of the CipherTrust Manager.

Verifying Your Integration

Perform the following steps to ensure the integration is successful:

1. Create a test file on any node and then copy it to **/fs0**. Consider that the user created a file **test.txt**, and then copied it to **/fs0**.

```
cp /tmp/test.txt /fs0
```

2. Verify that file is encrypted with the key present in the CipherTrust Manager, using the following commands:

```
[root@ss1 ~]# /usr/lpp/mmfs/bin/mmlsattr -n gpfs.Encryption /fs0/test.txt
```

```
[root@ss1 ~]# /usr/lpp/mmfs/bin/mmlsattr -n gpfs.Encryption /fs0/test.txt
file name:           /fs0/test.txt
gpfs.Encryption:      "EAGC????#???????????????? ?????"??,???????????????? ?6[?6?
?B??d??L??l?????????m??f?test.txt aaaaaaaaaacccccccccceeeeeerrrrrrff?client?"
EncPar 'AES:256:XTS:FEC:HMI
      type: wrapped FEK  wrpPar 'AES:KWRAP'  CmbPar 'XORHMACSHA512'
      aaaaaaaaaacccccccccceeeeeerrrrrrff:client
```

```
[root@ss1 ~]# /usr/lpp/mmfs/bin/mmlsattr -L /fs0/test.txt
```



```
[root@ssl ~]# /usr/lpp/mmfs/bin/mmfsattr -L gpfs.Encryption /fs0/test.txt
gpfs.Encryption: Stat failed: No such file or directory

file name:                /fs0/test.txt
metadata replication: 1 max 2
data replication:         1 max 2
immutable:                no
appendOnly:               no
flags:
storage pool name:        system
fileset name:              root
snapshot name:
creation time:             Wed May 20 07:03:46 2020
Misc attributes:          ARCHIVE
Encrypted:                 yes
[root@ssl ~]#
```

Since the Encryption flag is true (Encrypted: yes), the integration is successful.

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the Autoregistration option is selected in Admin Settings > System > Interfaces > kmip . If it is not selected, turn on the Autoregistration option. Refer to the steps mentioned in Enable KMIP Client Registration .
User corresponding to username location in certificate (example: OU/UID) has not been created	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"</i>	Check whether the user corresponding to UID/OU of the node certificates has been created in Keys & Access Management > Users . If the user is not created, create a new user with the same name as the OU/UID field of the node certificates. After creating the user, add this user to the Key Admins . Refer to steps mentioned in Create a New User .
User has not been added to Key Admin group	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "authorization denied: verdict was deny: CreateKey"</i>	Check whether the user corresponding to UID/OU of the node certificates has been created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.

Issue	Error Message	Remediation
The Username location in Certificate has been set incorrectly	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"</i>	Check whether the Username Location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the Username Location in the Certificate.



CipherTrust Integrations

4 February 2026

Contents

- CipherTrust Manager Integration with MongoDB 3
 - Integration of MongoDB 3
 - Verifying Your Integration 17
 - Appendix. 17

CipherTrust Manager Integration with MongoDB

MongoDB is a document database designed for the ease of development and scaling. The Enterprise Edition of MongoDB lets you encrypt data files such that only users with the decryption key can decode and read the data.

If encryption is enabled, MongoDB Enterprise uses the the AES256-CBC via OpenSSL as default encryption mode. AES-256 uses a symmetric key (same key to encrypt and decrypt the text). CipherTrust Manager can be integrated with MongoDB to act as an external Key Management Server.



Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

CipherTrust Manager

- CipherTrust Manager 2.4 and higher

MongoDB Version

- MongoDB Enterprise 4.0 and higher

Integration of MongoDB

This section lists the steps to integrate MongoDB with CipherTrust Manager.

- [Prerequisites](#)

-
- [Configuration on CipherTrust Manager](#)
 - [Configuration on MongoDB](#)

Prerequisites

This section provides the prerequisites for integration of MongoDB with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager Documentation](#).
- MongoDB communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager.
- IP address of the CipherTrust Manager and port of the KMIP interface must be accessible from the MongoDB system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the MongoDB cluster is registered as a KMIP client on the CipherTrust Manager. Refer to KMIP Client Registration for more details.

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)
4. [Creating or Adding a CA \(optional\)](#)
5. [Registering a KMIP Client](#)
6. [Configuring the KMIP Interface](#)

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

1. Navigate to **Admin Settings > Domains**.
2. Click **Add Domain**. The **Add Domain** page appears.
3. Specify the following information:
 - **Name** - Enter the domain name.
 - **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.

- **Parent CA** - Select parent CA as root CA.
- **Allow Subdomain User Management** - Select this check box if you want to enable the sub-domain user management through this domain.

4. Click **Save**.
5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Keys & Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Create New User**.
5. On the Create a New User screen provide the following details:
 - Enter Username.
 - Enter Password.
6. Click **Create**. The newly created user is listed on the Users page.

Sub-Domain

Note

To create a user in sub-domain, you must enable **Allow Subdomain User management**.

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the sub-domain.
2. Open the **Keys & Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Create New User**.
5. On the **Create a New User** screen provide the following details:

-
- Enter **Username**.
 - Enter **Password**.

6. Click **Create**. The newly created user is listed on the Users page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
2. Click **Edit**.
3. Click **Group Memberships > Add Group**.
4. In the search bar, type **Key Admins** and select the check box corresponding to it .
5. Click **Add Group**.

Creating or Adding a CA (Optional)

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under Pending CAs section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate

4. Click **Add External CA**
-

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Create a Registration Token using the following steps:
 - a. Log on to the CipherTrust Manager in root domain.
 - b. Go to **Access Management > Registration Tokens**.
 - c. Click **Create New Registration Token**.

Copy the **Registration Token** once it is created.

2. Turn ON **Auto Registration** using the following steps:
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipsis button (...) corresponding to the kmip interface.
 - c. Click **Edit**.
 - d. Under **Configure KMIP** window, select **Auto Registration**.
 - e. Paste the **Registration Token**.
 - f. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - g. Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.

-
- d. Click **Certificate Details**.
 - e. Paste the content of `client.csr`.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a Name Prefix.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on what you are using.
 - f. Select the appropriate CA from the dropdown menu and click Select Profile.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA which was created using openssl and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.
 - Specify client name and paste the **Registration Token** generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub-Domain

You can register a KMIP client on the CipherTrust Manager through:

- **Auto Registration**
- **Manual Registration**

Auto-Registration

1. Create a Registration Token in the sub-domain using the following steps:
 - a. Log on to the CipherTrust Manager in your specified sub-domain.
 - b. Go to **Access Management > Registration Tokens**.
 - c. Click **Add Registration Token > Begin**.
 - d. Add a **Name Prefix**.
 - e. Specify a **Token Lifetime value** along with the **Client Capacity** for the token.
 - f. Copy the value of the Registration Token once it is created.
 - g. Switch to Root Domain.
2. Turn ON Auto Registration using the following steps:
 - a. Log on to the CipherTrust Manager in the root domain.
 - b. Go to **Admin Settings > Interfaces**.
 - c. Click the ellipsis button (...) corresponding to the kmip interface.
 - d. Click **Edit**.
 - e. Under **Configure KMIP** window, select **Auto Registration**.
 - f. Paste the **Registration Token**.
 - g. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
 - h. Click **Update**.

Manual Registration

1. Log on to the CipherTrust Manager into your domain.
 2. Go to **Products > KMIP**.
 3. Create **Client Profile** using the following steps:
-

-
- a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the Certificate Details section.
 - e. You can either paste the content of a generated client.csr or you can create one, by filling in the details.

For domain, the format to enter the Common Name field of the cert is always: `<domain name> | | <Hostname/IP>`

- f. Click **Save**.
4. Create a Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as **Local/External** depending on what you are using.
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.
 5. If you are using External CA then you can select the external CA which was created using openssl and uploaded on the CipherTrust Manager.
 6. Go to Registered Clients and click **Add Client**.
 - a. Specify client name and paste the Registration Token generated in the above step.

Note

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- b. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the Auto Registration is disabled.

4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

Local CA for Automatic Server Certificate Generation should be set to **Turn off auto generation from Local CA** in case of External CA.

6. Select the CA according to your preference:
 - If you are using **External CA** then select the CA under **External Trusted CAs**
 - If you are using **Local CA** then select the CA under **Local Trusted CAs**
7. If you are using an External CA, expand the **Upload Certificate** section:
 - In the **Certificate** field, paste the contents of Server Certificate, CA, and Server Key file in the same order. **Do not introduce any space or character or symbol between the contents of these files.**
 - Select certificate **Format** as **PEM**.
 - **Password** field is optional and can be skipped.
 - Click **Update**.

Sub-Domain

1. Switch to **Root Domain**.
2. Go to **Admin Settings > Interfaces**.
3. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
4. Select the **Auto Registration** checkbox if you auto-registered your KMIP client and paste the value of the registration token that you created.

Note

By default, the Auto Registration is disabled.

5. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
6. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Note

Local CA for Automatic Server Certificate Generation should be set to **Turn off auto generation from Local CA** in case of External CA.

7. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

External CA

Note

If you are using an External CA in the root Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Go to **Admin Settings > Interfaces**.
- b. Click the **Add** icon to add the External CA.
- c. Click **Update**.
- d. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.
- e. Select the **Certificate Chain** option and click **Build Certificate Chain**.
- f. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
- g. Select certificate Format as **PEM**.
- h. Click on **Upload Certificate**.

Further, You need to perform the following configuration on CipherTrust Manager specific to MongoDB:

Creating a Client Certificate

Note

- This section is applicable to KMIP clients registered using Auto Registration but not to External CA.
- This section is not applicable when using External CA.

1. Log on to the CipherTrust Manager.
2. Navigate to the **CA > Local CA** and click on the name of the CA certificate you need to issue the certificate from.
3. Click on **Issue Certificate**.
4. Enter the **Display Name**, followed by **Common Name**.

Note

The **Common Name** of the certificate should be the Hostname/IP of CipherTrust Manager. In case of domains, the **Common Name** of the certificate should be of the format:

`<domain name> | | <Hostname/IP>`

For more information on the format of **Common Name** of the certificate, refer to [LDAP and Multi-Domain Client Usernames in KMIP Certificates](#).

5. Select the **Algorithm** and **Size**, and click **Issue Certificate**.
6. Save the **Private Key** and the **CSR**.
7. Select the Certificate Purpose as client, specify the validity of the certificate in days, and click **Issue Certificate**.
8. Download a copy of this certificate by clicking the ellipsis next to the certificate name.

Configuration on MongoDB

You can integrate MongoDB with the CipherTrust Manager using either of the following methods:

Note

These options are available with **MongoDB Enterprise** only.

- [Integrate Using a New Key](#)
- [Integrate Using an Existing Key](#)

Integrate Using a New Key

1. Create a client certificate. Refer to [Creating a Client Certificate](#) if you are auto-registering your KMIP client.
2. Concatenate the **clientcertificate.pem** and **clientkey.pem** files into a single file using:

```
cat clientcertificate.pem clientkey.pem > client.pem
```

3. In case of domains, concatenate the Domain CA and Root CA into a single file using:


```
cat domainCA.pem rootCA.pem > ca.pem
```

4. Start MongoDB manually using the following parameters to create an encryption key:

```
mongod --enableEncryption
--kmipServerName <IP/Hostname of the CipherTrust Manager>
--kmipPort <KMIP port number of the CipherTrust Manager>
--kmipServerCAFile ca.pem
--kmipClientCertificateFile client.pem
--dbpath /var/lib/mongodb
```

Note

You will need to add the CipherTrust Manager's hostname to the `/etc/hosts` file on your MongoDB server so that the name can be resolved to the IP address of the CipherTrust Manager.

Integrate Using an Existing Key

1. Create a client certificate. Refer to [Creating a Client Certificate](#) if you are auto-registering your KMIP client.
2. Concatenate the **clientcertificate.pem** and **clientkey.pem** files into a single file using:

```
cat clientcertificate.pem clientkey.pem > client.pem
```

3. In case of domains, concatenate the Domain CA and Root CA into a single file using:

```
cat domainCA.pem rootCA.pem > ca.pem
```

4. Create a key on the CipherTrust Manager using the **AES** Algorithm. Set the state as **Active**.

```
{
  "name": "Test_Key",
  "usageMask": 12,
  "algorithm": "aes",
  "meta": {
    "ownerId": "local|mongodb"
  },
  "state": "Active",
  "deactivationDate": "2023-10-02T14:24:37.436073Z",
  "protectStopDate": "2023-10-02T14:24:37.436073Z",
}
```

```
"aliases": [
  {
    "alias": "altname2",
    "type": "string"
  },
  {
    "alias": "altname2:keysecure:gemalto:com",
    "type": "uri"
  }
]
```

e. Click **POST**.

5. Add the key management security options to `/etc/mongod.conf`.

The MongoDB configuration file contains a security section. In this section, update the KMIP server name, certificate details, KMIP port number, and Key ID to use the CipherTrust Manager as a key management service.

```
security:
enableEncryption: true
kmip:
  serverName: <IP/Hostname of the CipherTrust Manager>
  port: <KMIP port number of the CipherTrust Manager>
  serverCAFile: <Path to the CA certificate>
  clientCertificateFile: <Path to the client.pem>
  keyIdentifier: <ID of the key created on the CipherTrust Manager>
```

6. Start MongoDB manually using the following parameters:

```
mongod --enableEncryption
--kmipServerName <IP/Hostname of the CipherTrust Manager>
--kmipPort <KMIP port number of the CipherTrust Manager>
--kmipServerCAFile ca.pem
--kmipClientCertificateFile client.pem
--kmipKeyIdentifier <ID of the key created on the CipherTrust Manager>
```

Note

You will need to add the CipherTrust Manager's hostname to the `/etc/hosts` file on your MongoDB

server so that the name can be resolved to the IP address of the CipherTrust Manager.

After the service starts successfully, encryption is enabled and the integration is complete.

Verifying Your Integration

Perform the following steps to ensure the integration is successful:

When Integrating Using a New Key

1. Log on to the CipherTrust Manager using your credentials.
2. Navigate to **Keys** and click the first key that you see listed.
3. Copy the **ID** of this key.
4. On the server where MongoDB is installed, navigate to `/var/lib/mongodb/key.store/`.
5. Notice that a directory with the same name as the **ID** of the key that you just copied exists. This verifies that the integration is successful.

When Integrating with an Existing Key

1. Log on to the CipherTrust Manager using your credentials.
2. Navigate to **Keys** and click the key that you created using the API playground.
3. Copy the **ID** of this key.
4. On the server where MongoDB is installed, navigate to `/var/lib/mongodb/key.store/`.
5. Notice that a directory with the same name as the **ID** of the key that you just copied exists. This verifies that the integration is successful.

Appendix

Troubleshooting

You might encounter some issues while performing the integration steps mentioned in the previous sections. Mentioned below are some of the most common issues and the ways you can resolve them:

Issue	Error Message	Remediation
KMIP auto-registration is not	The following error message is	Check whether the Auto registration

Issue	Error Message	Remediation
enabled	displayed in the CipherTrust Manager Records: "errorMessage": "Unregistered client, please register a new client from CLI or API or UI."	option is selected in Admin Settings > System Interfaces > kmip . If it is not selected, turn on the Auto registration option. Refer to Configuration steps for CipherTrust Manager using the GUI or ksctl.
Incorrect KMIP interface mode has been selected	The following error message is displayed in the mongod logs: "initandlisten Unable to retrieve key .system, error: socket exception [CONNECT_ERROR] for The server certificate does not match the host name."	Check whether the mode selected in Admin Settings > System Interfaces > kmip is "TLS, verify client cert, allow anonymous logins" or not. If not, make appropriate changes.
For any errors encountered while enabling the encryption		Clean up the contents in this directory using the command: <code>sudo rm -rf /var/lib/mongodb/*</code> . Now, retry enabling the encryption.



CipherTrust Integrations

4 February 2026

Contents

- DataStax Enterprise 3
 - Integration with CipherTrust Manager 3
 - Verifying Your Integration 13
 - Licensing 13
 - Appendix. 13

DataStax Enterprise

DataStax Enterprise (DSE), built on Apache Cassandra™, is the world's most scalable database, well known for 100% uptime, unmatched low latency, and the ability to handle massive data at planetary scale. DSE delivers enterprise capabilities used by the best internet companies, and hardened by demanding, modern use cases in e-commerce, customer loyalty, banking, IoT, supply chain, etc. at global leading companies, including 40 of the Fortune 100. It provides advanced security provides features that allow you to minimize risk, achieve granular access control, and work toward compliance with regulatory standards.

DataStax Enterprise supports using encryption keys from one or more remote Key Management Interoperability Protocol (KMIP) hosts to encrypt or decrypt table data or sensitive properties, or both. You can set up encryption using keys from the CipherTrust Manager to protect sensitive configuration file properties, system resources, and tables.

Supported Product Versions

Note

The integration is performed and certified on **LTS release versions** of CipherTrust Manager only. Feature releases support the integration and are covered under technical support but will not be explicitly certified. To know more about CipherTrust Manager release versions, click [here](#).

This integration is validated on the following software versions:

CipherTrust Manager

- CipherTrust Manager 2.16.1 and higher

DataStax Enterprise

- DataStax Enterprise 6.8 and higher

Operating System

Red Hat Enterprise Linux

Integration with CipherTrust Manager

This section lists the steps to integrate DataStax Enterprise with CipherTrust Manager.

- [Prerequisites](#)
- [Configuration on CipherTrust Manager](#)
- [Configuration on DataStax Cassandra](#)

Prerequisites

This section provides the prerequisites for integration of Datastax with CipherTrust Manager.

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager Documentation](#).
- Datastax communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the Datastax system.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the Datastax cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.
- Ensure that Datastax has proper license for data at Rest Encryption. To check the license information in Datastax, navigate to **Settings > Software and Licenses > License Information**.
- Ensure that the required licenses are activated on CipherTrust Manager. For more details, refer to the [CipherTrust Manager documentation](#).

Configuration on CipherTrust Manager

To configure the CipherTrust Manager, you need to perform the following steps:

1. [Creating a Domain \(Optional\)](#)
2. [Creating a User](#)
3. [Assigning User to a Group](#)
4. [Registering a KMIP Client](#)
5. [Configuring the KMIP Interface](#)

Creating a Domain (Optional)

Perform the following steps to be performed on CipherTrust Manager:

1. Navigate to **Admin Settings > Domains**.
2. Click **Add Domain**. The **Add Domain** page appears.
3. Specify the following information:
 - **Name** - Enter the domain name.

-
- **Admins** - Select the admins (one or more) from the list available in the drop down. For example, admin.
 - **Parent CA** - Select parent CA as root CA.
 - **Allow Subdomain User Management** - Select this check box if you want to enable the subdomain user management through this domain.

4. Click **Save**.
5. Switch to the newly created domain by clicking the top right on the current Domain Name.

Creating a User

Root Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI.
2. Open the **Keys & Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Create New User**.
5. On the Create a New User screen provide the following details:
 - Enter Username.
 - Enter Password.
6. Click **Create**. The newly created user is listed on the Users page.

Sub Domain

To create a user, perform the following steps:

1. Log on to the CipherTrust Manager GUI with the User you created within the Sub Domain.
2. Open the **Keys & Access Management** application.
3. On the left pane, click **Users**. The Users page is displayed.
4. On the Users page, click **Create New User**.
5. On the **Create a New User** screen provide the following details:
 - Enter **Username**.
 - Enter **Password**.

-
6. Click **Create**. The newly created user is listed on the Users page.

Assigning User to a Group

Perform the following steps to add user to a group:

1. Click the ellipsis button (...) corresponding to the user that you created in the previous step.
2. Click **Edit**.
3. Click **Group Memberships > Add Group**.
4. In the search bar, type **Key Admins** and select the check box corresponding to it .
5. Click **Add Group**.

Creating or Adding a CA (Optional)

Local CA

To create/add a Self-signed local CA, perform the following steps:

1. Navigate to **CA > Local**. Select **Add Local CA**, the **Add Local CA** page appears.
2. Provide the required information and click **Add Local CA**. The created Local CA will appear under Pending CAs section.
3. Click the ellipsis against the Local CA that you created and select the option **Self-sign**.
4. Select a valid duration for the Local CA. Click **Save**.

External CA

To add an external CA, perform the following steps:

1. Navigate to **CA > External**. Select **Add External CA**, the **Add External Certificate** page appears.
2. Provide the required information.
3. If you want to upload the external CA, select the **File Upload** option and click **Upload Certificate**. Browse and select the required External CA.

OR

Select **Text** and paste the contents of External Certificate

4. Click **Add External CA**.

Registering a KMIP Client

Root Domain

You can register a KMIP client on the CipherTrust Manager through:

1. Log on to the CipherTrust Manager.
2. Go to **Products > KMIP**.
3. Create Client Profile using the following steps:
 - a. Go to **Client Profile** and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in **Username Location in Certificate**.
 - d. Click **Certificate Details**.
 - e. Paste the content of `client.csr`.
 - f. Click **Save**.
4. Create Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a Name Prefix.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as Local/External depending on CA that you created [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

Note

If you are using External CA then you can select the external CA that you created and uploaded on the CipherTrust Manager.

5. Go to **Registered Clients** and click **Add Client**.
-

-
- Specify client name and paste the **Registration Token** generated in the above step.

If you are using external CA then you need to paste the signed client certificate in the **Client Certificate** field.

- Click **Save** to save the **client certificate**.

Sub Domain

You can register a KMIP client on the CipherTrust Manager through:

1. Log on to the CipherTrust Manager into your domain.
2. Go to **Products > KMIP**.
3. Create **Client Profile** using the following steps:
 - a. Go to Client Profile and click **Add Profile**.
 - b. Add a **Profile Name**.
 - c. Select **CN** in Username Location in Certificate.
 - d. Expand the Certificate Details section.
 - e. You can either paste the content of a generated client.csr or you can create one, by filling in the details.

For domain, the format to enter the Common Name field of the cert is always:

`domainName | | domainUser`

- f. Click **Save**.
4. Create a Registration Token using the following steps:
 - a. Go to **Registration Token** and click **New Registration Token > Begin**.
 - b. Add a **Name Prefix**.
 - c. Specify a Token Lifetime value along with the Client Capacity for the token.
 - d. Click **Select CA**.
 - e. Select the CA type as **Local/External** depending on what you are using [above](#).
 - f. Select the appropriate CA from the dropdown menu and click **Select Profile**.
 - g. Select the Client Profile from the dropdown which you have created in the above step.
 - h. Click **Create Token**.
 - i. Copy the value of the Token created and click **Done**.

-
5. If you are using External CA then you can select the external CA which was created using openssl and uploaded on the CipherTrust Manager.
 6. Go to Registered Clients and click **Add Client**.

- a. Specify client name and paste the Registration Token generated in the above step.

If you are using external CA then you need to paste the signed client certificate in the Client Certificate field.

- b. Click **Save** to save the client certificate.

Configuring the KMIP Interface

The KMIP interface can be configured through:

Root Domain

1. Go to **Admin Settings > Interfaces**.
2. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
3. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
4. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Local CA for Automatic Server Certificate Generation should be set to **Turn off auto generation from Local CA** in case of External CA.

5. Depending upon the CA that you created [above](#), select an option from the following:
6. If you are using **Local CA** then select the CA under **Local Trusted CAs**
7. If you are using **External CA** then select the CA under **External Trusted CAs** provided you have the external CA that you created and uploaded on the CipherTrust Manager. For more information on creating or uploading CA, refer to [Creating or Adding a CA](#).

Sub Domain

1. Switch to **Root Domain**.
2. Go to **Admin Settings > Interfaces**.
3. On the KMIP Interface, click the ellipsis button (...) and then click **View/Edit**.
4. Select the mode as **TLS, verify client cert, user name taken from client cert, auth request is optional**.
5. Specify selections for **Local CA for Automatic Server Certificate Generation** as desired.

Local CA for Automatic Server Certificate Generation should be set to **Turn off auto generation from**

Local CA in case of External CA.

6. Select the CA according to your preference.

Local CA

- a. Login to your sub-domain. Go to **CA > Local**. Click the ellipsis (...) and copy the contents of your CA Certificate.
- b. Logout of your sub-domain and now login to the root domain.
- c. Go to **CA > External > Add External CA**.
- d. Enter a name for this Domain CA and select the text radio button and paste the certificate contents.
- e. Click **Add External CA**.
- f. Go to **Admin Settings > Interfaces**.
- g. Click the **Add** icon to add the External CA.
- h. Click **Update**.

External CA

Note

If you are using an External CA in the root Domain, you need to add the CA as an External CA in both the root domain as well as the sub-domain and modify the interface accordingly.

- a. Go to **Admin Settings > Interfaces**.
- b. Click the **Add** icon to add the External CA.
- c. Click **Update**.
- d. On the **KMIP interface**, click the ellipsis (...) > **Certificate Options > Upload New Certificate > Ok**.
- e. Select the **Certificate Chain** option and click **Build Certificate Chain**.
- f. Click on Text and paste the contents of Server Certificate, CA, and Server Key file in the same order. Do not introduce any space, character or symbol between the contents of these files.
- g. Select certificate Format as **PEM**.
- h. Click on **Upload Certificate**.

Creating a Client Certificate

1. Log on to the CipherTrust Manager.
2. Navigate to the **CA > Local CA** and click on the name of the CA certificate you need to issue the certificate from.
3. Click on **Issue Certificate**.
4. Enter the **Display Name**, followed by **Common Name**.

The common Name of the certificate should be the same as the name of the user you created on CipherTrust Manager. Refer to [Creating a User](#) section.

For more information on the format of common name of the certificate, refer to [LDAP and Multi-Domain Client Usernames in KMIP Certificates](#).

5. Select the **Algorithm** and **Size**, and click **Issue Certificate**.
6. Save the **Private Key** and the **CSR**.
7. Select the Certificate Purpose as client, specify the validity of the certificate in days, and click **Issue Certificate**.
8. Download a copy of this certificate by clicking the ellipsis next to the certificate name.

Configuration on DataStax Cassandra

1. Verify the installed version of Java.

```
java --version
```

2. Verify that you have python installed:

```
python3 --version
```

3. Download and extract the **tar** file for DataStax Enterprise. Create a new data directory and logging directory or use the existing default data directory and logging directory locations. Create and change ownership for these directories using:

```
sudo mkdir -p /var/lib/cassandra
sudo chown -R $USER:$GROUP /var/lib/cassandra
sudo mkdir -p /var/log/cassandra
sudo chown -R $USER:$GROUP /var/log/cassandra
```

4. To edit the configuration file `cassandra.yaml`, perform the following steps:
-

```
data_file_directories:
  - <installation_location>/dse-data/data

commitlog_directory: <installation_location>/dse-data/commitlog
saved_caches_directory: <installation_location>/dse-data/saved_caches
hints_directory: <installation_location>/dse-data/hints
cdc_raw_directory: <installation_location>/cdc_raw
```

For example:

```
data_file_directories:
  - /root/dse-6.8.53/dse-data/data

commitlog_directory: /root/dse-6.8.53/dse-data/commitlog
saved_caches_directory: /root/dse-6.8.53/dse-data/saved_caches
hints_directory: /root/dse-6.8.53/dse-data/hints
cdc_raw_directory: /root/dse-6.8.53/dse-data/cdc_raw
```

5. Start the Cassandra (DSE) service:

```
bin/dse cassandra -R
```

6. Verify the status:

```
cd <installation_location>/bin
./nodetool status
```

7. Check if the service is up and running, you will get a similar output as mentioned below:

```
Datacenter: Cassandra
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens   Owns    Host ID
Rack
UN  127.0.0.1    82.43 KB     128      ?       40725dc8-7843-43ae-9c98-7c532b1f517e
rack1
```

8. Navigate to the `conf` directory.

```
cd <installation_location>/resources/dse/conf
```


- Update the **kmip_hosts** section in the **dse.yaml** file to include the host details.

```
kmip_hosts:
  kmip_group_name:
    hosts: <IP_of_CipherTrust_Manager>
    keystore_path: <installation_location>/certificates/kmip_truststore.jks
    keystore_type: jks
    keystore_password: <keystore_password>
    truststore_path: <installation_location>/certificates/kmip_truststore.jks
    truststore_type: jks
    truststore_password: <truststore_password>
```

Verifying Your Integration

To verify that the integration is successful and DataStax can connect to CipherTrust Manager:

- Create a key by executing the following command:

```
cd <installation_location>/bin
./dsetool createsystemkey 'AES/ECB/PKCS5' 128 -k kmip_group_name
```

- To list the key created above:

```
./dsetool managekmip list kmip_group_name
```

Licensing

This section provides the licensing requirements for KMIP client.

Condition	License Requirement
Encryption per instance	x1 KMIP client license for each instance

Appendix

Troubleshooting

Issue	Error Message	Remediation
KMIP registration is not enabled	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "Unregistered client, please register a new client from CLI or API or UI."</i>	Check whether the Autoregistration option is selected in Admin Settings > System > Interfaces > kmip . If it is not selected, turn on the

Issue	Error Message	Remediation
		Autoregistration option. Refer to the steps mentioned in Enable KMIP Client Registration .
User corresponding to username location in certificate (example: OU/UID) has not been created	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"</i>	Check whether the user corresponding to UID/OU of the node certificates has been created in Access Management > Users . If the user is not created, create a new user with the same name as the OU/UID field of the node certificates. After creating the user, add this user to the Key Admins . Refer to steps mentioned in Create a New User .
User has not been added to Key Admin group	The following error message is displayed in CipherTrust Records: <i>"errorMessage": "authorization denied: verdict was deny: CreateKey"</i>	Check whether the user corresponding to UID/OU of the node certificates has been created under Keys & Access Management > Users . If the user is not added, add the user to the Key Admins group.
The Username location in Certificate has been set incorrectly	The following error messages are displayed in CipherTrust Records: <i>"errorMessage": "username not found: "</i> <i>"errorMessage": "Could not authenticate certificate user, hence client is not auto registered in mode tls-cert-pw-opt"</i>	Check whether the Username Location in the Certificate option is set correctly to OU/UID in Admin Settings > System > Interfaces > kmip . If it is not set correctly, set the correct value for the Username Location in the Certificate.



CipherTrust Integrations

4 February 2026

Contents

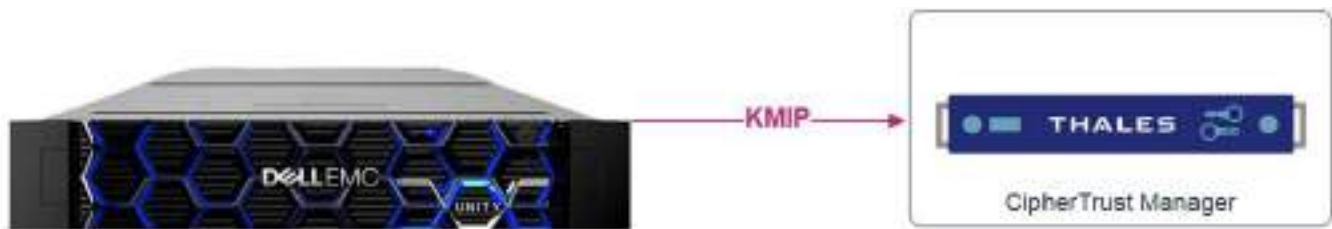
- Dell EMC Unity Storage 3
 - Configuring CipherTrust Manager 4
 - Configuring Dell EMC Unity Storage 6
 - Licensing 8
 - Appendix. 9

Dell EMC Unity Storage

Dell EMC Unity Storage helps the organization to solve the data storage needs. It provides the following benefits:

- Simplifies the data landscape. It streamlines the data storage needs with extensive coverage for enterprise workloads.
- Automates data services. It simplifies the storage management with policy-based automation and data mobility capabilities to quickly move data between environments or clouds.
- Extracts insight from data. Dell EMC provides a complete portfolio of solutions to unlock the value of data capital, with tight integration with leading applications and analytics platforms.
- Optimizes any workload at scale. It automate management to control costs and empower end-users with extensive self-service capabilities that lighten the load on IT. Secure your data assets. Significantly reduce application outages and latency issues while reducing storage requirements and network loads with advanced storage and deduplication.

CipherTrust Manager integrates with Dell EMC Unity Storage to act as an external KMS and secure the data when it is at rest. The integration reduces the threat of unauthorized access to the encrypted data.



Supported Product Versions

Note

This integration has been validated in the field by a partner or in a customer environment with the following software versions. It is recommended to test the integration in a non-production environment with desired software versions before deploying it to production. Thales will provide best-effort support.

CipherTrust Manager

- CipherTrust Manager 2.4

Dell EMC Unity

- Dell EMC Unity 480F

Prerequisites

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust Manager documentation](#).
- Dell EMC Unity Storage communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on CipherTrust Manager. Refer to [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from the Dell EMC Unity Storage.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that each node of the Unity cluster is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.
- Ensure that Dell EMC Unity Storage has proper license for Data at Rest Encryption. To check the license in Dell EMC Unity, go to **Settings > Software and Licenses > License information**.
- Ensure that the required licenses are activated. For more details, refer to the [CipherTrust Manager documentation](#).

Steps for Integration

To integrate CipherTrust Manager with Dell EMC Unity Storage:

1. [Configure the CipherTrust Manager](#)
2. [Configure the Dell EMC Unity Storage](#)

Configuring CipherTrust Manager

This section includes the configuration steps for the CipherTrust Manager.

1. Log on to the CipherTrust Manager as admin to root domain (root/admin).
2. Go to **Admin Settings > Interfaces**, ensure that KMIP interface is enabled and check authentication methods and supported cipher-suites.

For more information, refer to [Dell EMC Documentation](#).

3. Go to **KMIP > Client Profile** and click **Add Profile** button to add new client profile, where CN=<user name>.
4. Go to **KMIP > Registration Token** and click **New Registration Token** button to create a Registration Token for the added Client Profile.

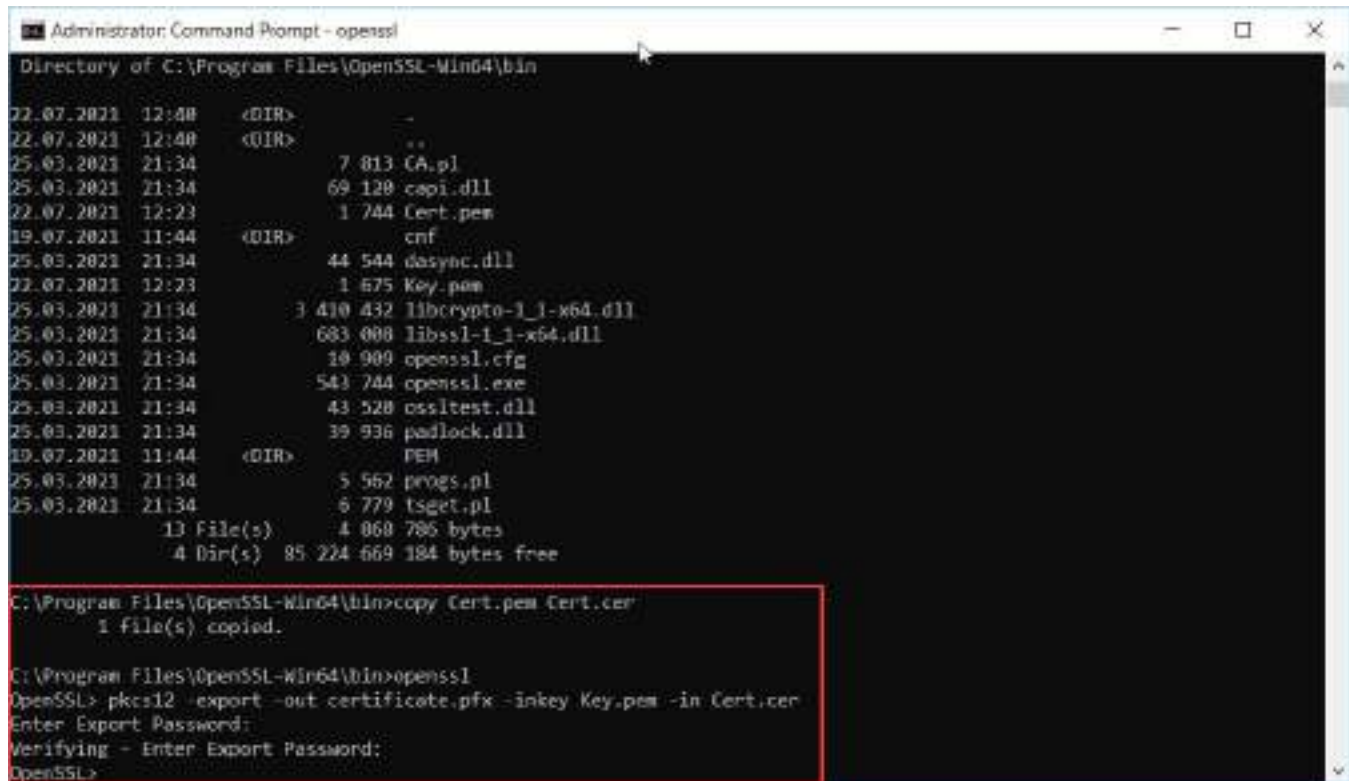
Copy the generated Registration Token.

- Go to **KMIP > Registered Clients** and click **Add Client** button to add a new client based on the created Registration Token.

Click **Save Private Key** and **Save Certificate** to download the private key and certificate.

- Go to **CA > Local** and download the Root CA Certificate.

After performing the above steps, move the client certificate and private key for Dell EMC Unity Storage provider in one file (PFX). Refer the following images:



```
Administrator Command Prompt - openssl
Directory of C:\Program Files\OpenSSL-Win64\bin
22.07.2021 12:40 <DIR> .
22.07.2021 12:40 <DIR> ..
25.03.2021 21:34      7 813 CA.pl
25.03.2021 21:34    69 120 capi.dll
22.07.2021 12:23      1 744 Cert.pem
19.07.2021 11:44 <DIR>   cnf
25.03.2021 21:34    44 544 dasync.dll
22.07.2021 12:23      1 675 Key.pem
25.03.2021 21:34    3 410 432 libcrypto-1_1-x64.dll
25.03.2021 21:34    683 000 libssl-1_1-x64.dll
25.03.2021 21:34    19 900 openssl.cfg
25.03.2021 21:34    543 744 openssl.exe
25.03.2021 21:34    43 528 ssltest.dll
25.03.2021 21:34    39 936 padlock.dll
19.07.2021 11:44 <DIR>   PEM
25.03.2021 21:34      5 562 prog.pl
25.03.2021 21:34      6 779 tsget.pl
      13 File(s)      4 060 786 bytes
       4 Dir(s)    85 224 669 184 bytes free

C:\Program Files\OpenSSL-Win64\bin>copy Cert.pem Cert.cer
1 file(s) copied.

C:\Program Files\OpenSSL-Win64\bin>openssl
OpenSSL> pkcs12 -export -out certificate.pfx -inkey Key.pem -in Cert.cer
Enter Export Password:
Verifying - Enter Export Password:
OpenSSL>
```

```

22.07.2021 13:11 <DIR> ..
25.03.2021 21:34 7 813 CA.pl
25.03.2021 21:34 69 120 capi.dll
22.07.2021 12:23 1 744 Cert.cer
22.07.2021 12:23 1 744 Cert.pem
22.07.2021 13:11 2 853 certificate.pfx
19.07.2021 11:44 <DIR> cnf
25.03.2021 21:34 44 544 dasync.dll
22.07.2021 12:23 1 675 Key.pem
25.03.2021 21:34 3 410 432 libcrypto-1_1-x64.dll
25.03.2021 21:34 683 008 libssl-1_1-x64.dll
25.03.2021 21:34 10 909 openssl.cfg
25.03.2021 21:34 543 744 openssl.exe
25.03.2021 21:34 43 520 ossltest.dll
25.03.2021 21:34 39 936 padlock.dll
19.07.2021 11:44 <DIR> PEM
25.03.2021 21:34 5 562 progs.pl
25.03.2021 21:34 6 779 tsget.pl
15 File(s) 4 873 383 bytes
4 Dir(s) 85 233 520 640 bytes free

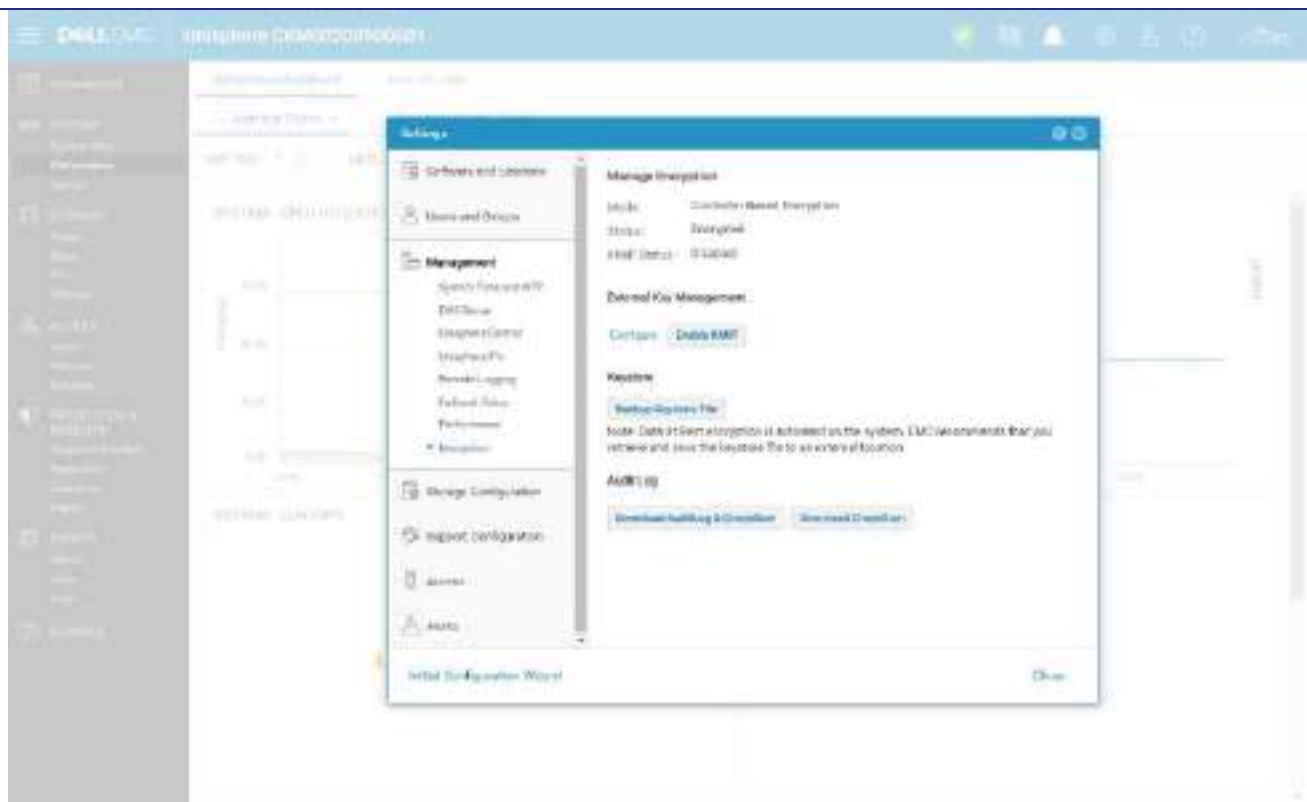
C:\Program Files\OpenSSL-Win64\bin>

```

Configuring Dell EMC Unity Storage

This section includes the configuration steps for Dell EMC Unity storage.

1. From the Dell EMC Unity UI, go to **Settings**.
2. Under **External Key Management**, click **Configure**.

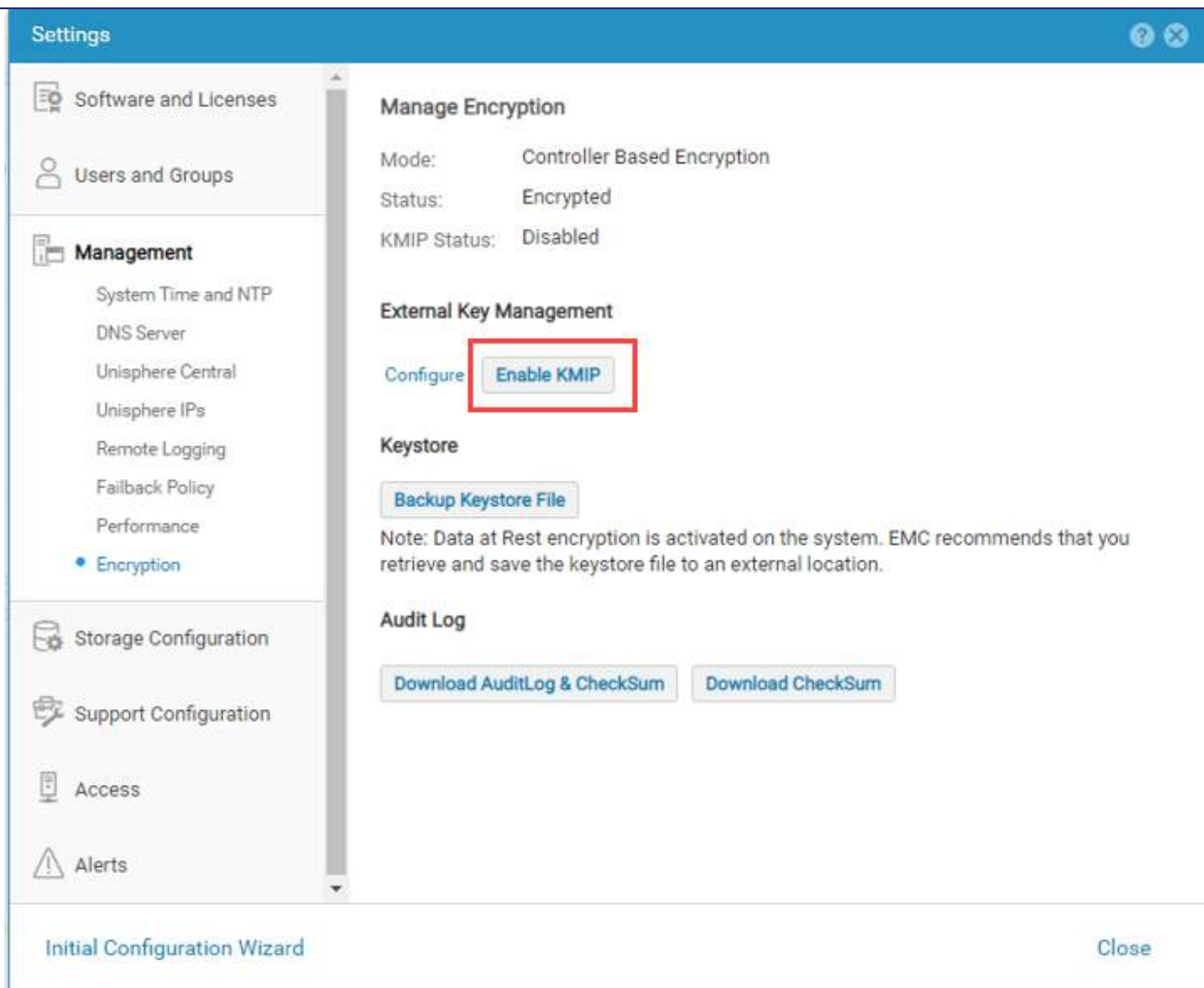


3. In the **KMIP Configuration** window, specify the required details and click **Manage Certificates**.

Note

Under **IP Addresses of the KMIP Server Cluster**, you can add one or more than one (cluster) CipherTrust Manager instance.

4. In the **Certificate Management** window, click **Add** button and import the client certificate and the internal root CA of the CipherTrust Manager.
5. In the **Settings** window, click **Enable KMIP**.

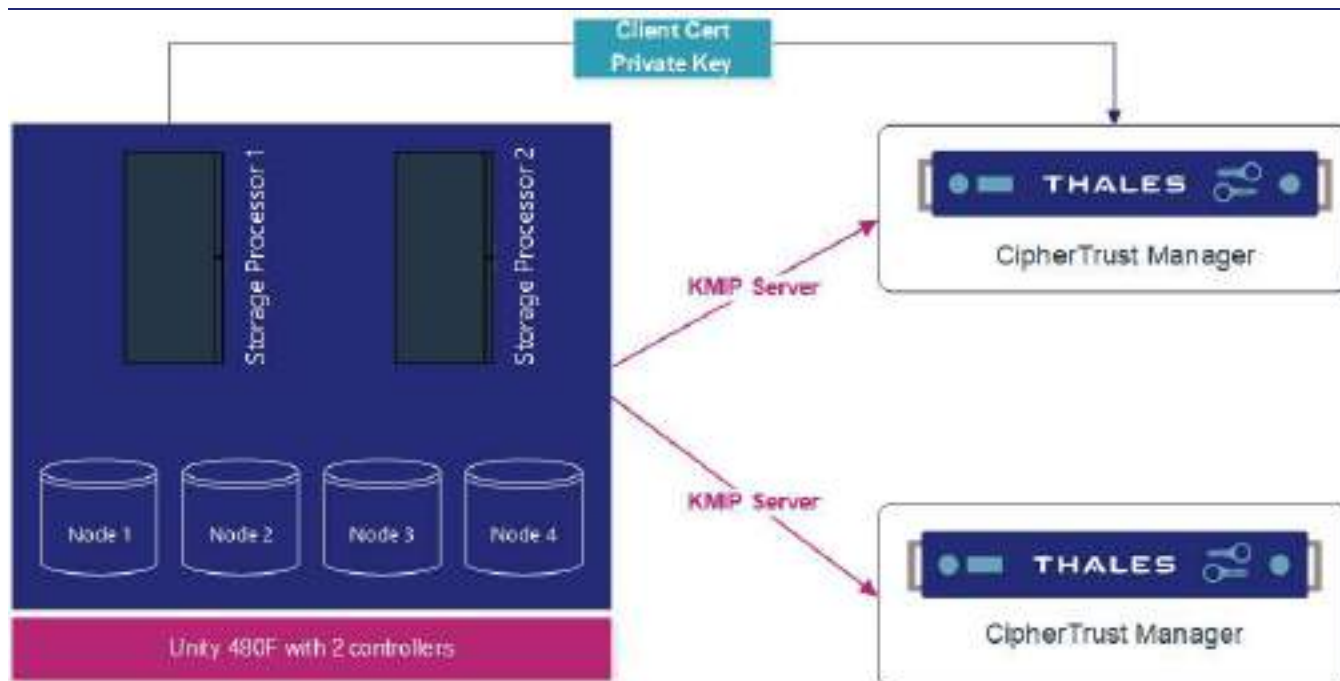


After the successful completion of the above steps, Dell EMC Unity:

- Creates keys on the CipherTrust Manager.
- Encrypts drives

Licensing

Number of required KMIP license(s) should be equivalent to the number of available/active controller in all matrix communicating with the CipherTrust Manager.



Appendix

Verifying integration

After successful integration:

- Unity creates keys in the CipherTrust Manager.

Keys

Name:

Filters: ☒ Disk ☐ Raw

Types: Size: Status: Dates:

☐ Latest Version Only

Type: Symmetric Key ☒

[+ Create a New Key](#)

Key Name	Version	Owner	Modified	Type	Algorithm	Size	Links
ka-010921e60c4bce82a2747f5e8b77194a72a20c...	0	local/admin	00 Jul 2021, 01:58	Symmetric	ACS	256	...
ka-cf238373f80e4802acc51171e1f829452284e91e...	0	local/c1f0735-9679-48ee-91bb-471aced8420f	22 Jul 2021, 11:58	Symmetric	ACS	256	...
ctruv-88781c63-f725-4c80-9d71-d271ee6bce...	0	No owner	23 Jun 2021, 02:54	Symmetric	ACS	256	...

3 Keys 50 per page

- Unity encrypts drives as shown in the following logs:

```

2021-07-15 09:35:01 0000000000000000 ADD_SLIC FCMD200100048 SPB BE 1
2021-07-15 09:35:06 0000000000000000 ENCRYPTION_ENABLED FCMS194900079
2021-07-15 09:35:06 0000000000000000 CREATE_RG_KEY OBJ_ID 11 RG_ID 1004 POS 0 DRIVE ZVNA0M701996
2021-07-15 09:35:06 0000000000000000 CREATE_RG_KEY OBJ_ID 11 RG_ID 1004 POS 1 DRIVE ZVNA0M702189
2021-07-15 09:35:06 0000000000000000 CREATE_RG_KEY OBJ_ID 11 RG_ID 1004 POS 2 DRIVE ZVNA0M702162
2021-07-15 09:35:06 0000000000000000 CREATE_RG_KEY OBJ_ID 11 RG_ID 1004 POS 3 DRIVE ZVNA0M702195
2021-07-15 09:35:06 0000000000000000 ENCRYPTION_ENABLED FCMS194900079
2021-07-15 09:35:06 0000000000000000 ENCRYPTION_ACTIVATION_START FCMS194900079
2021-07-15 09:35:06 0000000000000000 ENCRYPTION_ACTIVATION_COMPLETED FCMS194900079
2021-07-15 09:35:09 0000000000000011 RG_ENCRYPTION_COMPLETED OBJ_ID 11 RG_ID 1004 POS 0 DRIVE ZVNA0M701996
2021-07-15 09:35:09 0000000000000012 RG_ENCRYPTION_COMPLETED OBJ_ID 11 RG_ID 1004 POS 1 DRIVE ZVNA0M702189
2021-07-15 09:35:09 0000000000000013 RG_ENCRYPTION_COMPLETED OBJ_ID 11 RG_ID 1004 POS 2 DRIVE ZVNA0M702162
2021-07-15 09:35:09 0000000000000014 RG_ENCRYPTION_COMPLETED OBJ_ID 11 RG_ID 1004 POS 3 DRIVE ZVNA0M702195
2021-07-15 09:35:10 0000000000000019 RG_ENCRYPTION_FINALIZED OBJ_ID 11 RG_ID 1004 POS 0 DRIVE ZVNA0M701996
2021-07-15 09:35:10 0000000000000016 RG_ENCRYPTION_FINALIZED OBJ_ID 11 RG_ID 1004 POS 1 DRIVE ZVNA0M702189
2021-07-15 09:35:10 0000000000000017 RG_ENCRYPTION_FINALIZED OBJ_ID 11 RG_ID 1004 POS 2 DRIVE ZVNA0M702162
2021-07-15 09:35:10 0000000000000018 RG_ENCRYPTION_FINALIZED OBJ_ID 11 RG_ID 1004 POS 3 DRIVE ZVNA0M702195
2021-07-15 09:42:52 0000000000000019 CREATE_RG_KEY OBJ_ID 106 RG_ID 0 POS 0 DRIVE ZVNA0M702235
2021-07-15 09:42:52 0000000000000014 CREATE_RG_KEY OBJ_ID 106 RG_ID 0 POS 1 DRIVE ZVNA0M702012
2021-07-15 09:42:52 0000000000000018 CREATE_RG_KEY OBJ_ID 106 RG_ID 0 POS 2 DRIVE ZVNA0M702013
2021-07-15 09:42:52 000000000000001C CREATE_RG_KEY OBJ_ID 106 RG_ID 0 POS 3 DRIVE ZVNA0M702070
2021-07-15 09:42:52 000000000000001D CREATE_RG_KEY OBJ_ID 106 RG_ID 0 POS 4 DRIVE ZVNA0M702230
2021-07-15 09:42:52 000000000000001E CREATE_RG_KEY OBJ_ID 106 RG_ID 0 POS 5 DRIVE ZVNA0M702232
2021-07-15 09:42:52 000000000000001F CREATE_RG_KEY OBJ_ID 106 RG_ID 0 POS 6 DRIVE ZVNA0M702162
2021-07-15 09:42:52 0000000000000020 CREATE_RG_KEY OBJ_ID 106 RG_ID 0 POS 7 DRIVE ZVNA0M702195
2021-07-15 09:42:52 0000000000000021 CREATE_RG_KEY OBJ_ID 106 RG_ID 0 POS 8 DRIVE ZVNA0M702189
2021-07-15 09:42:52 0000000000000022 CREATE_RG_KEY OBJ_ID 106 RG_ID 0 POS 9 DRIVE ZVNA0M701996
2021-07-20 12:00:23 0000000000000025 KNIP_ENABLED FCMS194900079
2021-07-20 12:32:13 0000000000000026 KNIP_DISABLED FCMS194900079
2021-07-20 12:34:06 0000000000000027 BACKUP_START FCMS194900079
2021-07-20 12:34:10 0000000000000028 BACKUP_COMPLETED FCMS194900079
2021-07-20 12:34:11 0000000000000029 BACKUP_START FCMS194900079
2021-07-20 12:34:15 000000000000002A BACKUP_COMPLETED FCMS194900079
2021-07-20 13:21:42 000000000000002B KNIP_ENABLED FCMS194900079
2021-07-20 13:29:51 000000000000002C KNIP_DISABLED FCMS194900079

```

Troubleshooting

Issue: KMIP Server is not available for Unity Storage.

Solution: Check the network connectivity and the required licenses.



CipherTrust Integrations

4 February 2026

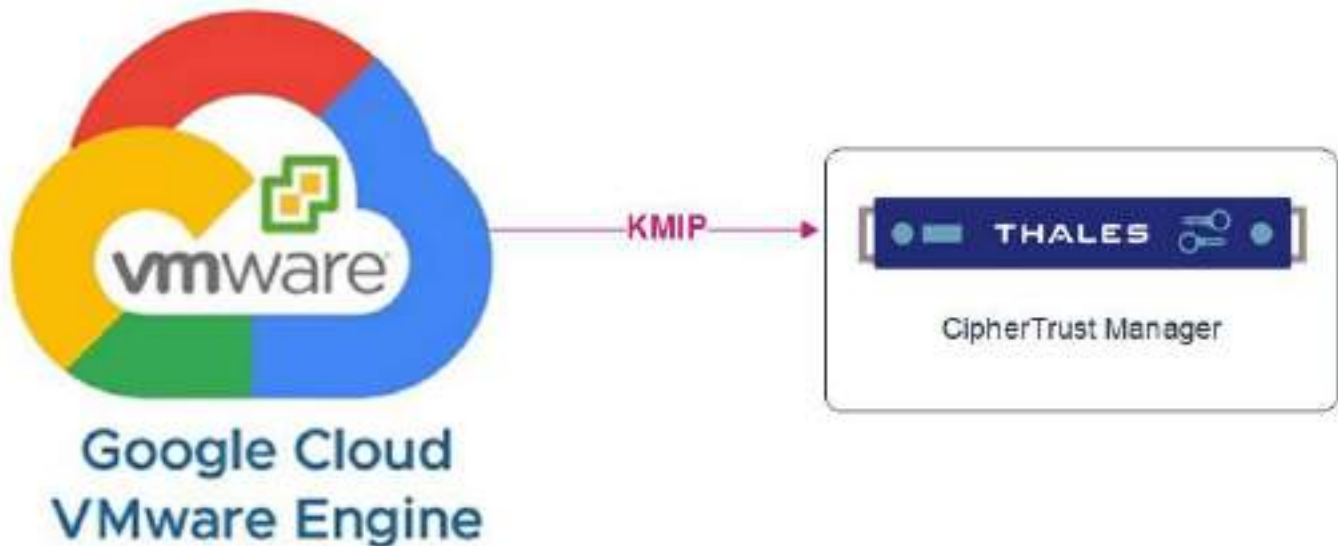
Contents

- Google Cloud VMware Engine (GCVE) 3
 - Licensing 4
 - Configuring CM on Google Cloud Platform (GCP) 4
 - Enabling vSAN Encryption 7
 - Enabling VMware KMIP Connection 8
 - Registering KMIP Client 14
 - Configuring VMWare VCenter 17

Google Cloud VMware Engine (GCVE)

Google Cloud VMware Engine (GCVE) enables you to migrate or extend your on-premise VMware workloads to Google Cloud without refactoring the applications. This allows you to benefit from increased scale and agility while continuing to leverage the value of your existing VMware investments. Additionally, it helps you to maintain continuity with existing tools, policies, and unify management across environments.

CipherTrust Manager (CM) can be integrated with GCVE to act as an external KMS, as shown in the following diagram.



Supported Product Versions

Note

This integration has been validated in the field by a partner or in a customer environment with the following software versions. It is recommended to test the integration in a non-production environment with desired software versions before deploying it to production. Thales will provide best-effort support.

Google Cloud vSphere

- GCVE 6.7

CipherTrust Manager

- CipherTrust Manager 1.9.1

Prerequisites

- Ensure that the CipherTrust Manager is installed and configured. For more details, refer to the [CipherTrust](#)

Manager documentation.

- GCVE communicates with the CipherTrust Manager using the KMIP interface. Ensure that the KMIP interface is configured on the CipherTrust Manager. Refer to the [CipherTrust Manager Administration Guide](#) for details.
- **IP address** of the CipherTrust Manager and **port** of the KMIP interface must be accessible from GCVE.
- CipherTrust Manager recognizes only registered KMIP clients. Ensure that GCVE is registered as a KMIP client on the CipherTrust Manager. Refer to [KMIP Client Registration](#) for more details.
- Ensure that the required licenses are activated. For more details, refer to the [CipherTrust Manager documentation](#).

Steps for Integration

To integrate GCVE with the CipherTrust Manager:

1. [Configure CM on Google Cloud Platform \(GCP\)](#)
2. [Enable vSAN Encryption](#)
3. [Enable VMware KMIP Connection](#)
4. [Register KMIP Client](#)
5. [Configure VMWare VCenter](#)

Licensing

This section provides the licensing requirements for KMIP client, vSAN, and vCenter.

Condition	License Requirement
Encryption at VM Level	x1 KMIP client license for vCenter
Encryption at vSAN Level	Minimum x2 KMIP client license. x1 for vCenter+ x1 per node/ESXI host

Configuring CM on Google Cloud Platform (GCP)

To configure the CipherTrust Manager on Google Cloud Platform (GCP):

1. [Setup the Google Cloud SDK](#)
2. [Initialize The Google Cloud Project and Select Region](#)
3. [Access and Install the CipherTrust Manager Image](#)

Setup the Google Cloud SDK

To setup the Google Cloud SDK, refer to the [Google Documentation](#).

Initialize The Google Cloud SDK and Select Region

Refer the following Google documentation:

- To initialize the Google Cloud SDK
- To Select Region

Access and Install the CipherTrust Manager Image

Upon successful completion of the purchase order, you will receive an email from Thales. Select **Accept this invitation** to get access privileges to the Google image

Ensure that you have appropriate permissions to access the disk image. You can view the image(s) using the following `gsutil` command:

```
gsutil ls gs://kylo-images
```

Output:

```
gs://kylo-images/k170v-2369-20181008172807.tar.gz
gs://kylo-images/k170v-2411-20181031022613.tar.gz
```

You can also check the version mapping vs packages names using the following command:

```
gsutil cat gs://kylo-images/readme.txt
```

Output:

```
This folder contains k170v images for import into Google Cloud Platform.
This images can be imported into an account with the following command:
gcloud compute images create <image name> --source-uri <source uri>
Example: gcloud compute images create <image name> --source-uri gs://kylo-images/
k170v-4281-
20200227163646.tar.gz
1.6.0: gs://kylo-images/k170v-2689-20190305130140.tar.gz
1.7.0: gs://kylo-images/k170v-3137-20190604023249.tar.gz
1.8.0: gs://kylo-images/k170v-3506-20190927033321.tar.gz
1.9.1: gs://kylo-images/k170v-4281-20200227163646.tar.gz
1.10.0: gs://kylo-images/k170v-4610-20200619081338.tar.gz
2.0.0: gs://kylo-images/k170v-4933-20200909123423.tar.gz
2.1.0: gs://kylo-images/k170v-5170-20201118110327.tar.gz
```

To create an image, run the following `gc loud` command:

```
gcloud compute images create <image name> --source-uri <source uri>
```

Example:

```
C:\Users\username\AppData\Local\Google\Cloud SDK>gcloud compute images create k170v-1-9-1 --source-uri gs://kylo-images/k170v-4281-20200227163646.tar.gz
```

or (for CipherTrust Manager image here)

```
C:\Users\username\AppData\Local\Google\Cloud SDK>gcloud compute images create CPH-ERTRUST-MANAGER --source-uri gs://kylo-images/k170v-5170-20201118110327.tar.gz
```

Generate the CipherTrust Manager Image

To generate the CipherTrust Manager image:

1. In the GCP UI, go to **Compute Engine > Images**.
2. Under **IMAGES** menu, select the CipherTrust (k170v) image.

Note

Version number associated with the image may change.

3. Under **CREATE INSTANCE** menu, select **New VM Instance** and setup the machine configuration.

Thales recommends the following minimum system requirements:

- **Boot Disk:** 1 volume, minimum size: 30 GB for evaluation and 135 GB for production
 - **Memory:** 16 GB
 - **vCPUs:** 2
 - **NICs:** 1
4. Create an SSH Key via PuTTYGen Key Generator. SSH Key allows access to CipherTrust (K170v) CLI via SSH.
 5. Save the SSH Key, Private, and Public key in a secure location.
 6. Copy the output from the Public Key generated via PuTTYGen Key Generator, and paste it under **SSH Keys** in GCP.

7. Click **Create**.

Your CipherTrust Manager (k170v image) is now generated in the designated region.

For clustering, please follow all the above steps to generate a second instance. Further, to create a cluster and add both the instances (nodes) into it, refer to the [CipherTrust Manager Documentation](#).

After configuring CipherTrust Manager on GCP, you need to enable vSAN encryption. For more details, refer to [Enabling vSAN Encryption](#).

Enabling vSAN Encryption

To enable vSAN encryption with a connected CipherTrust Manager cluster:

1. On the CipherTrust Manager UI, configure the following:
 - a. Hostname (mandatory). For more details, refer to the [CipherTrust Manager documentation](#).
 - b. DNS servers (optional). For more details, refer to the [CipherTrust Manager documentation](#).
 - c. NTP server (optional). For more details, refer to the [Google documentation](#).
 - d. Cluster / members (recommended, with at least 2 nodes). For more details, refer to the [CipherTrust Manager documentation](#).
2. Perform the following steps to configure CipherTrust - KMIP Tenant:
 - a. In the CipherTrust UI, create the following users:
 - KMIP_admin: Created by the ROOT admin, this user is dedicated to KMIP administration purpose.
 - KMIP_user: Created by the ROOT admin, this user can operate inside the KMIP domain only, assigned by the KMIP admin.

Note

In this document we have created KMIP_admin and KMIP_user users. However, you can create user as per your convenience.

For more details on creating user, refer to the [CipherTrust Manager documentation](#).

- b. (This is an optional step.) Create domain and assign KMIP_admin under **Admins** option.

For more details on creating domain, refer to the [CipherTrust Manager documentation](#).

- c. Log on as KMIP ADMIN into the KMIP domain as <your domain>|<your user>. In this example: `KMIP_domain|KMIP_admin`.

- d. Go to **Keys & Access Management > CA**.

You can view the Local Certificate Authority created with the KMIP domain.

- e. Download the Local Certificate Authority using the contextual menu.

This domain certificate will be uploaded further, so that the KMIP interface can recognize it.



- f. Go to **Users > Assign User** and assign the KMIP user to the KMIP domain.
- g. Select the user and assign the Key Users group to it.

Repeat this step and assign the Audit group to the user, so that respective user can check the records in the CipherTrust Manager.

Finally, log out from the CipherTrust platform. Now, you are ready to prepare the VMWare client KMIP connection. For more details, refer to [Enabling VMware KMIP Connection](#).

Enabling VMware KMIP Connection

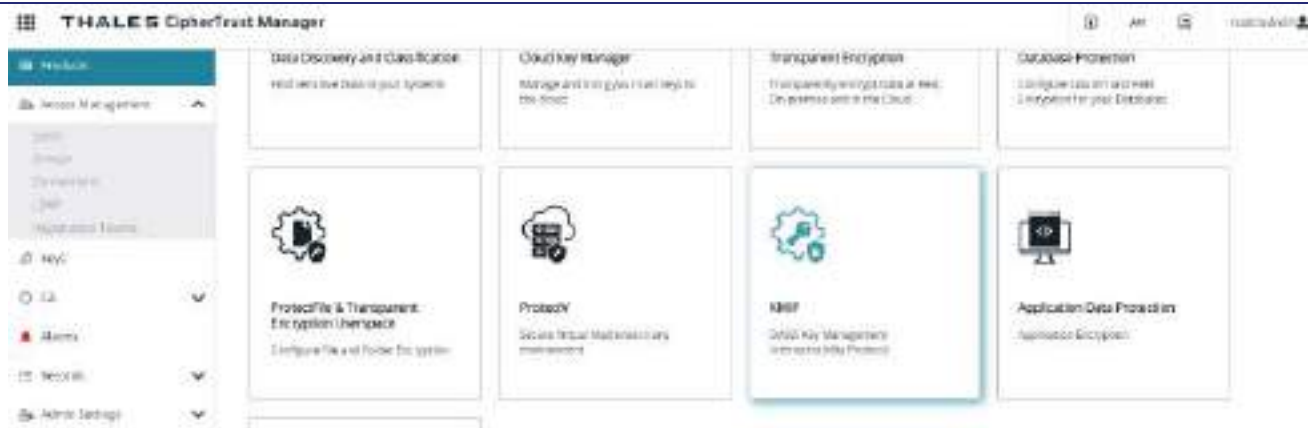
To enable the connection between VMWare client and KMIP create the following objects:

1. A client profile
2. A registration token
3. A registered client

Create a client profile

To create a new client profile:

1. Log on to the CipherTrust Manager as a domain administrator of the KMIP domain (`KMIP_domain` | `KMIP_admin`).
2. Select the **KMIP** menu from the main dashboard.



3. Click **Client Profile** from the left sidebar and click **Add Profile**.



4. Enter `KMIP_domain|KMIP_user` as the **Common Name**.

Add Profile

Profile Name*
vmware_profile

Username Location in Certificate:
CN

Certificate Details

CSR
Contents of .csr file

OR

Organization
Organizational Unit
Email
UID
City
State
Country

Common Name
KMIP_domain | KMIP_use

Device Credentials

Save Cancel

5. Click **Save**.

Create a Registration Token

To create a new registration token:

1. Log on to the CiphTrust Manager as a domain administrator of the KMIP domain (KMIP_domain | KMIP_admin).
2. Click **Registration Token** from the left sidebar and then click **New Registration Token**.



- Under **Configure Token** menu, specify **Name Prefix**.

The 'Create New Registration Token' dialog is shown at Step 01: Configure Token. The progress bar indicates the sequence: 01 Configure Token, 02 Select CA, 03 Select Profile, and 04 Create Token. The 'Name Prefix' field contains 'vmware_token'. Below it, 'Token lifetime' is set to 'unlimited' (with a green checkmark), 'Certificate Duration' is '730', and 'Client Capacity' is '100'. 'Back' and 'Select CA' buttons are at the bottom right.

- Under **Select CA** menu, select **Local CAs**.

The 'Create New Registration Token' dialog is shown at Step 02: Select CA. The progress bar shows: 01 Configure Token, 02 Select CA, 03 Select Profile, and 04 Create Token. Under 'Local CAs', the first option is selected: '/C=US/ST=MD/L=Belcamp/Q=Gemalto/OU=NA/CN=KeySecure CA for Domain KMIP_domain'. Under 'External CAs', the second option is listed: '/C=US/ST=MD/L=Belcamp/Q=Gemalto/CN=KeySecure Root CA'. 'Back' and 'Select Profile' buttons are at the bottom right.

- Under **Select Profile** menu, select the profile.



The screenshot shows a dialog box titled "Create New Registration Token" with a close button (X) in the top right corner. A progress bar at the top indicates four steps: 01 Configure Token, 02 Select CA, 03 Select Profile (highlighted in blue), and 04 Create Token. Below the progress bar, the "Client Profile" dropdown menu is open, showing "vmware_profile" as the selected option. At the bottom right, there are two buttons: "Back" and "Create Token".

- Copy the generated token from **Create Token** menu.



The screenshot shows the same dialog box, now at Step 4: Create Token. The progress bar highlights the "04 Create Token" step. Below the progress bar, a text box displays the generated token "oMkRejprTJeqOy3rVpWal" next to a "Copy" button with a clipboard icon. At the bottom right, there is a "Done" button.

Create a Registered Client

To create a registered client:

- Log on to the CipherTrust Manager as a domain administrator of the KMIP domain (`KMIP_domain | KMIP_admin`).
- From the left sidebar, click **Registered Clients** and then click **Add Client**.



3. Enter the name and token string copied in step 6 of [Create a Registration Token](#) section.

Add Client

Name

vmware_client

Registration Token

oMKRejorTUEQOy3rVpWalqhZy65W1nwWavVv2TFuFiOgk9PkkIh7ijfHILZ'

Client certificate (needed only if your registration token uses an external CA)

Client certificate

Save

Cancel

4. Click **Save Private Key** and then click **Save Certificate**.

Add Client

Name

vmware_client

Registration Token

oMKRejorTUeQOy3rVpWalqhZy65W1nwWavVv2TFuFiOgk9PkkIh7ijfHILZ'

Client certificate (needed only if your registration token uses an external CA)

Client certificate

1

2

Save Private Key

Save Certificate

Save CSR

Close



After configuring the VMware KMIP connection, you need to register the KMIP client. For more details, refer to [Registering KMIP Client](#).

Registering KMIP Client

To register KMIP client:

1. In the CipherTrust Manager UI, import the KMIP domain CA.

Note

It is necessary to import the KMIP domain CA, so that the ROOT admin can link it to the KMIP interface of the CipherTrust platform.

- Go to **Keys and Access Management > CA** and paste the CA content in the **Upload External Certificate** field.
- Click **Upload**.



- Go to the **Admin settings > Interfaces** tab and edit the KMIP interface to add/declare the KMIP domain CA.



Configure KMIP

☐ Enable hard delete ⓘ
 ☐ Auto Registration

Mode

TLS, verify client cert, user name taken from client cert, auth request is optional

Local CA for Automatic Server Certificate Generation

/C=US/ST=MD/L=Belcamp/O=Gemalto/CN=KeySecure Root CA

Local Trusted CAs

CA	
/C=US/ST=MD/L=Belcamp/O=Gemalto/CN=KeySecure Root CA	
	

External Trusted CAs

CA	
	
/C=US/ST=MD/L=Belcamp/O=Gemalto/OU=NA/CN=KeySecure CA for Domain KMIP_domain	

Download Certificate

[Download Current Certificate](#)

Upload Certificate



[Update](#) [Cancel](#)

At this stage, CipherTrust Manager is ready to accept KMIP connection from the VMWare client using Private Key & Certificate.

It is recommended to restart the CipherTrust services to take into account the new CA. To do so:

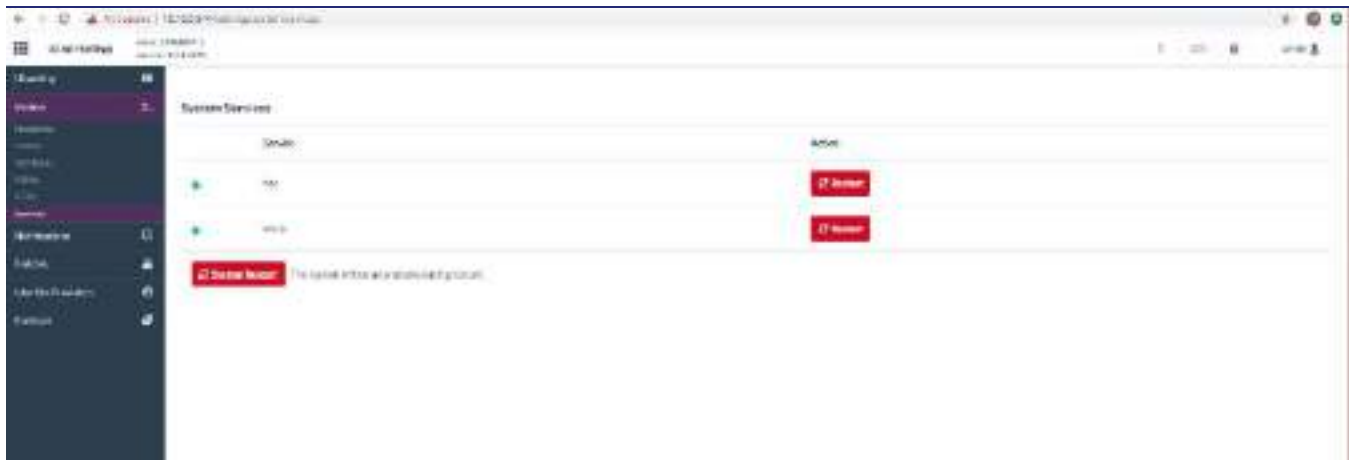
Using ksctl:

Execute the following command:

```
./ksctl services restart
```

Using CipherTrust Manager UI:

Go to **Admin Settings > System > Services** and restart the services.



After registering the KMIP client, you need to configure VMware. For more details, refer to [Configuring VMWare VCenter](#).

Configuring VMWare VCenter

Prerequisite

Ensure that:

- VCenter is up and running.
- CipherTrust Manager cluster is accessible and in good health. You can check it as admin user from **System > Cluster**.

Note

If this is not the case, you should be notified by an alarm under **Keys & Access Management > Alarms**

Declare a KMS cluster in VMware

To declare a KMS in VMware:

1. Go to **Security > Key Providers > Configure** tab.
2. Click **ADD STANDARD KEY PROVIDER**.
3. Specify the following information in **Edit Standard Key Provider** window:
 - a. Name for the cluster
 - b. Host/IP address and port for each KMS node

4. Click **EDIT KEY PROVIDER**.
5. Under **Establish Trust**, click **Make VCenter Trust KMS**.

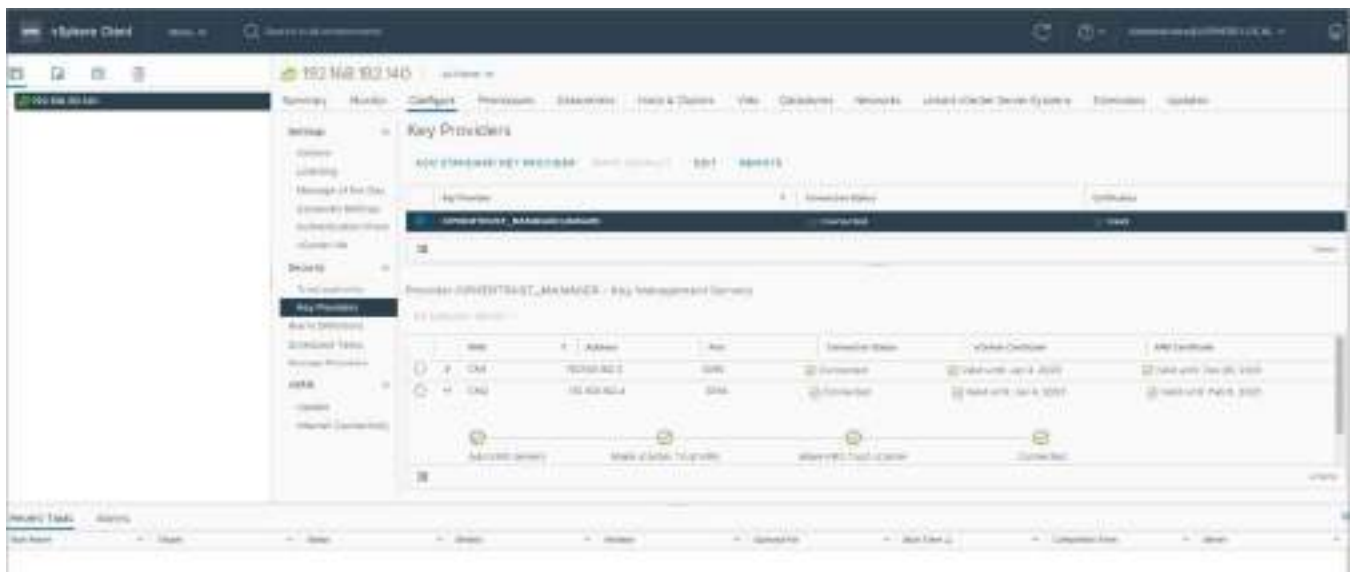
A popup window is displayed, click **TRUST**.
6. click **TRUST VCENTER** button.
7. From the left sidebar, under **Choose a Method**, select the **KMS certificate and private key** option.
8. From the left sidebar, under **Upload KMS Credentials**, either upload or copy and paste the KMS Certificate and Private Key.

Note

KMS Certificate and Private Key is downloaded previously. Refer to the [Create a Registered Client](#) section.

9. Click **ESTABLISH TRUST**.

At this stage, ensure that the setup looks similar to the following image:



Moreover, you can also check the logs in CipherTrust Manager UI under **Records** tab.



CipherTrust Manager

Version: v2.22 (latest)

4 February 2026

Contents

Data Discovery and Classification (DDC)	4
DDC overview	5
Concepts	9
User groups	12
Locations	14
Information types	15
Classification profiles	26
Scans	32
Reports	49
Agents	63
Similarity search	68
Discovering sensitive information	72
Local storage	73
Big data	80
Hadoop	80
Teradata	86
Network storage	93
NFS Share	93
SMB/CIFS Share	100
Server	107
Exchange Server	107
Sharepoint Server	118
Clouds	128
AWS S3	129
Azure Blobs	139
Azure Table	147
Google Workspace: GDrive	155
Google Workspace: Gmail	165
Office 365: Exchange Online	177
Office 365: OneDrive for Business	189
Office 365: SharePoint Online	201
Salesforce	213
Databases	228
IBM Db2	228
MongoDB	236
Microsoft SQL	243
MySQL	251
Oracle	258
PostgreSQL	266
SAP HANA	274
Logging	281
Operations	284

Troubleshooting	295
Appendix.	322
Deployment Security	341

Data Discovery and Classification (DDC)

This document is intended for CipherTrust Data Discovery and Classification (DDC) administrators responsible for configuring Data Stores and running Scans and Reports.

All products manufactured and distributed by Thales Group are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.

It is assumed that the users of this document are proficient with data security concepts.

Trademarks

- Thales CipherTrust Data Discovery and Classification is powered by Groundlabs.
- DDC ML cloud-enabled solution is powered by Thales.

What's in this document

This guide explains data discovery concepts such as Locations, Classification Profiles, Data Stores and Scans. The document also explains how to generate scan-based reports on the discovered data. Finally, the document describes how to read generated Reports.

Prerequisite

- It is assumed, for this document, that you have already configured the CipherTrust Manager appliance. Refer to the [CipherTrust Manager product documentation](#) for instructions.
- The next step is to activate and install the DDC license. Refer to [Licensing](#) for details.
- After the license is installed, you can configure DDC Agents. Refer to the [Data Discovery and Classification Deployment Guide](#) specific to your platform for details. Installation of the CTE Agents is required for protecting directories and files stored on clients.

Organization

This document contains the following sections:

- [DDC overview](#): Provides a high-level overview of the DDC solution.
- [Concepts](#): Describes DDC concepts such as Information Types, Classification Profile, Data Stores, Scans and Reports.
 - [User groups](#): Describes all predefined groups of DDC users with their rights to use various product features.
 - [Locations](#): Describes how to add a Location.

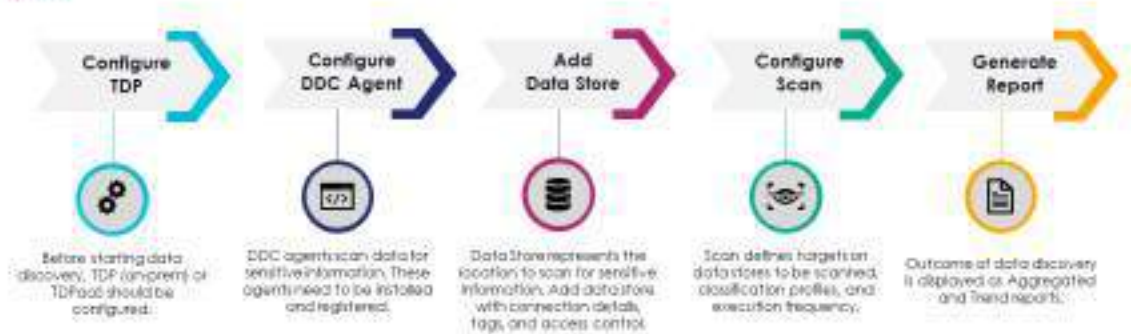
-
- **Information types:** Describes how to view, customize and edit a Information Type (infotype).
 - **Classification profiles:** Describes how to view, add and edit a Classification Profile.
 - **Scans:** Describes how to view and add a Scan.
 - **Reports:** Describes how to view and add a Report.
 - **Agents:** Describes how to add agents and view the agent list.
 - **Similarity search:** Describes how to run similarity search on DDC ML infotypes.
 - **Discovering sensitive information:** Describes how to view, add, edit and scan different Data Store.
 - **Logging:** Describes most common log messages from DDC.
 - **Operations:** Describes the operations that the DDC Administrator performs on the CipherTrust Manager. In some situations, you must reconfigure DDC Agents with a new hostname, IP address, Agent Memory, CPU, or throughput.
 - **Troubleshooting:** Describes how to handle the issues that you might face when using DDC with the CipherTrust Manager.
 - **Appendix:** Additional useful information and tools related to system administration, such as system error messages, handy commands.
 - **Deployment security:** Information on system messages related to security of the deployment and tips on how to increase it.

DDC overview

This section provides an overview of the Thales CipherTrust Data Discovery and Classification (DDC) solution.

Workflow

This diagram provides a high-level flow of the DDC and DDC ML solution.

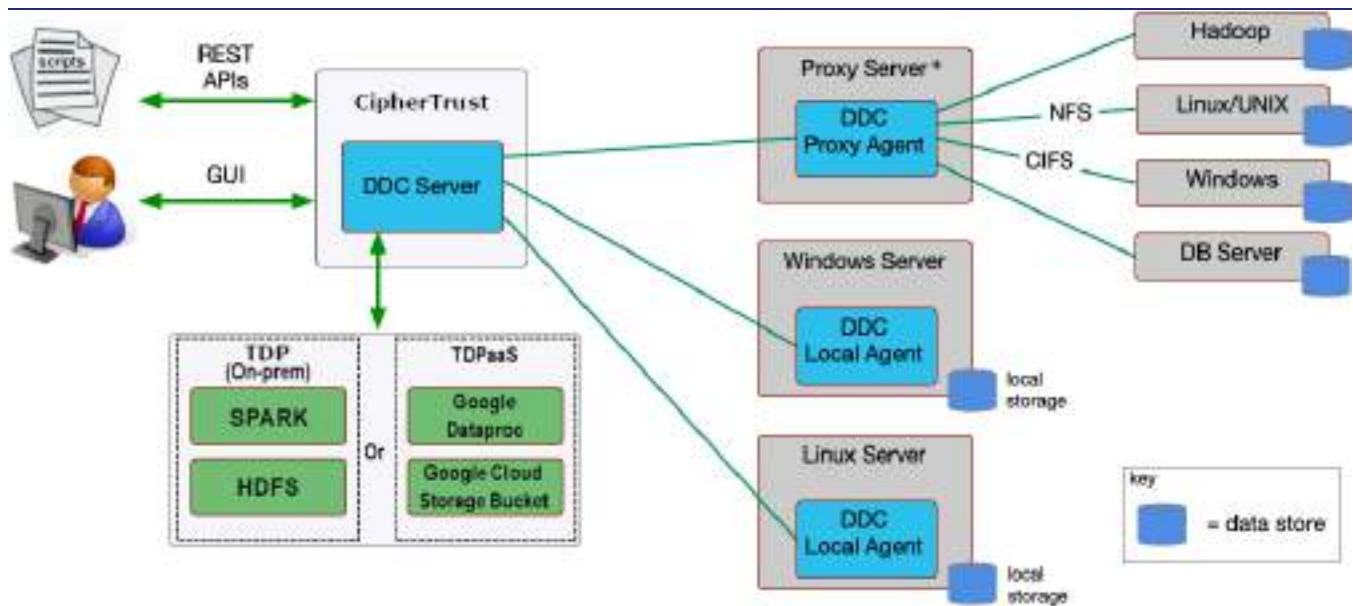
DDC**DDC ML**

Solution architecture

DDC solution operates using a standard DDC architecture and a DDC ML architecture. The two architectures differ in types of components included and in the way these components are configured and interact with one another.

DDC architecture

The following image shows the standard DDC architecture.



Architecture components

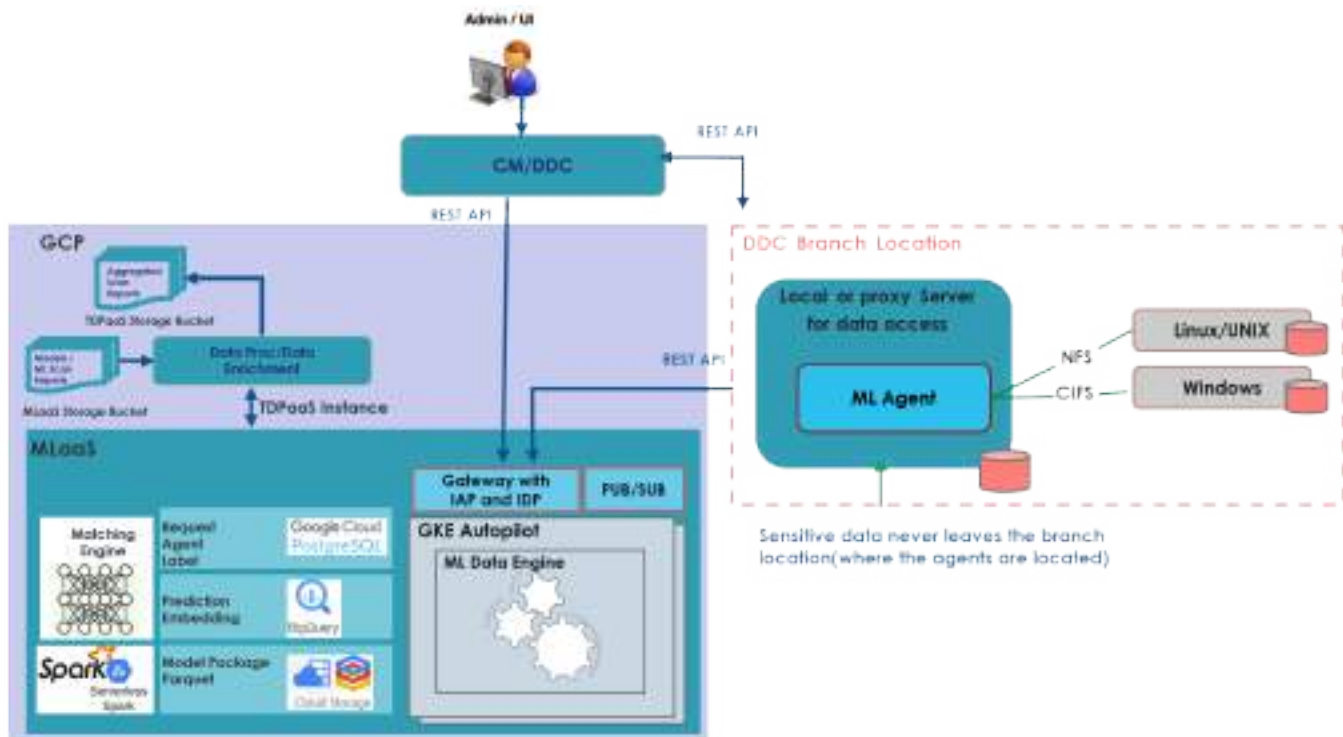
The following table outlines different architecture components along with their respective roles:

Component	Description
CipherTrust Manager, DDC Server	At the heart of the Data Discovery and Classification solution is CipherTrust Manager on which runs the DDC Server. It is from here that users interact with the DDC GUI or use the DDC APIs to create classification profiles, add data stores, launch scans and generate reports.
TDP (On-prem): Hadoop, Spark, HDFS	TDP (On-prem) is configured to work with Hadoop data clusters. DDC uses Hadoop to generate reports from scans and to store their results (report data). DDC can directly query HDFS but it requires Spark to interface with Hadoop's HBase. For installing TDP (On-prem), see Thales Data Platform Deployment Guide .
TDPaaS	TDPaaS is a server-less, cloud-based service used for storing the scan and report data. It is a SaaS component that offers an alternative to Hadoop services of on-prem TDP. For configuring DDC to use TDPaaS, see Configuring TDPaaS .
DDC Agent	DDC Agents perform the actual scanning jobs and report the results back to the DDC Server for analysis and processing. DDC supports two types of Agent configurations: • Local Agents: Installed and configured directly on the machine that contains sensitive data. • Proxy Agents: Installed and configured on a proxy machine that is used to scan sensitive data on other machines. For detailed information about agents, see Agents.

Component	Description
Data Stores	A data store is where the data actually resides. It can be a file server, a database, or a Hadoop cluster. For more information, see Discovering Sensitive Information .

DDC ML architecture

The following image shows the DDC ML architecture.



Architecture components

The following table outlines different architecture components along with their respective roles:

Component	Description
CipherTrust Manager, DDC Server	At the heart of the Data Discovery and Classification solution is CipherTrust Manager on which runs the DDC Server. It is from here that users interact with the DDC GUI or use the DDC APIs to create classification profiles, add data stores, launch scans and generate reports.
TDPaaS	TDPaaS is a server-less, cloud-based service used for storing the scan and report data. It is a SaaS component that offers an alternative to Hadoop services of on-prem TDP. For configuring DDC to use TDPaaS, see Configuring TDPaaS .

Component	Description
MLaaS	MLaaS (Machine Learning as a Service) is a multi-tenant service on Google Cloud Platform that is completely managed. It offers pre-trained machine learning models to complete tasks like text embedding, document classification, and named entity recognition. MLaaS works with CipherTrust Manager to process scan, index, and search requests. It takes the scan metadata information produced by the DDC ML agent and uploads it to Google Cloud. It then creates an inverted index that connects embedding and document metadata, allowing search for similar items using the k-nearest neighbors algorithm. For configuring DDC to use MLaaS, see Configuring MLaaS . Additionally, see Similarity search for details on semantic features powered by MLaaS.
DDC ML Agent	A unified agent with machine learning capabilities that also encompasses the features of the standard DDC agent. DDC ML agents perform the actual scanning jobs and report the results back to the DDC Server for analysis and processing. For installing DDC ML agent, see DDC ML Agent .
Data Stores	A data store is where the data actually resides. Currently, DDC ML supports only Local and Network data stores. For more information see Discovering Sensitive Information .

Concepts

Location

A location specifies a site where the file servers, databases, and data centers that contain data to scan are located. Locations are used to indicate where different data stores are physically located. For more information see [Locations](#).

Sensitivity levels

A sensitivity level defines how sensitive the data is. Sensitivity levels are required in creating classification profiles and data stores. Prebuilt sensitivity levels are:

- **None:** The sensitivity level for such data has not yet been specified.
- **Public:** Specifies the least sensitive data with no specific need for data security. Such data can be shared with anybody.
- **Internal:** Specifies the data with low sensitivity. Exposure of such data may not affect an organization, but is not meant for public disclosure.
- **Private:** Specifies that the data is personal. Such data should be protected from public viewing.

-
- **Restricted:** Specifies highly sensitive data, for example, customer's personal data and trade secrets etc. This type of data requires the best possible data security. Disclosure of such data can lead to severe financial and legal consequences for an organization. Businesses must prioritize remediation efforts related to this type of data.

Information type

An information type (or infotype) categorizes data to look for during a scan. A large number of predefined information types are available to better categorize the data. For more information see [Information Types](#).

Tag

A tag helps group data together. Tags are used to filter data for generating reports. They can be specified when creating data stores and classification profiles.

Data Discovery and Classification includes a number of predefined Tags, but also provides the ability to create custom Tags when creating data stores and classification profiles.

Predefined tags

The predefined Tags are APA, APPI, CCPA, FINANCIAL, GDPR, GDPR-FINANCIAL, GDPR-HEALTHCARE, GDPR-ID, GDPR-PII, HEALTH, HIPAA, KVKK, LEGAL, LGPD, NYDFS, PCI, PERSONAL, PHI, PII, SHIELD and UK-GDPR.

Classification profile

A classification profile defines what kind of sensitive information to search for during a scan. It includes information such as a sensitivity level, information types, and tags. Classification profiles can be created based on predefined templates or custom templates. For more information see [Classification Profiles](#).

Data object

A file, a database table, and a BLOB in a database table as stored in a data store are called Data Objects.

Sensitive data object

A data object that contains any data match is called a Sensitive Data Object.

Data Match

A concrete instance of any of the infotypes is called a Data Match.

Risk

A risk is the presence of a sensitive data object in a data store. The risk is calculated per the data object and data store. The risk is directly related with the matches found in the data object or data store.

Scan

A scan is an entity that helps in scanning data stores. Each scan specifies the location to scan and what to look for during scanning. Findings of scans can be used to generate reports for different purposes. Scans can be either run manually (any time) or scheduled to run and stop at a specified time.

Note

- DDC supports partial database scanning. To enable this, you need to configure the number of rows to be scanned. When you run a database scan, the scan results included data from the specified number of rows only.
- If you don't specify the number of rows to scan, the entire databases will be scanned.

For more information see [Scans](#).

Labels

Labels in Data Discovery Classification (DDC) are used to categorize and control standard agents. Labels are tags that highlight an agent's specific features, like its ability to scan certain types of data stores or its performance capabilities. An agent can have several labels. Agent labels can only be created in the **Labels** section inside the agents. See [Managing agent labels](#) to learn how to add or edit agent labels.

Labels have following characteristics:

- **Customizable:** There are no pre-defined labels. You can create them based on your requirements. For example, you could label an agent "Paris Data Center" or "AWS" to ensure DDC selects an agent that is co-located with the data store, providing low latency and high bandwidth access. Alternatively, you might use labels like "Critical" for agents dedicated to scanning highly sensitive data.
- **Immutable after creation:** Once a custom label is created, you can't modify it. However, you can always assign new or existing labels to an agent.

Agent labels vs data store labels:

It's important to understand the relationship between agent labels and data store labels:

- **Agent Labels:** These represent an agent's capabilities.
- **Data Store Labels:** These indicate the capabilities required for any agent to scan that specific data store.

For an agent to be selected to scan a particular data store, it must have all the labels defined for that data store. It can, however, have additional labels. See [Automatic agent selection](#) for more information.

You can also use labels to ensure that only the right agents handle your sensitive data, increasing DDC deployment's security. For more information, see [Mitigating security risks](#).

Encryption keys

DDC uses AES256 encryption to protect sensitive data. DDC creates a number of encryption keys that are stored in the CipherTrust Manager. You can find these DDC keys in the **Keys & Access Management** application in the CipherTrust Manager.

- Four encryption keys to protect the Hadoop configuration before storing it inside the DDC database. Each key is used to protect one configuration parameter (HDFS Server, and HDFS credentials).
 - **Encryption key format:** `citrus-<UUID>`. Example:
`citrus-6e0cb668-3a3d-4f2c-8687-17092b83b41b`.
- One encryption key for each data store. Each key is used to encrypt data store credentials before storing it in the DDC database, and encrypt scan results of that data store before storing them in HDFS. Datastore encryption key formats:
 - **For CipherTrust Manager version 2.15 or earlier:** `d<UUID>`. Example:
`d8b2d8404-c9ae-4a34-800a-01258dfaa383`.
 - **For CipherTrust Manager version 2.16 or later:** `DDC-Data-Store-<UUID>`. Example: `DDC-Data-Store-8b2d8404-c9ae-4a34-800a-01258dfaa383`.
- One encryption key for each scan. Each key is used to encrypt the scan data before storing it in HDFS. Scan encryption key formats:
 - **For CipherTrust Manager version 2.15 or earlier:** `s<UUID>`. Example:
`s14912791-bed5-4e73-b733-6a36ecfe338f`.
 - **For CipherTrust Manager version 2.16 or later:** `DDC-Scan-<UUID>`. Example: `DDC-Scan-14912791-bed5-4e73-b733-6a36ecfe338f`.

Warning

Never delete any encryption key. Otherwise, DDC will not be able to process the related datastores and scans properly.

User groups

System predefined groups

DDC has different kinds of users with different responsibilities in administering and using the system. A number of predefined groups are included to ensure that users are granted minimal permissions needed to perform their tasks while ensuring flexibility to meet security requirements across industries.

The table below lists all predefined groups with their rights to use various DDC features. **R/W** in a cell means that the user has **view** and **edit** rights to this aspect of the product. **R** means that the user has only **view** rights.

	Agents	Data Stores	Scans	Profiles	Reports	Sensitivity Levels	Branch Locations	Infotypes	Hadoop Configuration	Similarity Search
Admin ¹	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDC Admins ^{2, 7}	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDC Store Viewers ⁸	R	R				R	R			
DDC Store Admins ⁷	R/W	R/W				R	R/W			
DDC Scan Viewers ⁶			R					R		
DDC Scan Admins		R	R/W	R			R	R		
DDC Profile Viewers				R				R		
DDC Profile Admins			R	R/W		R		R		
DDC Report Admins		R	R	R	R/W	R	R	R		
DDC Full Report Admins		R	R	R	R/W	R	R	R		
DDC Infotype Viewers								R		
DDC Infotype Admins								R/W		
DDC L3 Support									R/W	
DDC Full Similarity Searches Admins										R
DDC Similarity Searches Viewers		R						R		R
DDC Similarity Searches Admins		R						R		R/W
DDC V2 Admins	R/W									
DDC V2 Clients	R/W									

[1] **Admins** can see their own and other users' reports. Admins can also decrypt scan packages from the Hadoop database.

[2] **DDC Admins** can only see their own reports.

[3] **DDC Report Admins** can only see their own reports.

[4] **DDC Full Report Admins** can only see their own reports.

[5] The difference between **Report Admins** and **Full Report Admins** is that **Full Report Admins** do not need access to specific user defined groups to be able to view or generate reports that use data stores restricted to user defined groups. For more information, see [User Defined Groups](#) below.

[6] Scan Viewers are allowed to run scans.

[7] Can view the list of the agents, and activate or deactivate the "Local Storage Only" option.

[8] Can view the list of the agents, but are not allowed to activate or deactivate the "Local Storage Only" option.

[9] **DDC Store Viewers**, **DDC Store Admins**, and **DDC L3 Support** do not have access to custom infotypes.

The users belonging to the "L3 Support" group are Data Discovery and Classification support administrators. These users can help identify and troubleshoot issues you may encounter when using DDC. They can also can also decrypt scan packages from the Hadoop database.

User defined groups

Apart from system predefined groups, CipherTrust Manager also allows you to create [User Defined Groups](#). User defined groups can be used to prevent certain users from viewing sensitive information in reports. These groups are defined by the application administrators, please refer to [Managing User Defined Groups](#) for further information. In DDC, they are applicable when creating a data store, in the **Data Store** screen when you are granting access to selected groups. For details, see the **ACCESS: Selected group/s** setting in the "Tags & Access Control" section of any data store in [Discovering Sensitive Information](#).

In other words, a data store that is restricted to a specific user defined group is visible to all the groups with permissions to see data stores, the same goes for scans. However, a user without a permission to see a data store which is restricted to a group, but with a permission to create and generate reports, will not be able to generate reports for those data stores. For this user, the scan executions will not be visible in the **New Report** screen.

Note

The only users that do not have to belong to a specific user defined group to be able to see reports for all the data stores are **Full Report Admins** and **Admins**.

For example, if a user created a report that has Data Store "DS1" but the data store is restricted to a specific group, that user will see the report template, but when he tries to access the report he will get an "insufficient permissions" error.

Locations

You manage locations through the **Locations** page, which is accessed by clicking the **Settings > Locations** link in the **Data Discovery** sidebar on the left.

From the **Locations** page you can:

- View all currently available locations. See [Viewing locations](#).
- Create a new location. See [Adding locations](#).

Viewing locations

The **Locations** page lists available locations. This view also shows the total number of existing locations. The **Locations** page shows the following details:

Item	Description
Location Name	Name of the location.
Country	Name of the country.
State/Province	Name of the state/province. This field is applicable to the United States. For other countries, the field is unavailable, indicated by N/A.
City	Name of the city.
Description	Description of the location.

Tip

Use the **Search** text box to filter locations. Search results display locations that contain the specified text in their names.

Adding locations

Adding a location requires specifying the country and city of the location.

To add a location:

1. Click **+ Add Location**.
2. In the **Add Location** screen, enter the following details:
 - **Location Name** - Specify a unique name for the location. The name must be longer than two characters and up to 64 characters. This field is mandatory.
 - **Country** - Select the country from the drop-down list. This field is mandatory. **State/Province** | Select the state/province from the drop-down list. This field is applicable to the United States. For other countries, the field is unavailable.
 - **City** - Specify name of the city. This field is mandatory.
 - **Description** - Describe the location (up to 250 characters).
3. Click **Save**.

The newly created location appears on the Locations page. By default, locations are sorted in alphabetic order by name.

Information types

An information type (infotype) categorizes data to look for during a scan. A large number of prebuilt information types are available to better categorize the data.

Different regions and countries can have different regulatory requirements, so these information types are

categorized based on geographical regions. These regions are Global, Africa, Americas, Asia, Europe, and Oceania. The information types can be further categorized into:

- **Financial:** Financial data such as credit card numbers and bank account details.
- **Personal Data:** Personal data such as age, gender, race, and religion.
- **Medical:** Medical data such as history of medical problems and disabilities.
- **National ID:** National identity documents such as Social Security Number (SSN).
- **Cryptographical:** Cryptographic keys such as AWS cloud keys.

Refer to [Information types](#) and [DDC ML information types](#) for a list of all available predefined information types.

DDC also allows you to create custom information types. For more information, see [Creating custom infotypes](#).

Creating custom infotypes

You can create a custom information type, if you require one.

To create custom infotype, click **Settings > Infotypes** in the sidebar on the left. The **Infotype** page is displayed.

1. Click the **+Add Infotype** button. The **Add Infotype** screen is displayed.
 - a. In the **General Info** tab, provide the following information:
 - **Name:** Choose a name for your infotype.
 - **Category:** Select a category to which your infotype belongs (Financial, Personal Data, Medical, National, or Cryptographical).
 - **Family:** Select a family for your infotype. A family is a subcategory inside the **Category** and the choice of options depends on what you selected in the **Category** menu. The following families are available inside their corresponding categories:
 - **Financial:** Credit/Debit Cards; Bank Account Info
 - **Medical:** Patient Health Data
 - **National ID:** Personal Identification
 - **Personal Data:** Email addresses; Login credentials; Card Number; Ethnicity; License Number; Roll Number; Passport Number; Date Of Birth; MAC Address; Mailing Address; Telephone Number; Gender; Religion; IP Address; Phone Number; Name
 - **Cryptographical:** AWS Key, RSA Key
 - **Weight:** Set the risk weight for the custom infotype, which will be used to calculate the risk formula per Data Object. Minimum weight is 1 and maximum is 10. That weight can be modified later, if needed, by editing the custom infotype in the **Infotypes** screen.

Note

- You can set a weight only for a custom infotype; built-in and ML infotypes have a pre-defined weight that cannot be changed.
- If you update a custom weight, you need to run a new scan and generate a new report to see the effects of the update. Historical reports will not be updated.

- **Region:** Select the region for your infotype (Global, Africa, Americas, Asia, Europe, and Oceania).

Click **Next** to go to **Infotype Definition** screen.

- In the **Infotype Definition** tab, you configure the rules for your new information type. You can start by configuring the rules on the **Simple View** tab, and then you can view or edit these rules as translated into an expression of the internal language of the Data Discovery and Classification engine on the **Expert View** tab. Each new expression is validated when you select **SAVE**, and errors, if any, are displayed.

Tip

Alternatively, you can directly click the **Expert View** tab ignoring the **Simple View**. If you add any expression not supported by the **Simple View** in the **RULE** text area, this tab will be disabled. For an extended guide on the expressions available in this internal language refer to [DDC GLASS Reference](#).

To configure the rules for your new information type, click to expand the **Add Rules** menu in the **Simple View** tab and select one of the following types:

- **Character:** Search for one or more specific characters as specified in the **Select Rule** menu. If the character is found, the location will be returned as a match. For a list of available character type rules, refer to [Character type rules](#).

Use the **From** and **To**, controls to set the number of consecutive occurrences of the selected character.

- **Phrase:** Search for a specific pattern as defined in the **Phrase** textbox (in layman terms, it is used to look for specific words). Searching for phrases is **case insensitive**.
- **Built-In:** Pre-defined infotypes can be used in combination with other types (Character or Phrase). The complete list of built-in information types is available in the [Appendix](#).

Note

Currently, you cannot use the Indian Passport Number built-in infotype for creating custom infotypes.

Use the **Apply** button to complete your selection. The selection is displayed in the list of defined infotype rules. You can remove it from the list of rules by clicking the **Remove** link on its right.

You can use each of these types on their own, or combine them to form a more complex rule, involving multiple types in various configurations. See [Examples of custom infotypes](#) for some examples.

Note

Due to a known limitation, it is currently not possible to have two built-in infotypes, one after the other. They must be separated by something, such as another type infotype, or even a plain space. For example:

"american express" " " "english name"

Due to a known limitation, when adding a range of characters, all the possible combinations inside the range will appear as matches. For example, for a range of numbers from 1 to 4, when in a document you have a sequence 1234, the search will yield the following matches:

"1", "12", "123" and "1234"

When you introduce spaces at the beginning or the end of the phrase, the spaces are removed. Also, when you introduce more than one space between words, only one space is considered.

- Click **Save** to save your new infotype. Your new information type has been added and is now listed in the **Infotypes** screen, and marked 'Custom' in the **Type** column.

Character type rules

Specific predefined characters are used to create custom infotypes using character based rules. They are explained below:

Rule	Expert View Keyword	Match
Space	SPACE	Any white-space character.
Horizontal space	HSPACE	Tab characters and all Unicode "space separator" characters.
Vertical space	VSPACE	All Unicode "line break" characters.

Rule	Expert View Keyword	Match
Any	BYTE	Wildcard character that will match any character.
Alphanumeric	ALNUM	ASCII numerical characters and letters.
Alphabet	LETTER	ASCII alphabet characters.
Digit	DIGIT	ASCII numerical characters.
Printable	PRINTABLE	Any printable character.
Printable ASCII only	PRINTABLEASCII	Any printable ASCII character, including horizontal and vertical white-space characters.
Printable non-alphabet	PRINTABLENONALPHA	Printable ASCII characters, excluding alphabet characters and including horizontal and vertical white-space characters.
Printable non-alphanumeric	PRINTABLENONALNUM	Printable ASCII characters, excluding alphanumeric characters and including horizontal and vertical white-space characters.
Graphic	GRAPHIC	Any ASCII character that is not white-space or control character.
Same line	SAMELINE	Any printable ASCII character, including horizontal white-space characters but excluding vertical white-space characters.
Non-alphanumeric	NONALNUM	Symbols that are neither a number nor a letter; e.g. apostrophes ', parentheses (), brackets [], hyphens -, periods ., and commas ,.
Non-alphabet	NONALPHA	Any non-alphabet characters; e.g. ~ ` ! @ # \$ % ^ & * () _ - + =
Non-digit	NONDIGIT	Any non-numerical character.

Examples of custom infotypes

Example 1

You want to search for a "Driver License Number" from Illinois, whose format is **M532-4218-1341**. You would then create the following rule:

Character Alphabet From 1 to 1 Times

Character Digit From 3 to 3 Times

Phrase -

Character Digit From 4 to 4 Times

Phrase -**Character Digit From 4 to 4 Times**

The above example will have the following syntax in the **Expert View**:

```
RANGE LETTER TIMES 1-1
THEN RANGE DIGIT TIMES 3-3
THEN WORD NOCASE '-'
THEN RANGE DIGIT TIMES 4-4
THEN WORD NOCASE '-'
THEN RANGE DIGIT TIMES 4-4
```

In this rule, the expression "**Character** Alphabet **From 1 to 1 Times**" means that you only expect one alphabetic character, the expression "**Character** Digit **From 3 to 3 Times**" means that you expect exactly three digits, and the expression "**Phrase -**" means that you expect to find a hyphen (-) in the sequence.

Example 2

You want to search for a name and last name separated by a number of spaces between 1 and 3. To that end you would create the following rule:

Phrase John**Character Space From 1 to 3****Phrase Gordon**

The above example will have the following syntax in the **Expert View**:

```
WORD NOCASE 'John'
THEN RANGE SPACE TIMES 1-3
THEN WORD NOCASE 'Gordon'
```

This rule will allow you to search for a combination of **John** and **Gordon** with one through three spaces between them. By comparison, using the rule **PhraseJohn Gordon** will only allow you to search for a combination **John Gordon**, with only one space. Any additional spaces in the phrase will be truncated.

Example 3

You want to search for the Spanish NIE (foreigner's identity number) preceded by the phrase **NIE:** and a number of spaces between 0 and 5. For example, **NIE X8691474Q**.

Phrase NIE**Character Space From 0 to 5****Built-in Spanish NIE**

The above example will have the following syntax in the **Expert View**:

```
INCLUDE 'DEFINE_NID'
WORD NOCASE 'NIE'
```

```
THEN RANGE SPACE TIMES 0-5
THEN REFER 'NID_SPAIN_NIE'
```

This rule will find both **NIE** X8691474Q and **nie** x8691474q since searching (regardless of the type - Character, Phrase, or Built-in) is case insensitive.

Search precision

Search precision allows you to change the precision level (in other words, the confidence type) for certain infotypes. Search precision is supported for built-in infotypes. [Infotypes that support search precision](#) section lists all built-in infotypes supporting search precision.

Modifying search precision for an infotype

You can change the precision for a infotypes with which you want to scan, before creating a scan. This can be achieved from the **Infotypes** screen.

1. To access it, on the DDC main page click **Classification Profiles** then **Infotypes** in the sidebar on the left.
2. To change the precision value for an infotype, click the toggle switch in the **Search Precision** column - select **LOW** or **HIGH**.
 - **HIGH** - higher search confidence (i.e. precision), in other words, more robust search.
 - **LOW** - lower search confidence, in other words, more relaxed search.

In case an infotype does not support search precision, it will show **N/A** in the **Search Precision** column.

Creating scans with modified search precision

To create a scan with an infotype with a modified search precision, you use the same steps as for creating any scan for any data store. Just make sure that you select the classification profile that uses the infotype that you modified. For more information, see [Adding scans](#).

Reports with modified search precision

- **Scan Trend Report:** When a scan is generated using search precision, the scan trend report will display an asterisk "*" for the scans when the info type Search Precision value was changed.
- **Aggregated Report:** Data objects in aggregated reports will display an asterisk "*" after the infotypes detected where search precision was modified (this only applies to low search precision, as high precision setting will not display asterisks).

For more information on these reports, see [Report details](#).

Infotypes that support search precision

Below, you have a list of all infotypes that support search precision. Within brackets you have the infotype IDs as coded internally by DDC, for example to be used with the REST API.

Note

The API will apparently allow you to change the search precision level for **any** infotype, however, this will not work if that infotype does not support search precision.

- Australian Business Number (BANK_AUSTRALIA_ABN)
- Australian Company Number (BANK_AUSTRALIA_ACN)
- United Kingdom VAT Number (BANK_UNITED_KINGDOM_VAT)
- Australian Bank Account Number (BANK_AUSTRALIA_BANK_ACCOUNT)
- Canadian Bank Account Number (BANK_CANADA_BANK_ACCOUNT)
- Japanese Bank Account Number (BANK_JAPAN_BANK_ACCOUNT)
- South Korean Corporation Registration Number (BANK_SOUTH_KOREA_CRN)
- South Korean Taxpayer Identification Number (BANK_SOUTH_KOREA_TIN)
- South Korean NH Bank Account Number (BANK_SOUTH_KOREA_BANK_ACCOUNT_NH)
- South Korean KB Bank Account Number (BANK_SOUTH_KOREA_BANK_ACCOUNT_KB)
- South Korean KEB Hana Bank Account Number (BANK_SOUTH_KOREA_BANK_ACCOUNT_KEB_HANA)
- South Korean Shinhan Bank Account Number (BANK_SOUTH_KOREA_BANK_ACCOUNT_SHINHAN)
- South Korean Gwangju Bank Account Number (BANK_SOUTH_KOREA_BANK_ACCOUNT_GWANGJU)
- South Korean Jeju Bank Account Number (BANK_SOUTH_KOREA_BANK_ACCOUNT_JEJU)
- South Korean Jeonbuk Bank Account Number (BANK_SOUTH_KOREA_BANK_ACCOUNT_JEONBUK)
- Australian Tax File Number (NID_AUSTRALIA_TFN)
- Austrian SSN (NID_AUSTRIA_SSN)
- Austrian Personalausweis (NID_AUSTRIA_PERSONALAUSSWEIS)
- Belgian eID (NID_BELGIUM_EID)
- Belgian National Number (NID_BELGIUM_NN)
- Brazilian CPF (NID_BRAZIL_CPF)
- Brazilian Registro Geral (NID_BRAZIL_RG)

-
- Bulgarian EGN (NID_BULGARIA_EGN)
 - Canadian Social Insurance Number (NID_CANADA_SIN)
 - Chilean RUN (NID_CHILE_RUN)
 - Croatian OIB (NID_CROATIA_OIB)
 - Czech Republic RC (NID_CZECH_RC)
 - Danish CPR (NID_DENMARK_CPR)
 - French Carte Vitale (NID_FRANCE_CV)
 - French INSEE (NID_FRANCE_INSEE)
 - French CNI (NID_FRANCE_CNI)
 - Greek AFM (NID_GREECE_AFM)
 - German Personalausweis (NID_GERMAN_PERSONALAUSSWEIS)
 - Hong Kong ID (NID_HONGKONG_HKID)
 - Hungarian Personal ID (NID_HUNGARY_PIN)
 - Icelandic Kennitala (NID_ICELAND_ID)
 - Indian Aadhaar Number (NID_INDIA_AADHAAR)
 - Iranian National Identification Number (NID_IRAN_NID)
 - Irish Personal Public Service Number (NID_IRELAND_PPS)
 - Israeli Identity Number (NID_ISRAEL_ID)
 - Italian Codice Fiscale (NID_ITALY_CF)
 - Italian CARTA D'IDENTITÀ (NID_ITALY_CID)
 - Latvian Personas Kods (NID_LATVIA_PK)
 - Luxembourg ID (NID_LUXEMBOURG_ID)
 - Malaysian NRIC (NID_MALAYSIA_NRIC)
 - Maltese eID (NID_MALTA_EID)
 - Dutch Burgerservicenummer (NID_NETHERLANDS_BSN)
 - New Zealand Inland Revenue Number (NID_NEW_ZEALAND_IRD)
-

-
- Portuguese Citizen's Card (NID_PORTUGAL_CC)
 - Portuguese Fiscal Number (NID_PORTUGAL_FN)
 - Portuguese Identity Number (NID_PORTUGAL_IN)
 - Romanian Identity Card (NID_ROMANIA_IDC)
 - Slovakian RC (NID_SLOVAKIA_RC)
 - South African Identity Number (NID_SOUTH_AFRICA_ID)
 - South Korean RRN (NID_SOUTH_KOREA_RRN)
 - South Korean Foreigner Number (NID_SOUTH_KOREA_FRN)
 - Spanish DNI (NID_SPAIN_DNI)
 - Sri Lankan National Identity Card (NID_SRI_LANKA_NIC)
 - Swedish Personnummer (NID_SWEDEN_PIN)
 - Swiss Social Security Number (NID_SWITZERLAND_AVS)
 - United Arab Emirates ID (NID_UAE_IC)
 - United Kingdom NI Number (NID_UNITED_KINGDOM_NINO)
 - United Kingdom Self Assessment UTR Number (NID_UNITED_KINGDOM_SA_UTR)
 - United States Social Security Number (NID_UNITED_STATES_SSN)
 - Australian Medicare Card (PHD_AUSTRALIA_MEDICARE)
 - United Kingdom Health and Care Number (PHD_UNITED_KINGDOM_HCN)
 - United Kingdom National Health Service Number (PHD_UNITED_KINGDOM_NHS)
 - United States Health Insurance Claim Number (PHD_UNITED_STATES_HICN)
 - United States Health Plan Identifier (PHD_UNITED_STATES_HPID)
 - United States Medicare Beneficiary Identifier (MBI) (PHD_UNITED_STATES_MBI)
 - United States National Provider Identifier (PHD_UNITED_STATES_NPI)
 - Australian Telephone Number (PII_AUSTRALIA_PHONE)
 - Austrian Telephone Number (PII_AUSTRIA_PHONE)
 - Canadian Telephone Number (PII_CANADA_PHONE)
-

-
- South Korean Phone Number (PII_SOUTH_KOREA_PHONE)
 - Luxembourg Phone Number (PII_LUXEMBOURG_PHONE)
 - Portuguese Phone Number (PII_PORTUGAL_PHONE)
 - Credentials username (PII_MISC_CREDENTIALS_USERNAME_ASCII)
 - Dutch Driver License Number (PII_NETHERLANDS_DLN)
 - Dutch Telephone Number (PII_NETHERLANDS_PHONE)
 - New Zealand Telephone Number (PII_NEW_ZEALAND_PHONE)
 - United Kingdom Telephone Number (PII_UNITED_KINGDOM_PHONE)
 - United States Telephone Number (PII_UNITED_STATES_PHONE)
 - United States Mailing Address (PII_USA_ADDRESS)
 - Turkish Telephone Number (PII_TURKISH_PHONE)
 - Spanish Driver License Number (PII_SPAIN_DLN)
 - Spanish Telephone Number (PII_SPAIN_PHONE)
 - Belgian Telephone Number (PII_BELGIUM_PHONE)
 - French Telephone Number (PII_FRANCE_PHONE)
 - German Driver License Number (PII_GERMAN_DLN)
 - German Passport Number (PII_GERMAN_PASSPORT)
 - German Telephone Number (PII_GERMAN_PHONE)
 - Irish Driver License Number (PII_IRELAND_DLN)
 - Irish Passport Number (PII_IRELAND_PASSPORT)
 - Irish Telephone Number (PII_IRELAND_PHONE)
 - Italian Telephone Number (PII_ITALY_PHONE)
 - Polish Passport Number (PII_POLAND_PASSPORT)
 - Polish Telephone Number (PII_POLAND_PHONE)
 - Portuguese Driver License Number (PII_POLAND_DLN)
 - Swedish Driver License Number (PII_SWEDEN_DLN)
-

- South Korean Passport (PII_SOUTH_KOREA_PASSPORT)
- South Korean Driver License Number (PII_SOUTH_KOREA_DLN)

For more information on these infotypes, refer to the [Information types](#) section in the appendix.

Classification profiles

You manage classification profiles through the **Classification Profiles** page, which is accessed by clicking the **Classification Profiles** link in the **Data Discovery** sidebar on the left.

From the **Classification Profiles** page you can:

- View all the available classification profiles. See [Viewing classification profiles](#).
- Create a new classification profile. See [Adding classification profiles](#).
- View details of a selected classification profile. See [Viewing details of classification profiles](#).
- Modify an existing classification profile. See [Editing classification profiles](#).
- Create a new classification profile from an existing one. See [Duplicating classification profiles](#).
- Remove a classification profile. See [Removing classification profiles](#).

Viewing classification profiles

The **Classification Profiles** page lists available classification profiles. Initially, the page shows prebuilt classification profile templates only. Newly created and duplicate classification profiles are also shown on this page. Duplicating classification profiles is the process of creating copies of existing profiles with identical properties. Additionally, the page shows the total number of available profiles.

The list view of the **Classification Profiles** page shows the following details:

Item	Description
Name	Name of the classification profile.
Sens. Level	Sensitivity level applied to the profile.
Modified	Time when the profile is modified.
Tags	Number of applied tags.

Tip

Use the **Search** text box to filter classification profiles. Search results display classification profiles that contain specified text in their names.

By default, classification profiles are listed in ascending alphabetic order of their names. Classification profiles can be sorted by their names and percentage of sensitivity levels.

Classification profiles can be filtered using the **Sens. Level** filter.

Note

To run ML scans on data stores, you require classification profiles that include both built-in and machine learning (ML) infotypes.

Classification profile templates

Classification profiles can be created based on predefined templates or custom templates. You can use the following predefined templates:

- **GDPR (General Data Protection Regulation):** Affects organizations that process the personal data of EU citizens, regardless of where the organization is headquartered. Apart from the general GDPR template you can narrow down the scope by selecting one of these specific GDPR templates:
- **GDPR Financial:** Covers a subset of GDPR information related to personal financial data.
- **GDPR Personal Details:** Covers any GDPR personal data not captured by the other GDPR templates.
- **GDPR Healthcare:** Healthcare data related GDPR template.
- **GDPR National ID:** Can be used to create a classification profile related to national ID data.
- **HIPAA (Health Insurance Portability and Accountability Act):** Covers the healthcare information in the US. HIPAA relates to protection, encryption, and key management.
- **PCI DSS (Payment Card Industry):** Affects organizations that play a role in processing credit and debit card payments. These organizations must comply with the strict PCI DSS (Data Security Standard) compliance requirements for the processing, storage, and transmission of data.
- **CCPA (California Consumer Privacy Act):** Affects organizations that process the personal data of a California resident, regardless of where the organization is headquartered.
- **Privacy Shield:** Privacy Shield Framework (US). Regulates transatlantic exchanges of personal data for commercial purposes between the European Union and United States.
- **UK GDPR:** Although EU GDPR no longer applies in the UK after the Brexit, UK's DPA 2018 has already enacted the EU GDPR's requirements into UK law (effective from January 1, 2021). The template captures this new regime known as UK GDPR.
- **LGPD (Brazil):** Brazilian General Data Protection Law.
- **PDP:** The Personal Data Protection Bill.

Adding classification profiles

Use the Add Classification Profile screen to add a classification profile. Adding a classification profile involves the following steps:

Select Profile Template

1. In the **Classification Profiles** page, click the **+ Add Profile** button on the right. The **Add Classification Profile** screen is displayed.

2. In the **Select Profile Template** step, select the required profile template from the available options.

See [classification profile templates](#) for more information about these classification profile templates.

3. Click **Next** to go on to the **Name and Describe** screen.

General Info

1. Specify a **Profile Name**. The name must be longer than two characters and up to 64 characters.

2. Provide a **Description** for the profile (up to 250 characters).

3. Select a **Sensitivity Level** from the drop-down list. A sensitivity level suggests DDC what level of sensitivity is OK to find in this data store. For details, see [Sensitivity Levels](#).

4. Click **Next** to go on to the **Select Infotypes** screen.

Select Infotypes

1. The **Select Infotypes** screen is displayed. The screen shows the list of available [information types](#). The screen shows details such as Infotype Name, Category, and Region. !!! note

Based on the selected profile template, certain information types may already be applied/selected. The applied information types can be viewed by turning on the ****Selected only**** toggle switch.

2. Search for the required infotypes. You can use the following options:

- **Search** text box: Enter text to filter information types. Search results display information types that contain specified text in their names.
- **Category** filter: Click the filter icon, select or clear categories, and click **OK**.
- **Region** filter: Click the filter icon, select or clear categories, and click **OK**.

3. Click **Next** to go on to the **Apply Tags** screen.

Apply Tags

1. The **Apply Tags** screen is displayed.

Note

Based on the selected profile template, certain tags may appear already applied. You can select existing tags, enter new tags, and remove existing tags, as appropriate.

2. Select a tag from the **Add Tags (optional)** drop-down list. Please check the list of prebuilt tags in **Predefined Tags**.

Tip

New tags can also be added. Start typing a new tag, and click the **New:** link that appears below the drop-down list.

Add as many tags as needed.

To remove a tag, click the close icon in the tag name.

3. Click **Save**.

The newly created classification profile appears on the **Classification Profiles** page. By default, profiles are displayed in alphabetic order by name. Depending on the number of entries per page, you might need to navigate to other pages to view the newly created profile.

Viewing details of classification profiles

The default view of the Classification Profiles page lists prebuilt profile templates, created profiles, and duplicate profiles. Names of profiles and their modification times are also shown. Additionally, the view shows infotypes, sensitivity levels, and tags applied to available classification profiles.

The edit view of the page shows additional details of each classification profile. The details include the number of linked scans, profile name, profile description, applied sensitivity level, list of linked infotypes, and applied tags.

To view details of a classification profile:

1. In the left pane of the **Data Discovery** application, click **Classification Profiles**. The **Classification Profiles** page is displayed. This page lists available classification profiles.
2. Click the ellipses icon (**⋮**) corresponding to the desired profile. A shortcut menu appears.
3. Click **View**. The **Classification Profiles** page shows additional details of the profile.

Note

For new and duplicate classification profiles, the button name is **View/Edit**. Clicking this button shows the edit view of the **Classification Profiles** page. The details can be viewed and edited on this page. Refer to [editing classification profiles](#).

Only the users with appropriate rights can see the **View/Edit** button. For all other users, only the **View** button is visible.

Editing classification profiles

Newly created and duplicate classification profiles can be modified to suit your requirements. Use the edit view of the page to modify individual classification profiles. You can edit the profile name, profile description, applied sensitivity level, linked infotypes, and applied tags.

Note

Prebuilt classification profiles cannot be edited. However, you can duplicate them and edit the copy to suit your requirements.

To edit a new or a duplicate classification profile:

1. In the left pane of the **Data Discovery** application, click **Classification Profiles**. The **Classification Profiles** page is displayed. This page lists available classification profiles.
2. Click the ellipses icon () corresponding to the desired profile. A shortcut menu appears.
3. Click **View/Edit**. The edit view of the **Classification Profiles** page appears.

Note

For new and duplicate classification profiles, the button name is **View/Edit**. Clicking this button shows the edit view of the **Classification Profiles** page. The details can be viewed and edited on this page.

Only the users with appropriate rights can see the **View/Edit** button. For all other users, only the **View** button is visible.

4. Expand **GENERAL**. General details are displayed.
5. Modify the required details.

6. Expand **INFOTYPES**. The list of infotypes is displayed.
7. Select or clear infotypes, as required.
8. Expand **TAGS**. The applied tags, if any, are displayed.
9. Add new tags or modify existing tags, as required.
10. Click **Save Changes**.

The list view of the Classification Profiles page shows updated information.

Duplicating classification profiles

Duplicating classification profiles is the process of creating copies of existing profiles with identical properties. This process simplifies the creation of new profiles. Duplicate profiles can be modified later to suit your requirements, if needed.

To duplicate a classification profile:

1. In the left pane of the **Data Discovery** application, click **Classification Profiles**. The **Classification Profiles** page is displayed. This page lists available classification profiles.
2. Click the ellipses icon (**⋮**) corresponding to the desired profile. A shortcut menu appears.
3. Click **Duplicate**.

A message appears stating that the profile has been duplicated successfully. The duplicate profile with the name **< original_profile_name> - Copy** appears on the Classification Profiles page. For example, if the profile **APA - Australia Privacy Amendment** is duplicated, a profile named **APA - Australia Privacy Amendment - Copy** is created.

Removing classification profiles

You can remove a classification profile in the **Classification Profiles** screen. To remove a classification profile follow these steps:

1. Click the ellipses icon (**⋮**) corresponding to the desired classification profile.
2. In the shortcut menu that is displayed, select the **Remove** option.
 - If the classification profile has scans associated with it, a **Remove Classification Profile** message box with the following information will be displayed:

"Are you sure you want to remove this classification profile? This cannot be undone. This classification profile is actively associated with the following scans:..."

The list of associated scans is displayed further down in the message box. **Keep in mind that removing such a classification profile can have a significant effect on the associated scans and reports.**

Confirm the classification profile removal by clicking the **Remove** button or click the **Cancel** button to

cancel the operation.

- If the classification profile has no scans associated with it, you can simply click the **Remove** button to confirm its removal.

Scans

DDC provides pattern-matching scan capabilities based on regular expressions. DDC ML, an extension of DDC, enhances DDC's scan capabilities by using semantic context-based discovery, classification of sensitive entities, and enabling similarity search. Refer to [DDC ML Infotypes](#) for infotypes supported by DDC ML.

DDC ML only supports local and network storage data stores.

You can manage scans using the **Scans** page. The **Scans** page can be accessed by clicking the **Scans** link in the **Data Discovery** sidebar on the left.

From the **Scans** page you can:

- View all currently available scans. See [Viewing scans](#).
- Create a new scan. See [Adding scans](#).
- Run a scan manually. See [Running scans](#).
- Delete a scan. See [Removing scans](#).
- Modify an existing scan. See [Editing scans](#).
- View scan history. See [Viewing scan history](#).
- Create a copy of a scan. See [Duplicating scans](#).

Viewing scans

The graphical view shows the following details:

- **Scan Age:** Displays a pie chart representing the distribution of scans conducted over different periods.
- **Scanned Data Objects:** Displays the count of scanned data objects, distinguishing between sensitive and non-sensitive types.

Click the refresh button to refresh the displayed information.

The list view of the **Scans** page shows the following details:

Item	Description
Scan Name	Name of the scan. Click the scan name to view scan configuration details. See Adding scans to understand details of the configuration section.
Status	Status of the scan. For more information, see Scan Statuses .

Item	Description
Duration	Time taken to complete the run.
Last Scan	Time when the scan last ran.
Schedule	Schedule of the scan.
Profiles	Number of classification profiles.

Tip

If you are planning to perform a CipherTrust Manager upgrade, make sure that you do not have scans in progress.

Use the **Search** text box to filter scans. Search results display scans that contain specified text in their names. By default, scans are listed in ascending order by name, but you can also sort them by last scan time, duration, and status.

Click the expand arrow (▸) next to the scan name to view additional scan details, including the number and list of data stores scanned, and the number of target paths. These details are available for the Completed, Partially Completed, and Failed scans.

In the expanded panel, click **View details** to see the scan status for each target path in the data store. You can hover over the status to see the error message related to the scan failure. Click **Close** to return to the Scans page.

Adding scans

To add a scan, navigate to the **Scans** screen (**Data Discovery > Scans**). Click the **+Add Scan** button to open the **Add Scan** screen.

In the screen, you have to go over these configuration steps for each scan that you add:

1. **General Info** - Name the scan, provide short description, and advance configuration.
2. **Select Data Stores** - Select which data stores will be scanned.
3. **Add Targets** - Narrow down the scan scope by selecting specific scan targets.
4. **Select Profiles** - Choose which Classification Profile you want to scan for.
5. **Apply Filters** - Add a list of rules to filter some targets when the scan is launched.
6. **Schedule Scan** - Configure when you want your scan to run.

General Info

1. In the **General Info** screen, specify a unique name and short description for the scan.
 - **Name** - The name must be longer than two characters and up to 64 characters.

- **Description** - (optional) Description can be of up to 250 characters.

2. Expand **Advanced Configuration** and specify the following details:

Parameter	Description
Scan Priority	Set the scan priority relative to other applications in terms of CPU utilization. You can select Low or Normal. The default setting is Low Priority. It applies only to local storage. If you want to increase the scan performance, set the scan priority to Normal .
Content Supported (Not applicable for DDC ML)	Select content type that the scan will process: ▪ OCR - Scans images for sensitive data using Optical Character Recognition (OCR). By default, it is disabled (the scanning of images will be skipped). For more details, see Using OCR in Scans. ▪ EBCDIC: Scan file systems that use IBM's EBCDIC encoding. By default, it is disabled. Note Use EBCDIC mode only if you are scanning IBM mainframes that use EBCDIC encoded file systems. This mode forces scanning of targets as EBCDIC encoded file systems, which means that it does not detect matches in non-EBCDIC encoded file systems.
Trace Logs	Use toggle switch to enable trace logs and capture detailed scan trace messages when scanning a target. By default, it is disabled. For more information, see Viewing Scan Log. Note ▪ You need to run the scan again after enabling trace logs in order to download them. ▪ Trace logs may take up a large amount of disk space, depending on the size and complexity of the scan, and may impact system performance. Enable this feature only for troubleshooting. ▪ DDC supports trace logs for multiple scan statuses. Refer to Scan Statuses for the list of supported statuses and the supported trace log download formats.
Enable Indexing	Use toggle switch to enable or disable similarity search on the scan results. When it's enabled, the similarity search can be performed on the scan results. This option becomes available after onboarding MLaaS. See Configuring MLaaS .

Parameter	Description
Memory Usage Limit (MB)	Set the maximum memory usage (in MB) that the scanner service can use on the data store host. The default memory usage limit is 2048 MB. If you want to increase the scan performance, set the memory usage limit between 4GB to 8GB.
Throughput (MBps)	Set the maximum I/O rate (in MBps) that the scanner service will use to read data from the data store. By default, it is set to 0 (for unlimited).
Amount of Data Object Volume	Select the amount of data object volume prioritizing either number of data objects or metadata per data object. Choose from: ▪ <i>Low - maximum metadata</i>: Captures maximal detail per file. ▪ <i>Medium - core metadata</i>: Balances quantity of files and matching detail in each file. ▪ <i>High - minimal metadata</i>: Results in a more even spread of match data across a large quantity of files. This is the default option.
Number of Rows (Only for Relational Database DS)	(Optional) Set the number of rows to scan in each relational database table. The maximum number of allowed rows for the database scan is 2147483647. The supported databases are: ▪ IBM DB2 ▪ Oracle ▪ Microsoft SQL ▪ PostgreSQL ▪ SAP HANA ▪ MySQL ▪ Teradata Note ▪ If you don't specify the number of rows to scan, the entire databases will be scanned. ▪ The number of rows to scan for all tables are selected in the descending order of the primary key. ▪ SAP HANA - Rows are selected in the ascending order. ▪ Teradata - Rows are selected randomly, if no primary key is defined.

The **Restore Defaults** button resets the advanced settings to their default values, however, if you already previously modified these settings for a scan and ran it with the changed configuration the **Restore Defaults** button will roll back the changes to the last saved configuration. In other words, the **Restore Defaults** button only reverts the current modifications.

3. Click **Next** to move on to the **Select Data Stores** screen.

Select Data Stores

The **Select Data Stores** screen lists all data stores in tabular form. By default, no data stores are selected. The table has three columns:

- **Data Store Name:** Lists available data stores (with their number).
- **Type:** The type of the data store, such as Local Storage, Network Share, etc.
- **Ready to Scan:** Displays if the agent that is connected to the data store. In this column, you can also see if the Agent is ready (that is, if the data store is ready).

Note

- **Ready/Not Ready data store:** A scan cannot run unless there is an identified Agent for every data store included in the scan. Data store that has an agent associated with it has the status **Ready**. If a scan contains a data store with **Not Ready** state, scan run will fail and display an error. If there are more than one data store with **Not Ready** state associated with a scan, the scan run will fail on the first data store that is in **Not Ready** state and will not scan the remaining data stores.
- **Disabled/Enabled data store:** You can manually deactivate a data store from the Data Store page. Deactivated data store has a status **Disabled** and is not scanned. A scan will successfully run (without an error) if it has at least one enabled data store and several disabled data stores associated with it. However, only the enabled data stores will be scanned. If a scan contains all **Disabled** data stores, then scan will not run at all.

To select a data store to scan:

1. Search for the desired data stores by specifying the search criteria in the **Search** box. The search results will be displayed in the table under it.
2. Select a data store for the scan by selecting the corresponding check box. Similarly, select multiple data stores, if needed.

Tip

Use the **Selected only** toggle switch to display only the selected data stores or all data stores (if the switch is 'off' all data sources are displayed).

3. Click **Next** to move on to the **Add Targets** screen.

Add Targets

In the **Add Targets** screen you can review a list of the data stores that you selected for the scan. By default, the scan will scan the entire data store, and this screen allows you to narrow down the scan scope by selecting specific targets for your selected data stores. The **Add Targets** screen is divided into three columns:

- **Data Store Name:** Lists selected data stores.
- **Add Target:** You can type in the complete target path in the field and add it to the scan parameters. Or, you can use the **Browse** button to navigate the target path from the root level or starting from an initial path and add it to the scan parameters. The scan will be performed only on the selected target paths.

Note

- For SAP HANA, the full data store scan is not supported. You need to specify at least one path.
- When adding Oracle and IBM DB2 targets, specify the table name exactly as in the database. Table names are case-sensitive.
- Any scan target that you add must be **valid**, otherwise the scan will fail. For more information on what a valid scan target is, see [Target Format Limitations](#).
- For performance sake, try running smaller scans and then generate a report in which you aggregate them. You may schedule a different scan per Data Store and/or per Classification Profile and/or subpaths (such as folders and tables) in the original scan path.
- You can scan emails in a Gmail label if you move the emails to the label - otherwise, they will be kept in your inbox. For the default system labels, Gmail creates some folders that do not match the label name. Please refer to the [Gmail documentation](#) to learn the right path to scan a particular system label.
- In the case of a Sharepoint Server data store with an API passwords file configured, you have to use an empty target path.
- To perform a scan on an Office365: OneDrive for Business or Exchange Server data store you **have to specify a scan target path**. For details, see [Target Format Limitations](#).

To add a scan target for a selected data store, do one of the following:

- Type complete scan target path in the **Add Target Path** field and click **Apply**.
- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level. Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

Note

Paths are case sensitive. Providing incorrect initial paths may lead to unexpected results. See [issues encountered while browsing target paths](#) for more details.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

- b. In the left pane, navigate and select the desired target path.

Note

To view subfolders within the folder hierarchy of a *SharePoint Online* or *SharePoint Server* data store, select the folder name and click **List**.

- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

To remove a scan target for a selected data store:

1. Click the arrow button next to the data store name for which you want to remove a scan target.
2. Click the **Remove** link on the right of the scan target to remove it.

Once all targets are added, click **Next** to move to the **Select Profiles** screen.

Tip

Make sure that you do not have nested target paths in a scan for the same data store. This can affect the performance of the scan and you can get duplicated data in the reports.

Select Profiles

The **Select Profiles** screen lists all [classification profiles](#) in tabular form. By default, no profiles are selected. The table has two columns:

- **Classification Profile Name:** Lists available profiles. Items marked with a letter "T" are predefined classification profile templates. For more information about these templates, see [Classification Profile Templates](#). The other items are custom classification profiles.

Sensitivity: Displays the sensitivity level assigned to this classification profile. See [Sensitivity Levels](#) for more information.

To select a classification profile for the scan:

1. Use the search box to specify the search criteria for the desired profiles. The search results are displayed in the table under it.
2. Select the check boxes corresponding to desired profiles for selecting the profiles for the scan.

Tip

Use the **Selected only** toggle switch to display only the selected classification profiles or all classification profiles (if the switch is 'off' all classification profiles are displayed).

3. Click **Next**.

Note

The **Next** button remains disabled if any of the selected classification profiles contains only the ML infotype. This button is enabled if all the selected classification profiles contain at least one built-in or custom infotype.

Apply Filters

In the **Apply Filters** screen you can add a list of rules to filter some targets when the scan is launched. By default, there are no filters applied, and in this step you can add specific rules which affect the data stores selected and their targets (if you specified any). You can configure as many filters as you want.

1. Click the **Select Filter** menu to expand it. The menu shows you the filters as follows:

- **Exclude Path/DO by prefix**

This filter excludes search locations with paths that begin with a given string. It can be used to exclude entire directory trees. Example of such a filter: c:\windows\system32

- **Exclude Path/DO by suffix**

This filter excludes search locations with paths that end with a given string. For example, entering led.jnl, excludes files and folders such as canceled.jnl, totaled.jnl.

- **Exclude Path/DO by expression**

This filter excludes search locations by expression. Wildcards "*" and "?" can be used to form expressions for this filter. For example, *data.txt excludes files that end by "data.txt" in any path.

- **Include DO modified recently**

Use this filter to include search locations modified within a given number of days from the current date. For example, enter 14 to display files & folders that have been modified not more than 14 days before the current date.

- **Exclude DO greater than size**

This filter excludes files that are larger than a given file size (in MB).

- **Include DO's within modification date**

Use this filter to include search locations modified within a given range of dates. Files and folders that fall outside of the range set by the selected start and end date are not scanned.

Note

The exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using wildcard characters](#) to learn how wildcards work.

2. For each new filter added click **Apply** to save and apply its rules.
3. Click **Next** to move on to the **Schedule** screen.

Note

Filters are case insensitive. That is to say, if you have two directories, "TEST" and "test" and apply the filter */test both directories will be excluded as a result. The same goes for filenames.

Depending on the type of data store, some considerations should be taken into account. For more information, see individual data store pages at [Discovering sensitive information](#).

Schedule Scan

Scans can be run either manually or automatically at a scheduled time. To configure this:

1. In the **Schedule** screen select the frequency with which you want the scan to run. The options are:
 - **Manual:** Select to run the scan manually. This is the default setting. In this case the scan will be run

whenever you manually launch it from the **Scans** screen. For more information about running a scan manually, see [Running Scans](#).

- **Automatic Scan Pause** - Use this switch to schedule the time when a scan will pause. For example, you should pause all scans (by using the automatic scan pause) during working hours so they do not affect production servers. Next, use the **Time Zone** and **Select days and time** controls to set the days and time when the scan should pause.
- **Run Now** - If you select this check box, the scan will be run just once after the scan is added successfully.
- **Scheduled**: Select to specify a schedule for the run. The scan will be run automatically on the specified schedule. When **Scheduled** is selected, the following fields appear on the screen:
 - **Increment**: Select the increment pattern of the run. This is a mandatory field. The options are **Daily**, **Weekly**, and **Monthly**. By default, **Daily** is selected.
 - **Every**: Specify when the run should repeat. This is a mandatory field.

For example, if Daily is selected as Increment, enter 2 to run the scan once every two days. If Weekly is selected as Increment, enter 2 to run the scan once every two weeks. Similarly, if Monthly is selected as Increment, enter 2 to run the scan once every two months.

- **Time**: Specify the time when the run should start. This is a mandatory field. Specify the time in 12-hour format.
- **Time Zone**: Select a time zone from the drop-down list.
- **Starting**: Specify the day when the schedule should start. This is a mandatory field. By default, **Today** is selected. To specify a particular start date, select **On this date**, click the calendar icon, and select the date.
- **Ending**: Specify the day when the schedule should end. This is a mandatory field. By default, **No End** is selected. To specify a particular end date, select **On this date**, click the calendar icon, and select the date.
- **Automatic Scan Pause** - Use this switch to schedule the time when a scan will pause. For example, you should pause all scans (by using the automatic scan pause) during working hours so they do not affect production servers. Next, use the **Time Zone** and **Select days and time** controls to set the days and time when the scan should pause.

2. Click **Save** to complete adding the scan.

As a result, the newly created scan appears on the **Scans** page. By default, scans are displayed in alphabetic order by name. Depending on the number of entries per page, you might need to navigate to other pages to view the newly created scan. By default, the Status of a newly created scan is Unscanned.

Note

If your CM system clock does not match the Agent's system clock, your scans will not run as scheduled, so it is highly recommended to set up a NTP server to synchronize the clocks. This can be achieved in CM through the **Admin Settings -> System -> NTP**. For details, refer to the [Thales CipherTrust Manager Administrator Guide](#).

Running scans

To run a scan manually:

1. Navigate to the **Scans** screen (**Data Discovery > Scans**).
2. Search for the scan to run.

Tip

Use the **Search** text box to filter scans. Search results display scans that contain specified text in their names.

By default, scans are listed in ascending alphabetic order of their names. Scans can be sorted by their name, last scan time, duration, and status.

3. Move the mouse pointer to the row that contains the scan. The **Run Now** button appears. This button disappears as soon as the mouse pointer is moved out of the row.
4. Click **Run Now**.

As soon as the scan is initiated, its status changes to **Pending**, then the status changes to **Processing**. If the automatic scan pause is configured for the scan and you are running it within the set time window, the status of that scan will be **Autopaused** throughout the duration of the time window. After that, the scan is resumed. For more details on the scan auto pause feature, refer to the information in [Schedule scan](#).

After running a scan, you can generate aggregate and trend reports for them. See [Reports](#) for more details.

Scan statuses

The status of the scan changes in the sequence: Unscanned > Validating > Pending > Running now / Paused / Stopped > Processing > Completed / Failed / Partially Completed.

The progress of the scan (that is, its current status) is displayed in the **Status** column on the **Scans** screen.

The table below provides information about possible statuses and their log download formats.

Status	Log Download Format	Description
Unscanned	-	By default, the Status of a newly created scan is Unscanned.
Validating	JSON	Checking if all the data stores are ready.
Pending	JSON	Scan is pending and the linked data stores are being contacted. Depending on factors such as the network connectivity, this stage may: • Complete in a flash. You may not see it on the Scans page. • Remain for some time in this state.
Running now, Paused, Stopped	JSON	Scan is running, paused, or stopped. See Scan progress for more information.
Autopaused	JSON	Scan is paused as a result of automatic scan pause.
Uploading	-	Scan results are being uploaded to TDP.
Indexing	JSON	(Applicable to DDC ML scans) Scan is creating entity and document indexes for similarity search.
Processing	JSON	Scan is processing the collected data.
Completed	JSON	Scan was successful.
Failed	JSON	For some reason, the scan failed. Hover the mouse over the "Failed" icon to learn more about the reason why it failed.
Syncing	JSON	DDC is communicating with the agents to sync the status of ongoing scans that were active when Ciphertrust Manager was last stopped. Displays when Ciphertrust Manager starts.
Partially completed	JSON	Scan is partially completed. A scan is partially completed when at least one, but not all, of its target paths have been fully scanned. Note DDC ML currently does not support partial scanning of

Status	Log Download Format	Description
		data stores. This functionality is only available with regex-based scans.

Note

DDC will always select an agent for every data store when the scan execution begins. It could be the same agent as the previously assigned one, or a new one, regardless of the health status of the assigned agent.

Scan progress

The progress status of **Running** and **Paused** scans is displayed in the form of a progress bar accompanied by a numeric percentage value.

Additionally, you can click the *magnifying glass* on the right of the progress bar to see detailed information about the scan progress.

Scan in progress displays the following information:

- **Progress bars:**
 - The **Regex Process** shows the percentage of regular expressions processed by the scan (to check the scan path).
 - The **ML Process** shows the percentage of DDC ML scan (to check the scan path).
- The **Regex Process** and **ML Process** tabs show up to 5 unfinished scan paths per Hostname/IP ("...% completed") with detailed information displayed in a table, in these columns:
 - **Data Store** - Identifier of the data store on which scan is running. Generally, the identifier matches with the data store's Hostname/IP, but depending on the data store type, it might show different value.
 - **Paths** - Currently scanned scan paths.
 - **% completed** - Scan completion of the current scan path in percent.
 - **Matches** - Number of matches (sensitive items) found in the current scan path.
- The **Agents** tab shows details of the standard agents running the scan in the following columns:
 - **Data Stores** - Identifier of the data store on which the scan is running.
 - **Agent Name** - Agents that are executing scan on the data store.
 - **Last Agent Connected** - Time when the agent was last connected. *This value is updated every five*

minutes.

- **Status** - Connectivity status of the agent (Connected or Not Connected).

Note

- For disconnected agents, **Last Agent Connected** column doesn't display the time of disconnection. It only displays the most recent time when agent was connected.
- The **Agents** tab doesn't show details of the ML agents running the scan.

Editing scans

To edit a scan:

1. Log on to the DDC console.
2. Open the **Data Discovery** application.
3. In the left pane, click **Scans**. The **Scans** page is displayed. This page lists available scans.
4. Search for the scan to edit.
5. Use the **Search** text box to filter scans. Search results display scans that contain specified text in their names.

By default, scans are listed in ascending alphabetic order of their names.

Scans can be sorted by their name, last scan time, duration, and status.

6. Click the ellipsis icon () corresponding to the desired scan. A shortcut menu appears.
7. Click **View/Edit** from the shortcut menu.

The selected scan is displayed, with its configuration settings distributed over these sections (which are exactly the same as the steps of the **Add Scan** screen):

- **GENERAL**
- **DATA STORES**
- **TARGETS**
- **CLASSIFICATION PROFILES**
- **APPLY FILTERS**
- **SCHEDULE**

Select **Run Now** to initiate the scan run after any configuration change. This check box is available for scans that are not in the running state.

For more details on these sections, refer to the [Adding scans](#) section.

8. Click **Expand All** to expand all sections or a plus button (+) in the section in which you want to edit the scan configuration to expand just that section. For information on the available settings, refer to [Adding scans](#).
9. Make the desired changes and click **Save Changes** to save the changes.

When you edit a scheduled scan that was disabled, it gets automatically enabled.

When you edit a scan, you must run it again to see the corresponding report.

Viewing scan log

You can download and view a log of a selected scan if it has trace logs enabled in the advanced scan configuration settings. For more information, see the "Advanced Configuration" under the [General Info](#) section.

1. In the **Scans** screen, click the ellipsis icon () corresponding to the desired scan. A context menu is displayed, with the **Download Logs** option available.
2. Click **Download Logs** in the menu. A dialog box with information "Download logs? Logs for scan "Xyz" are available for downloading." is displayed.
3. Click the **Download** button in the dialog box to confirm the download.

DDC supports the JSON format for log download. Refer to [Scan statuses](#) for the supported log download formats for different scan statuses.

Note

- Trace logs are only available for download when Trace Logs toggle in advanced configuration is enabled, otherwise only troubleshooting logs are available for download.
- When you try to download logs of scans in the Running/Stopped/Paused/Autopaused/Interrupted state, latest logs until that point in time are downloaded.
- When the Scan Trace Logs download is in progress for a scan, downloading the trace log for that scan in parallel is not allowed. You can trigger the API again only after the previous request is complete.
- The Scan Trace Logs can be huge in size depending on the amount of data being scanned, and can also take considerable amount of time in processing before the download starts.

The information written to the log has the following format:

For JSON Format

Parameter	Data Type	Description
timestamp	number	Time stamp (Unix time format) for each action that happened on a path or location during a scan.
action	string enum: source, opening, opened, parsing, decoding, decoded, completed, scanning, inaccess	Action performed on a path or location during a scan.
agent_name	string	Name of the Agent that performed the scan.
path	string	Full path where the action happened.

Removing scans

1. In the **Scans** screen, use the **Search** text box to filter scans and search for the scan that you want to remove.
2. Click the ellipsis icon () corresponding to the desired scan. An overflow menu is displayed, with the **Remove** option available.

Note

The **Remove** option is not always available in the menu, only if a scan is **Failed**, **Completed**, **Stopped**, or **Disabled**.

3. Click **Remove** in the menu. As a result, a warning message "Remove Scan? Are you sure you want to remove this scan?" is displayed.
4. Click the **Remove** button in the warning message window to confirm the removal of the selected scan.

Viewing scan history

You can view the history details of past scan executions and download their logs.

To view the history details of a scan execution:

1. On the **Scans** page, click the ellipses icon (...) corresponding to the desired scan.
2. Select **View Executions**.

The **Execution History** page displays the scan execution history in following columns:

Column Name	Description
Scan Executions	Displays scan execution time stamp in descending order.
Status	Displays the status of scan execution (Failed, Completed, or Stopped). Use the filter button to filter scan execution by their status.
Duration	Displays the duration of scan execution
Logs	Allows you to download scan logs. Click the download button to download logs.

Duplicating scans

You can make copies of existing scans for creating new variants and reducing manual effort of creating scans from scratch. All the configuration details, classification profiles, data stores, target locations, filters, access & tags, schedules, and all other details of the existing scan are replicated in the cloned copy.

To make a copy of a scan:

1. Navigate to **Scans** screen (**CipherTrust Manager > Data Discovery and Classification > Scans**).
2. Click the ellipses icon (...) next to the desired scan that you want to clone.
3. Select **Clone**.
4. Provide a unique name for the new scan.
5. Click **Clone**.

Using Optical Character Recognition in scans

DDC features Optical Character Recognition (OCR) on a number of image file formats. The formats that can be recognized are **JPG / JPEG, BMP, PNG, GIF, TIFF**, and **PDF** that contains any of these image formats.

Note

OCR scans will usually have a lower accuracy than raw text data scans. They may not always recognize all characters in an image due to multiple factors such as poor image quality, unusual fonts, and complex layouts. This may cause unexpected data object matches.

OCR caching

The DDC scanning engine caches the result of OCR on an image within a scan, which can then be reused if the same image is later found in multiple locations within the same scan, for example, when scanning data sources like email in which identical images frequently occur in different email messages.

OCR limitations

The OCR mechanism employed by DDC has the following limitations:

- It cannot detect handwritten information - only typed or printed characters.
- It does not find information stored in screenshots or images of lower quality. The images you scan with OCR enabled must have a minimum resolution of 150 dpi (300dpi or higher is recommended).

At the same time, the accuracy of scans involving OCR will depend on:

- The quality of the image. Any noise in the image such as scanner marks, lines or soft color tones, dust from scanned images, etc.
- The format of the image. Some image formats will result in better detection rates (lossless vs lossy compression).
- Font face, font size and context stored in the image. Fonts within scanned images must be at least 10pt in size. Fonts below that size will not be reliably detected. Abnormally styled fonts may not be clear or consistent.

Note

OCR is not supported for HP UX 11.31+ (Intel Itanium) and Solaris 9+ (Intel x86) operating systems.

Reports

This section covers the following topics:

- Viewing reports.
- Viewing versions of a report.
- Report Types
- Configure reports.
- Generating reports.
- Report details.
- Removing reports.
- Exporting report's data objects.
- Exporting report's inaccessible data objects.

You manage reports on the **Reports** page. To access the **Reports** page, click **Data Discovery and**

Classification > Reports from the left sidebar.

Viewing reports

The **Reports** page lists all available reports including the newly configured reports. Additionally, the page shows the total number of available reports.

The page shows the following details:

Column	Description
Report Name	Name of the report.
Status	Status of the report. The status could be Not Generated, In Progress, Completed, or Failed.
Type	Type of the report (Aggregated or Trend).
Last Run	Time when the report was run.
Duration	The duration time of the report run.
Re-generation	Whether to automatically trigger report regeneration on subsequent runs of the linked scans. Select to enable automatic report regeneration, clear to disable.

You can use the **Search** box to filter reports. Search results display reports that contain specified text in their names. Reports can be sorted by their name (Report Name), and the last time that the report was generated (Last Run).

To view the details of a specific report, click the report name.

Alternatively, click the ellipsis icon () corresponding to the report name and select **View** from the context menu. The context menu also provides options to **Generate**, **Export D.O.**, and **Remove** the report.

Viewing versions of a report

DDC supports report regeneration. As a result, multiple versions of a report can be generated. The details view of the **Reports** page shows available versions of a report.

To view details of a report version:

1. Click the report name. The details of the report are displayed.
2. At the top of the detailed view, click the expand icon () to expand the versions list. The available versions of the report is displayed.
3. Click a version link to view its details.

The details view shows the details of the selected report version. To switch to a different version, click its link in the detail view under the expand icon ().

Report types

You can generate two types of reports with DDC.

- **Aggregate report:** Aggregate report displays consolidated information for multiple scan executions. The two ways to generate aggregate reports are:
 - **Scan run on specific date:** A report that is based on a scan run on a specific date. Such a report shows the scan findings on the selected date. In case of multiple executions of the scan for the selected day, the report will include the information for the latest execution on that day.
 - **Latest scan execution:** A report that always shows the information found on the latest scan execution. This way, the results will reflect (or update) the changes in the sensitive data discovered if the underlying data in the Data Store or the Classification Profile is modified. Still, the report will actually get generated when the user chooses to generate it.
- **Trend Report:** The trend report presents information about the trend of the chosen scan over the last 15 scan executions. It includes trends for various elements such as scanned data objects, identified sensitive data objects, and infotypes. This report assists in comprehending how the scan information develops over time.

Note

To see an updated trend or aggregated report, you can manually regenerate the report to see the results of the last scan execution. Alternatively, for aggregated reports, you can select the **Enable Report Regeneration** check box from the [General Info](#) section to automatically generate the updated report for latest scan execution.

Configure reports

You can configure a report to include information of one or more scans. After the report is configured, it can be generated any number of times. The generated report contains the results of the executed scans.

To configure a report:

1. In the **Data Discovery and Classification** application, click **Reports > + Add Report**.
2. Complete the following steps:
 - a. [General Info](#)
 - b. [Configure Content](#)

General Info

1. Specify the following details:
 - **Name:** Provide a unique name for the report. The name must be longer than two characters and up to 64 characters. This field is mandatory.
 - **Description(optional):** Provide a description for the report.

- **Select report type:** Select the type of the report to generate: **Aggregated** or **Trend**.

Note

- For a partially completed scan, only an aggregated report can be generated.
- DDC ML currently does not support partial scanning of data stores. This functionality is only available with regex-based scans.

- **Enable Report Re-generation** (Applicable only for Aggregated reports): Select the check box to trigger automatic report regeneration. The report will be regenerated automatically whenever any scan linked with the report is executed subsequently (that is, when Latest Execution time is changed).

2. Click **Next**.

Configure Content

The **Configure Content** screen shows available scans with their number and the number of selected scans.

For **Aggregated** reports, scans executions identified in this step will be merged in a single report. For **Trend** reports, you can select the desired scan for which you want to analyze the trend.

Tip

To include the removed scans in the list, leave the **Show Removed Scans** check box selected.

1. Use the **Search** box to filter available scans. Search results display scans that contain specified text in their names.
2. Select the desired **Scan Name**.

After selecting the scans, turn the **Selected Only** toggle on to list only the selected scans.

3. (Applicable only for **Aggregated** reports) Select the preferred way for generating the report.
 - A report that is based on a scan run on a specific date. For such a report, click "Latest Execution" and select the date of the scan that you wish to use.

If **Enable Report Re-generation** was selected and:

- An earlier scan run date is selected for the report, the report will not be automatically regenerated.
- The current date is selected, the report will be automatically regenerated for subsequent scan runs for the current day. However, after the date changes, any new scan runs won't trigger report regeneration.

- A report that can change if the underlying data store or protection profile is modified and a scan is run again. For such a report, leave "Scan Execution" as "Latest Execution".

Note

Dates in orange represent partially completed scans, while dates in green represent fully completed scans.

4. Leave the default settings for **Generate Now** check box. This will immediately generate the report after it is configured. Alternatively, you can clear the check box to generate report at later stage. See [Generating reports](#).
5. Click **Save**.

Generating reports

After you have configured a report, it can be generated at anytime and any number of times.

To generate a report:

1. On the **Reports** page, search for the report that you want to generate.

Tip

- Use the **Search** box to filter the reports. Search results display reports that contain specified text in their names.
- By default, reports are listed in ascending alphabetic order of their names. Reports can be sorted by their name (Name), and the last time that the scan was run (Last Run).

2. Click the ellipsis icon () corresponding to the desired report.
3. Click **Generate**.

As soon as the report starts to run, its status becomes **Not Generated**. The status of the report changes in the sequence: Not Generated > In Progress > Completed / Failed.

Note

Permissions to access the data stores accessed by the scans included in a scan-based report are checked every time the report is run. If the current user no longer has the correct permission for any of them, an

error is displayed.

Report details

The report details page displays the detailed information about the configured report. The upper part of the report details page displays general information about the report, such as the report name, version, and the number of scans included in the report. Click the expand arrow (▸) next to *scans* to see details of completed and partially completed scans.

You can use the **Print Preview** button in the top right corner of the page to print the report. See [Printing report details](#).

Cards

The findings from the scan executions is reported on the following four cards:

- **TOTAL DATA OBJECTS SCANNED** - The count of all data data objects that were included in the scan.

Note

- When scanning Exchange Online and Exchange Server notes data objects, DDC counts notes and their associated attachments as separate data objects.

TOTAL DATA OBJECTS SCANNED count is equal to the sum of *number of notes* and *number of attachments linked with each note* in the corresponding target path. For example, if you scan a target path containing 4 notes, one of which includes an image attachment, the TOTAL DATA OBJECTS SCANNED count will be 5.

- Due to a known limitation, in reports for MongoDB, Azure table, Salesforce, and G-Mail data stores, you will see "N/A" for the TOTAL DATA OBJECTS SCANNED values.
- For G-Mail, DDC ignores copies of the emails that were received in the same second, from the same sender, and with the same subject.
- In case of Binary Large Objects (BLOBs), TOTAL DATA OBJECTS SCANNED reports the parent and child data objects in a given blob.

- **SENSITIVE DATA OBJECTS FOUND** - The count of all data objects containing sensitive data identified during the scan run.
- **SENSITIVE DATA MATCHES** - The count of all sensitive pieces of information found inside the sensitive data objects.

-
- **SELECTED INFOTYPES FOUND** - The count of infotypes found during the scan out of those configured for the scan when it was created (" out of ").

The lower part of the report details screen displays the report information in the following tabs.

Scans tab

This tab provides the list of scans that contributed the information for the report. Scans with orange tick next to them represent partially completed scans, while scans with green tick next to them represent completed scans.

Additionally, you can see the report data in graphical format containing the following information.

- Infotypes Discovered
- Sensitive Data Objects by Content
- Sensitive Infotype Distribution
- Sensitive Data Objects

Information about scan filters

You can view the information about filters applied in the scans. On the **Scans** tab, click the expander arrow next to the scan name. This will display the information about the number and types of filters applied.

Example:

1 Scan Filter Exclude DO greater than size 14000 MB_

Data Stores tab

This tab provides the list of data stores included in the report, with the information about their risk score, sensitivity level, scan name, last scan time, targets, infotypes, data objects scanned, and sensitive objects found in each data store that was scanned.

Expand the data store name to see the following information in graphical format.

- Sensitive infotypes
- Sensitive content
- Classification profiles
- Business department (*applicable only for ML infotypes*)

An orange tick next to a data store name indicates a partial scan, while a green tick indicates a complete scan.

Data Objects tab

This tab provides the list of data objects scanned with below details. Only the top 1000 data objects are displayed, sorted by their Risk score. The first 25 data objects are displayed in the list but you can view more by clicking **Show more**.

Column Name	Description
Data Object Name	The name of the data object scanned and listed in the report details. Click the expand icon (▶) next to the data object to see additional scan details and meta data information about data object. For Oracle and IBM DB2 the result will be displayed in uppercase.
Risk	The number of risks found by the report in the given scanned data object. A risk is the presence of a sensitive item of data.
Type	The type of the scanned data object listed in the report details, such as "File" or "Folder".
Path	The path to the object that is listed in the report details.
Data Store	The name of the data store where the object listed in the report details was found.
Infotypes	The number of information types found in the data object that is listed in the report.

Filter Data Objects:

Use the fields of **Filter Data Objects** filter to search for a specific data objects.

Field	Description
Search	Use the search box to search data objects by name or path. The data object search has following characteristics: - Case-sensitivity is applicable - Partial search is conducted. For example, searching for "bal" will include "baseball" and "balloon" in the search results. - Search is performed for the object name and path.
Sort By	Use the Sort By options to sort the search results ascending or descending order by data object name or risk score.
Type	Select the data object type to narrow down search results.
Data Store	Select the associated data store to narrow down search results.

Note

When searching for the first time, the result will take some time (usually 1 to 3 minutes) to display. During this time, you can navigate through the different tabs inside the report, however, if you leave the **Reports** page, you must repeat the search. The results of a successfully completed search are cached and will allow repeating the same search with a response time of a few seconds only. Generating the report after the search is completed will invalidate the cache as it will render the information outdated.

Note

- Details of a data object scanned partially due to any issue are displayed in the **Inaccessible Data Objects** tab.
- A scanned ZIP file is displayed under the Data Objects tab. However, this tab doesn't indicate whether all the files included in the ZIP file are scanned successfully. To confirm this, review the Inaccessible Data Objects tab.

Inaccessible Data Objects tab

This tab is visible for Aggregated reports. This tab shows the list of inaccessible, skipped, or partially scanned data objects.

The table in the report detail lists the following findings distributed among columns:

Column Name	Description
Data Object Name	The name of the data object scanned and listed in the report details.
Data Store	The name of the data store where the object listed in the report details was found.
Path	The path to the object that is listed in the report details.
Severity	The severity of why the listed data object is inaccessible, skipped, or partially scanned. The severity could be Intervention (least severe), Notice, Error, Critical (most severe).
Reason	The reason why the listed data object is inaccessible. Some of the possible reasons could be: ▪ File size is more than 50% of the RAM configured in the scan properties (for example, the file size is more than 1024 MB and the RAM configured in scan is 2 GB (default)). ▪ File is corrupt

Column Name	Description
	- File permission issue - Unsupported file type - File contains encrypted data Resolve the issues that caused skipping of files and retry the scan.
Date	The date and time when the report is generated.

Note

Due to some known limitations, you might see same data objects listed in both Sensitive and Inaccessible tabs of the aggregated reports.

Examples:

- A partially scanned file (possibly, due to insufficient buffer memory).
- A table with partially scanned rows (due to some password protected content).
- Inaccessible, skipped, or partially scanned data objects by DDC ML agent are not listed on the **Inaccessible Data Objects** tab.

Additional metadata fields in reports

For each report, metadata fields of data objects can be displayed on the reports page. To see the metadata fields, go to the **Data Objects** tab and click the expander arrow next to data object name.

Note

- The number of fields displayed on the GUI depends on the level of details selected in the [scan configuration](#), data store, and file types.
- For MongoDB and Azure Table, no metadata is displayed in Aggregated reports.
- Metadata information of the data object is not displayed when the data object contains only ML infotypes.

The following table shows the complete list of metadata fields that can be displayed for the data objects.

Key	Description
Catalog	Name of the database or catalog.
Client Modified	Client modification timestamp.

Key	Description
Category	Category to which the data object belongs. See Category for list of supported categories.
Date	Date for the resource.
Date Modified	Date when the resource is modified.
Document Created	Date when the document is created.
Document Creator	Name of the document creator.
Document Modified	Date when the document is last modified.
Document Modifier	User who last modified the document.
Encoding	Whether the match is found in an EBCDIC-encoded resource.
File Created	Date when the file is created.
File Modified	Date when the file is last modified.
File Owner	Owner of the file.
Filename	File name for the resource.
Folder	Folder name for the resource.
Instance	Name of the database instance.
Key Columns	Name of the column(s) used as a key in the database scan. If multiple columns are used as the key, the column names will be comma-separated.
Key Source	Source of the key used in the "Key" and "Column:Key" metadata (for example, whether the key is obtained from a primary key column, or unique key column etc.). Enum: Primary Key, Integer Unique Column, String Unique Column, Blob *, Integer Non Unique Column, String Non Unique Column • Blob is the default "Key Source" metadata value for matches detected in BLOB objects as no key column information is stored for BLOB objects.
Number of matches	Displays the number of instances of an infotype found in the data object.
Median of score	Displays the median similarity score per data object and infotype.
Object Created	Date when the Google Cloud Storage object is created.
Object Modified	Date when the Google Cloud Storage object is last modified.
Permission Execute	List of groups, users, or user classes that have execute permissions to the matched object.
Permission Full	List of groups, users, or user classes that have full permissions to the matched object.

Key	Description
Permission Modify	List of groups, users, or user classes that have modify permissions to the matched object.
Permission Read	List of groups, users, or user classes that have read permissions to the matched object.
Permission Special	List of groups, users, or user classes that have special permissions to the matched object.
Permission Write	List of groups, users, or user classes that have write permissions to the matched object.
Processed Rows	Number of rows that were scanned for the table in a database scan.
Schema	Name of the database schema.
Server Modified	Date and time of the last modification by the server. Applicable only to the Dropbox Business Target types.
Subcategory	Subcategory to which the data object belongs. See Subcategory for list of supported subcategories.
Table	Name of the database table in a database.
Track 1	Whether a Track 1 data type was detected.
Track 2	Whether a Track 2 data type was detected.

Printing report details

1. On the **Reports** page, click the report name.
2. Click **Print Preview > Print**.
3. Click **Exit Print View** to return to report details page.

Note

- It is **recommended** to use **A3** and **landscape** settings as print settings to avoid printing distorted charts.
- For the best experience of exporting reports to PDF, use Chrome or Firefox.

Risk score

The risk score reflects the level of the risk to the business that would result from the exposure of the sensitive data objects (i.e., sensitive information) found by a scan. A low risk score number represents lower risk. The risk score depends on the type and number of sensitive data objects found. For example, if a risk score for a single email address found during scan is 10, the risk score for a document that contains thousands of such email addresses will be much higher.

Note

Only a complete removal/deletion of a sensitive data object would reduce the risk score to zero.

Removing reports

You can remove a report in the **Reports** screen. Since reports have no dependencies (i.e. do not affect other resources) you can remove them without problems.

Note

Only users with the right permissions can remove reports, that is Admin, DDC Admin, DDC Report Admin, and DDC Full Report Admin.

To remove a report:

1. Click the ellipsis icon () corresponding to the desired report.
2. In the shortcut menu that is displayed, select the **Remove** option.

A warning message "Remove Report? Are you sure you want to remove this report? This cannot be undone." is displayed.

3. Confirm the report removal by clicking the **Remove** button in the warning message dialogue box. To cancel the report removal, click the **Cancel** button.

Note

After deleting a report, you can create another report with the same name as the one that you deleted.

Reports are not deleted from TDP, which means that if you have the URL of the removed report, with the report ID, you can still view the report after you removed it.

Exporting report's data objects

You can export all the data objects of a report as Newline Delimited JSON (NDJSON) format. You can then view the exported NDJSON file in any editor supporting this format.

There are two ways of exporting those data objects.

- From the **Reports** page, using the **Export D.O.** option.

- From the **Report Details** page, using the **Export All Data Objects** button.

To export the data objects from the **Reports** page:

1. Click the ellipsis icon (**⋮**) corresponding to the desired report.
2. Click **Export D.O.**.
3. Choose the target location for the exported file.

To export data objects from the report details page:

1. Click the report name.
2. Go to the **Data Objects** tab.
3. Click **Export All Data Objects** to export the data objects.
4. Choose the target location for the exported file.

Tip

Please check the [ELK Reference](#) to see how to use the exported data.

Note

When you export the data objects of a *database* data store of a report, the exported NDJSON file also contains sensitive columns that were extracted from the scan results. The list of sensitive columns gathered in the exported file could be partial. The number of columns in the list depends on the scan configuration and the number of sensitive data matches found.

Exporting report's inaccessible data objects

You can export the inaccessible data objects of a report as Newline Delimited JSON (NDJSON) format. You can then view the exported NDJSON file in any editor supporting this format.

To export the inaccessible data objects associated with a report from its **Report Details** page:

1. Click the report name.
2. Go to the **Inaccessible Data Objects** tab.
3. Click **Export Data Objects** to export inaccessible data objects.

The inaccessible data objects report is downloaded as a .ndjson file.

Agents

Use the **Agents** page to manage the following types of DDC agents:

- **DDC Standard Agent:** Manage using **Standard Agents** tab.
- **DDC ML Agents:** Manage using **Machine Learning Agents** tab.

See [DDC agents](#) for installing DDC agents. To reconfigure DDC agents, see [Operations](#).

The **Agents** page allows you to:

- View a list of all available agents. See [Viewing list of agents](#).
- Edit the agent information. See [Editing agent information](#).
- Manage the agent's labels. See [Managing agent labels](#).
- View the agent's time difference information. See [Agent time difference information](#).

To access the **Agents** page, click **Data Stores > Agents**.

Viewing list of agents

The **Agents** page lists all available agents on the **Standard Agents** and **Machine Learning Agents** tabs. After a server restart, the list might take a moment to populate.

The page shows the following information about the installed agents:

Header	Description
Agent Name	Name of the agent (automatically assigned).
Operating System	Operating system on which the agent is running.
Version	Agent version from among the supported agent platforms (for example, Linux 2.3).
Connection IP	The IP address of the agent.
Status	The status of the agent - "Connected" or "Not connected".
Data Stores	The number of data stores that the agent is associated with. NOTE: Modifying data store settings after assigning an agent might trigger a new agent search. If multiple compatible agents are available, the same agent isn't guaranteed to be re-selected.
Labels	The number of labels assigned to the agent.
Local Storage Only	When this option is enabled, the agent can only be used with the Local Storage data stores. When disabled (default), the agent can be used as proxy, and can be associated with other data stores (in the same network).

Header	Description
	NOTE: Disabling a currently used agent prevents it from being selected for new data stores, but does not affect its behavior in the existing data stores.

Note

The agents are shown in any domain. The **Local Storage Only** option is managed independently of the domain, that is you can have an agent with the **Local Storage Only** option enabled in *Domain A*, and the same agent with the **Local Storage Only** option disabled in *Domain B*.

Use the **Search** box to search for an agent. Search results display agents that contain specified text in their names. By default, agents are listed in ascending alphabetic order of their names.

Editing agent information

You can edit agents information only for Standard Agents.

1. In the **Standard Agents** tab, click the ellipses icon () for the selected agent.
2. Click the **View/Edit** option that is displayed in the context menu.

The **Standard Agent** details screen displays. It shows the same information as the main **Agents** screen, split in the **GENERAL** and **LABELS** sections.

3. Modify the agent information as desired and click **Save Changes**.

Managing agent labels

You can add or edit labels only for Standard Agents.

A label is a way to mark an agent, for example its special features, such as its data store scanning capabilities and scanning performance.

For example, you could label an agent "Oracle" if it's set up to scan Oracle databases, or "Critical" if it's dedicated to highly sensitive data stores.

There are no predefined labels, but you can make custom labels as needed. After a custom label is created, it cannot be modified. However, you can assign new labels to the agent. An agent can have multiple labels. Agent labels can only be created in the **Labels** section inside the agents.

To add or edit the labels for an agent:

1. In the **Standard Agents** tabs, click the ellipses icon () for the selected agent.
2. Click the **View/Edit** option.
3. Modify the agent labels as desired (add or remove) and click **Save Changes**.

Note

Editing an agent's labels changes which data stores can automatically select it. Adding a label makes that agent available to any data store configured with the same label.

Refer to [Mitigating security risks](#) section to learn how to use agent labels to enhance the security of your environment.

Agent time difference information

When there is a time difference between the scanner service (where CM is deployed) and the agent machine (where the agent is installed) a clock icon is shown with the difference in seconds (always in seconds):

Icon	Description
	This is the kind of warning that you will see if the time is behind the CM clock.
	This is the kind of warning that you will see if the time is ahead of the CM clock.

Tip

When the date/time has been changed in the agent machine, it's necessary to restart the agent service.

Warning

Be careful when the scan is launched and there is a difference between the scanner service and agent machine. If the agent's clock is slower than scanner service, then the scan will start at that time. Furthermore, it will also delay resuming a scan when using the auto-pause functionality.

Note

If your Agent's system clock does not match with a Cloud Data Store's clock, you may hit issues while adding the Cloud Data Store in DDC, so it is highly recommended to set up a NTP server to synchronize the clocks. This can be achieved in the following manner:

- In CM through the **Admin Settings -> NTP**.
- For Windows agents, refer to: <https://docs.microsoft.com/en-us/troubleshoot/windows-server/identity/configure-authoritative-time-server>.
- For RedHat / CentOS agents, refer to: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-configuring_ntp_using_ntpd.
- For Ubuntu agents, refer to: <https://ubuntu.com/server/docs/network-ntp>.

Automatic agent selection

DDC automatically selects agents to scan data stores. For data stores without an agent on the same host, a proxy agent is selected. Proxy DDC agent scans the data residing on the data store. The automatic agent selection process is triggered under the following conditions:

- A new data store is added.
- An existing data store is updated.
- Labels on any associated agent are changed.
- At the start of every scan execution (to ensure agents are valid and up to date).

Once automatic agent selection process is triggered, the DDC server first identifies all the connected DDC agents and excludes any that are disconnected.

Note

To control the agents that can scan a particular data store:

- Allow access to the data store endpoint, so only desired agents can reach it, and other agents are blocked at the network or data store configuration side.
- Use labels, as indicated in the [Managing agent labels](#) section.

Agent selection for Local data store

For Local data store, DDC selects an agent in this specific order:

-
- **Hostname Match:** It looks for a DDC Agent whose hostname matches that of the data store.
 - **IP Address Match:** If no hostname matches, it then looks for an agent with a network interface whose IP address matches the data store's IP.
 - **Failure:** If neither of these conditions is met, the agent selection for that Local Data Store fails.

Agent selection for other types of Data Stores

For all other types of data stores, DDC follows a more detailed process:

- **Configuration Check:** DDC considers the minimum and maximum number of agents configured for that data store.
- **Label Matching:** It checks the labels associated with the data store. Any agent selected must have at least the same set of labels as the data store (it can have additional ones too).

Agent selection for existing data stores (Re-evaluating previously selected agents):

If this isn't the first time DDC is automatically selecting agents for this data store, it will first verify if the agents chosen in the previous execution are still suitable. They must:

- Be connected to the DDC server.
- Have the required set of labels.
- Not have the "Local Storage Only" configuration attribute set.
- Satisfy the minimum and maximum agent requirements.

The automatic agent selection succeeds if all conditions are met and at least the minimum number of DDC agents can connect to the data store endpoint. Otherwise, DDC runs complete agent selection as if it was a newly added data store instance.

Agent selection for new data stores (if previous agents are not reachable or it is a new selection):

If this is the first selection for this data store (or if the previously selected agents don't meet all conditions), the DDC server will:

- Iterate through all connected DDC agents.
- (For each agent) Verifies its suitability by checking whether:
 - It does not have the "Local Storage Only" configuration attribute set.
 - It has all the mandatory labels defined for the data store.
 - It can connect to the data store endpoint.

This iteration stops once the "maximum number of agents" that can connect are found, or when there are no more agents left to try.

The automatic agent selection succeeds if at least the minimum number of DDC agents can connect to the data

store endpoint. Otherwise, it fails.

Data store status

If agent selection for a data store is successful, DDC sets its status to Ready and includes it in the scan.

If DDC cannot find the required agents, it updates the data store status to Failed and that data store will not be included in the scan.

Note

- Instructions to install and configure DDC agents can be found in the [Data Discovery and Classification deployment guide](#).
- Port 11117 on the CipherTrust Manager appliance must be accessible from DDC agent hosts.
- Data store endpoint needs to be accessible from DDC agent hosts.
- Each data store type may require a specific DDC agent (Windows or Linux) to proxy requests. Refer to the [Agent Compatibility Matrix](#) to verify the required agent type for their Data Store.
- Automatic agent selection algorithm only considers DDC agents with "Local Storage Only" enabled for Local Storage Data Stores. Refer to [Viewing List of Agents](#) for details.

Similarity search

Similarity search enables retrieving sensitive information related to DDC ML specific NER infotypes such as email addresses and GPS. See [DDC ML infotypes](#) to check the complete list of scannable NER infotypes.

Similarity search is used to find specific entities in a database, for example finding sensitive information such as names, addresses, and social security numbers using user queries. It also supports searches in multiple languages by using advanced embedding models that are trained on data from many different languages. These models can understand the meaning of terms and concepts across languages and represent them in a common space. This means that when a user searches in one language, the system can still find relevant content written in another language, as long as the information has clear linguistic meaning.

Similarity search uses advanced text vectorization models and indexing algorithms for fast and accurate information retrieval. To perform similarity search, DDC scans the data store to discover and vectorize sensitive entities. It then uses these embeddings to build a vector index to facilitate querying based on nearest neighbor algorithms. Due to the semantic nature of the search model, multilingual search capabilities are also supported.

Each infotype found in the indexed scan is compared to the user's query using a similarity metric like cosine similarity. Filters for the selected infotype, minimum similarity score, and data stores are also applied. Data objects that don't have the required infotypes (as defined in the search setup) are excluded.

The results are grouped first by data object, then by matched infotype within each object. For each group, the system reports the [number of matches](#) and the [median similarity score](#), which are used to rank the results.

Prerequisites

- Ensure that DDC ML agent is installed and configured. See [DDC ML agent](#) for more details.
- Ensure you are using CipherTrust Manager version 2.20 or later.
- Configure TDPaaS.

Note

The Google GCP region associated with TDPaaS is used for onboarding MLaaS as well.

- Onboard MLaaS cloud component.
- Add Local or Network data stores where you want to perform the search.

See adding [Local storages](#) and [Network storages](#) for adding local and network data stores, respectively.
- Configure and execute a scan on the added data store with indexing enabled for a ML classification profile.
 - See [Classification profile](#) to learn how to create an ML classification profile using ML infotypes.
 - See [Scans](#) to learn how to execute a scan with indexing enabled.

Perform similarity search

1. Log on to the CipherTrust Manager console.
2. Click **Search > Add Search**.
3. On the **General Info** screen, provide the name for the search and click **Next**.
4. On the **Configure Search** screen, perform the following tasks:
 - a. In **Subject Search**, select either **Full Name** or **Primary Email** as mandatory infotype.
 - b. Enter the corresponding name or email address.
 - c. Select additional infotypes from the **Add a new optional field** drop-down.
 - a. From the **This field is** drop-down, select **Mandatory** or **Optional** for infotype type.
 - b. Enter the value for the infotype to search for.
 - c. Click **Add**.

Repeat the above steps to add more infotypes.

d. Click **Next**.

5. On the **Configure Accuracy** screen, provide the search accuracy threshold for different infotypes.

See [Refine search result](#) for more details.

Click **Restore Defaults** to restore the accuracy thresholds to the default value of 90.

6. Click **Next**.

7. On the **Select Data Stores** screen, select the target data stores where you want to search for the infotype.

Note

- Currently, only Local and Network data stores are supported.
- Datastore availability will depend on which scans were indexed previously.

8. Click **Save**.

DDC will perform similarity search for the selected infotypes and display results on the **Search** page.

Note

Similarity search depends on the named entities extracted through NER scans, so search effectiveness is limited by the accuracy of the NER process. Only the entities detected in the scan (as visible in the scan report) are indexed as potential matches for future search queries.

View search results

To view search results, click the corresponding search name. Alternatively, click the ellipsis icon (...) corresponding to the search name and select **Show Results**.

Edit search

You can view, edit, or rerun the search with modified parameters.

1. On the **Search** screen, click the ellipsis icon (...) corresponding to the search name.
2. Select **View/Edit**.
3. Expand the **GENERAL** section to edit the search name.
4. Expand the **CONFIGURE SEARCH** section to add or remove infotypes for searching.

5. Expand the **DATA STORES** section to add or remove data stores.
6. Select the **Run Now** check box to rerun the search after making changes.
7. Click **Save Changes**.

Delete search

To delete a previously created similarity search, click the ellipsis icon (...) corresponding to the search name and select **Remove**.

Refine search result

You can thoughtfully refine the search results by adjusting the threshold value and selecting the number of infotypes:

1. **Similarity Threshold:** This is a score from 0 to 100 that shows how similar the search results need to be to what you're looking for. A higher score means you'll get results that are very similar to your search query. Note that this score measures how related the meanings are, not just how many words match. The default settings are usually helpful, but you can adjust them to fit your needs. For example, if you want only very close matches, you can set the threshold between 95 and 100. On the other hand, cross-lingual queries should have their thresholds relaxed, such as ~40-50%, to get appropriate matches. The below table summarizes various settings for the similarity threshold.

Aspect	Default Threshold	Recommended Adjustments
General Use	90%	Standard starting point; adjust for finer control.
Precision Matches	95% - 100%	Higher thresholds for queries like email addresses, URLs, access keys to minimize false positives.
Ambiguous Infotypes	70% - 90%	Lower thresholds for personal names, mailing addresses, etc., due to variations in representation.
Cross-lingual queries	40% and above	Adjust based on vocabulary (lexicon) and word structure variations across languages.

2. **Mandatory vs. Optional Infotypes:** This means that some types of data must be included in your results (mandatory) while others can be included if they are available (optional). According to DDC requirements, either a full name or an email must always be included. If you require too many mandatory types, it could limit your results because fewer items will meet all those criteria.

Apart from the median similarity score, getting multiple matches for a subject search across a variety of infotypes is a positive indicator of relevance and accuracy.

Discovering sensitive information

DDC helps organizations identify and classify sensitive data within the data stores to ensure compliance, security, and data protection. A data store refers to a repository or location where your sensitive data is stored. This could include cloud storage, databases, local and network storages, servers, or any other storage system.

This section provides information on how to add the supported data stores on DDC and configure scans to discover and classify the data.

You can manage data stores through the **Data Stores** page, which is accessed by clicking the **Data Stores** link in the **Data Discovery** sidebar on the left. The list view of the Data Stores page shows the following details:

Column	Description
Data Store Name	Name of the data store.
Type	Type of the data store.
Ready to Scan	Whether the data store is configured successfully for scanning.
Last Agent Check	Time when the Agent was last scanned.
Location	Location of the data store.
Sens. Level	Sensitivity level applied to the data store.
Labels	The number of agent labels that the datastore has.
Status	Status of the data store - enabled or disabled. During a scan, DDC searches for agents in enabled data stores. Click the toggle switch to change the status.

The status of a data store could be disabled while it waits for an Agent or if it fails to select an Agent. Disabled data stores are skipped during the scan.

Note

- Any modifications in the data store will be effective from the subsequent scans.
- If a data store is associated with a scan, it can't be deleted. Disassociate the data store from the scan and try deleting it again.

Supported data stores

- Local Storages
- Network Storages
- Databases
- Big Data

- Servers
- Clouds

Local storage

You can scan Windows and Linux local storage data stores with Data Discovery and Classification application.

Note

- You cannot scan mounted network shares as local storage.
- As DDC uses the Hostname/IP to determine the system for scanning, avoid having multiple hosts with identical Hostname/IP combinations, as DDC may randomly choose one.
- Install DDC ML agent on local host machine to scan local storage data stores for regex and ML infotypes. If you only want to scan regex based infotype, install the DDC agent on the host machine.
- The DDC ML agent runs as a local system. Ensure that the SYSTEM account has appropriate permissions (typically Full Control) for all files and folders within the data store being scanned. Refer to [Microsoft documentation](#) for more information on default SYSTEM account permissions.

This section covers the following topics:

- [Add local storage data store](#)
- [Add local storage scan](#)

Adding local storage data store

To add the local storage data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Local**.
2. From **Select Data Store Type**, select **Local Storage**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. In the **Hostname/IP** field, specify a valid hostname, IP address, or Uniform Resource Identifier (URI) of the agent running on the machine where the local storage data store resides. Hostname/IP must be longer than two characters. For example, `10.0.0.1` or `agent123`.

Tip

Make sure the agent can resolve its hostname. If it cannot, include the agent name in the operating system's Hosts file. For example, if the agent's hostname is 'agent123', add the entry '127.0.0.1 localhost agent123' in the Hosts file.

2. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the `All groups (default)` access for reports. Alternatively, select a group.

2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Adding local storage Scan

To add a scan for local storage:

1. In the **Data Discovery & Classification** application, click **Scans > Add Scan**.
2. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)

- d. [Select Profiles](#)
- e. [Add Filters](#)
- f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the local data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

Note

If no specific target is added, the entire data store will be scanned.

1. To add a scan target, do one of the following:
 - Under the **Add Target** field, specify the correct target path and click **Apply**.

Note

Ensure that the target path syntax format matches the local storage directory structure. For example, for Windows, use `drive\folder\filename` as the path and for Linux, use `/directory/subdirectory/filename` as the path.

- Navigate and add target paths.

- a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- b. In the left pane, navigate and select the desired target path.
- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store.
You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down.

To filter the locations to scan a local storage data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using wildcard characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Windows Syntax	Windows Example	Linux Syntax	Linux Example
Folder	<DriveLetter> :\<FolderName>	C:\TestData	<TopLevelDir> /<FolderName>	/mnt/testdata
Sub Folder	<DriveLetter> :\<FolderName> >\<SubFolderName>	C:\TestData\Tests	<TopLevelDir> /<FolderName> /<SubFolderName>	/mnt/testdata/tests
File	<DriveLetter> :\<FolderName> >\<SubFolderName>\<FileName>	C:\TestData\Tests\ Data\file.txt	<TopLevelDir> /<FolderName> /<SubFolderName>/<FileName>	/mnt/testdata/tests/ data/file.txt

- Exclude Path/DO by suffix

Excludes paths or data objects that end with a given string. Specify <string>.

Filter Item	Windows Syntax	Windows Example	Linux Syntax	Linux Example
Folder	<FolderName>\ *	TestData* — Applies all paths starting with 'TestData' as suffix.	<FolderName>*	testdata* — Applies all paths starting with 'testdata' as suffix.
File	<FileName>	file.txt — Applies all paths ending with 'file.txt' as suffix.	<FileName>	file.txt — Applies all paths ending with 'file.txt' as suffix.

- Exclude Path/DO by expression

This filter is majorly used with wildcard characters.

Excludes paths or data objects that match the given expression. Specify <string>.

For example, to exclude locations that contain 'blob' in their path, use expression *blob*.

Filter Item	Windows Syntax	Windows Example	Linux Syntax	Linux Example
Folder	*<FolderName>	*TestData*	*<FolderName>	*testdata*

Filter Item	Windows Syntax	Windows Example	Linux Syntax	Linux Example
	*		*	
File	*<FileName> File within specific folder: *<FolderName> \<FileName>	*file1	*<FileName> File within specific folder: *<FolderName> /<FileName>	*file1

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify the **Start** and **End** dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Big data

- Hadoop
- Teradata

Hadoop

This section covers the following topics:

- Prerequisites
- Add Hadoop data store
- Add Hadoop scan
- Generate Kerberos authentication ticket

Prerequisites

- **Supported version:** Hadoop 2.7.3 and above
- **Considerations and Requirements:**
 - **DataNode:** Nodes where data blocks distributed by Hadoop Distributed File System (HDFS) are stored are called DataNodes. DataNodes are treated as “slaves” in a Hadoop cluster.
 - **NameNode:** A node that maintains the index of directories and files and manages data blocks stored on DataNodes is called a NameNode. A NameNode is treated as “master” in a Hadoop cluster.
 - Hadoop is not supported for Debian-based Linux agents.

To scan a Hadoop cluster with HDFS, you must have:

- a. A Target NameNode running Apache Hadoop 2.7.3, Cloudera Distribution for Hadoop (CDH), or similar.
- b. A Proxy host running the Linux 3 Agent with database runtime components for Linux systems.
- c. A valid Kerberos ticket if Kerberos authentication is enabled. See [Generate Kerberos authentication ticket](#).

Add Hadoop data store

To add the Hadoop data store:

1. Log on to the **CipherTrust Manager** GUI.
 2. Open the **Data Discovery and Classification** application.
 3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
-

4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Big Data**.
2. From **Select Server Type**, select **Hadoop Cluster**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name:** Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name:** Location of the data store.
 - **Add Location:** Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store:** Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the Hadoop domain:

Field	Description
Hostname/IP	Specify Hostname/IP of the Hadoop cluster's active NameNode. Specify a valid hostname, IP address, or Uniform Resource Identifier (URI). The hostname must be longer than two characters. For example, if your HDFS share path is <code>hdfs://hadoop-</code>

Field	Description
	server - name / share - name, the host name of the Name Node is share - name.
Port	The port on which the NameNode is accessed. Default is 8020. This is a mandatory field.

- (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
- Click **Next**.

Add Access Control & Tags

- (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
- Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.

- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Hadoop scan

To add a scan for the Hadoop:

1. Open the **Data Discovery and Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:

- Add target path manually.

Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- b. In the left pane, navigate and select the desired target path.
- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an Hadoop data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using wildcard characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify `<string>`.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example, to exclude locations that contain 'blob' in their path, use expression `*blob*`.

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and **End** dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:

- **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
- **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Generate Kerberos authentication ticket

To generate a Kerberos authentication ticket for your HDFS cluster, run these commands in a terminal on the designated Proxy Agent host.

1. Check if a valid Kerberos ticket has been issued for the principal user.

```
klint
```

2. Generate a Kerberos ticket as a principal user.

```
# kinit <username>@<domain>  
kinit DDCuser@example.com
```

Teradata

This section covers the following topics:

- [Prerequisites](#)
- [Add Teradata data store](#)
- [Add Teradata scan](#)

Prerequisites

- **Supported version:** Teradata 17.0.0 and above
- **Considerations and Requirements:**
 - Teradata data stores require **Teradata Tools and Utilities 17.00, 17.10, or 17.20** installed on the Agent along with the following utilities:
 - ODBC Driver for Teradata
 - FastExport

You may have to restart the agent after the installation.

- Scanning of large binary objects is now supported for Teradata. For a list of binary file types supported for Teradata scans, refer to the table in [Binary Large Objects](#).

Note

A scan of a Teradata data store may create temporary tables named `erecon_fexp_<YYYYMMDDHHMMSS><PID><RANDOM>`. **Do not remove these tables** while the scan is in progress. They are automatically removed when a scan completes. If a scan fails or is interrupted by an error, the temporary tables may remain in the database. In this case, it is safe to delete the temporary tables.

Add Teradata data store

To add the Teradata data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery and Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Big Data**.
2. From **Select Server Type**, select **Teradata**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name:** Name for the data store.
 - **Description** (Optional): Description for the data store.

- **Location Name:** Location of the data store.
- **Add Location:** Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
- **Sensitivity Level (Optional):** Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
- **Enable Data Store:** Whether to enable the newly added data store. Select the check box to enable the data store.

2. Click **Next**.

Configure Connection

1. Specify the credentials of the Teradata domain:

Field	Description
Hostname/IP	Specify a valid Hostname of the Teradata server. The hostname must be longer than two characters. This is a mandatory field.
Port	Default is 1111.
User	The name of the Teradata user. Warning Due to known Teradata limitations DDC cannot use the following internal Teradata users to scan: DBC, tdwm, LockLogShredder, External_AP, TDPUSER, SysAdmin, SystemFe, TDMaps, Crashdumps, Sys_Calendar, viewpoint, console.
Password	The password of the Teradata user.

2. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.

3. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Teradata scan

To add a scan for the Teradata:

1. Open the **Data Discovery and Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.
3. Complete the following steps:

-
- a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Add target path manually.

Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Target Path to scan	Syntax	Example
All locations	empty path or [:port]	Leave path blank or :1111
Specific user	<user_name[:<port>]>	userX:1111

Target Path to scan	Syntax	Example
Specific table belonging to a user	<code><user_name[:<port>/table]></code>	<code>userX:1111/accounts</code>
Specific database	<code><database[:<port>]></code>	<code>finance</code>
Specific table	<code><database[:<port>]/table></code>	<code>hr/employees</code>

Note

Paths are case-sensitive.

- Navigate and add target paths.

- a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- b. In the left pane, navigate and select the desired target path.
- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an Teradata data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using wildcard characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify `<string>`.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example, to exclude locations that contain 'blob' in their path, use expression `*blob*`.

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and **End** dates.

2. Click **Apply**.

3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Target limitations

Due to known Teradata limitations Data Discovery and Classification cannot scan the following Teradata internal databases:

SYSJDBC	All	TD_SYSXML	DBC
TDStats	TD_SYSGPL	PUBLIC	SQLJ
SYSBAR	Default	SYSLIB	TD_SYSFNLIB
LockLogShredder	tdwm	TDPUSER	External_AP
EXTUSER	dbcmngr	SystemFe	SysAdmin
TDMaps	TDQCD	Crashdumps	Sys_Calendar
viewpoint	TD_SERVER_DB	console	SYSUDTLIB
SYSUIF	SYSSPATIAL		

Network storage

- NFS Share
- SMB/CIFS Share

NFS Share

NFS (Unix Network File System) is a distributed file system protocol allowing a user on a client computer to access files over a computer network as if it were a local storage.

Note

It is not guaranteed that NFS type data stores will work properly on MAC.

This section covers the following topics:

- Prerequisites
- Add NFS Share data store
- Add NFS Share scan

Prerequisites

- Linux proxy agent.
- Mount target path on the Proxy agent host. See [Mounting an NFS Share](#).
- The target credentials must have minimum required permissions to scan target locations. DDC requires read access for scanning data from the targets.

Note

Install the DDC ML agent on the host server to scan network storage data stores for regex and ML infotypes. To scan only regex-based infotypes, install the standard DDC agent instead.

Add NFS Share data store

To add the NFS Share data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery and Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)

d. Add Access Control & Tags

Select Type & Category

1. Under **Select Data Store Category**, select **Network**.
2. From **Select Server Type**, select **NFS Share**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the NFS Share domain:

Field	Description
Hostname/IP	Provide a valid hostname or IP address of the NFS server. Example : If the NFS shared path is <code>server_ip:/srv/nfs/share</code> , then enter <code>server_ip</code> in the field.
Share Path	Provide a valid NFS path beginning with a slash ('/'). This is the path of the shared directory on the NFS server. Example : If the NFS shared path is <code>server_ip:/srv/nfs/share</code> , then enter <code>/srv/nfs/share</code> in this field.
Agent Hostname/IP	Provide a valid hostname or IP address of the host

Field	Description
	where DDC agent resides. It is mandatory to mount the file share on the host for scanning it.
Mount Point (On Proxy Agent)	Provide a valid directory path on the Proxy host where the NFS Shared path is mounted. This is the local directory on the proxy agent where the NFS shared directory will be accessible. The name of the local directory may be different from the Share path.

Warning

Mount the NFS export on the mount point on the proxy agent host before running any NFS scan.

Note

Use consistent values in the **Hostname/IP** field and the mount command on the DDC Proxy Agent. It is not possible to use the server *IP address* to execute the mount command, and then subsequently use the *hostname* to create the data store, or vice-versa.

2. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.

- **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add NFS Share scan

To add a scan for the NFS Share:

1. Open the **Data Discovery and Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.

3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:

- Add target path manually.

Under the **Add Target** field, specify the correct target path and click **Apply**. For example, `<folder_name>/filename.txt`.

- Navigate and add target paths.

- a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- b. In the left pane, navigate and select the desired target path.
- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification

profiles.

2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an NFS Share data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using wildcard characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify `<string>`.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example, to exclude locations that contain 'blob' in their path, use expression `*blob*`.

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and

End dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual**: This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled**: Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

SMB/CIFS Share

Common Internet File System (CIFS) is a cross-platform, transport-independent protocol that provides a mechanism for client systems to use file and print services made available by server systems over a network.

Note

SMB/CIFS is supported for Windows only. Currently, the SMB implementation on Linux (Samba) is not supported.

This section covers the following topics:

- [Prerequisites](#)
- [Add SMB/CIFS Share data store](#)
- [Add SMB/CIFS Share scan](#)

Prerequisites

- Use a Windows proxy agent.
- Ensure that the target storage is accessible from the Proxy agent host.

Note

Install DDC ML agent on host server to scan network storage data stores for regex and ML infotypes. To scan only regex based infotype, install the DDC agent on the host server.

Add SMB/CIFS Share data store

To add the SMB/CIFS Share data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery and Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Network**.
2. From **Select Server Type**, select **SMB/CIFS Share**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.

- **Enable Data Store:** Whether to enable the newly added data store. Select the check box to enable the data store.

2. Click **Next**.

Configure Connection

1. Specify the credentials of the SMB/CIFS Share domain:

Field	Description
Hostname/IP	Specify a valid hostname, IP address, or URI of the data store. For example, if Windows share path is <code>\\remote-server-name\share-name</code> , enter <code>remote-server-name</code> in the field.
Share Name	Specify a valid Windows share name. Following characters are not allowed in the Share Name: <code>=*?, <> ; : + [] " / \</code>. caution Do not confuse Share Name with Network Path. In Windows, Share Name is typically set in the Advanced Sharing settings in the folder sharing properties.
User	Specify user name. Use appropriate format for the target Windows hosts credentials. ▪ <code><domain\username></code> - Use this format when Windows target host resides in the same Active Directory domain as Windows proxy agent. ▪ <code><target_hostname\username></code> - Use this format when Windows target host does not reside in the same Active Directory domain as Windows proxy agent. ▪ <code><username></code> - Use this format if none of the above formats work.
Password	Specify password.

Tip

DNS / DNS reverse resolution may increase the time to scan. Make sure that you optimize your DNS resolution or modify the agent's hosts file to skip the external DNS resolution as indicated in this [technical note](#).

2. In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.

- **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add SMB/CIFS Share scan

To add a scan for the SMB/CIFS Share:

1. Open the **Data Discovery and Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.

3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:

- Add target path manually.

Under the **Add Target** field, specify the correct target path and click **Apply**. For example, `<folder_name>/filename.txt`.

- Navigate and add target paths.

- a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- b. In the left pane, navigate and select the desired target path.
- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification

profiles.

2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an SMB/CIFS Share data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using wildcard characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify `<string>`.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example, to exclude locations that contain 'blob' in their path, use expression `*blob*`.

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and

End dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual**: This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled**: Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Server

- Exchange Server
- Sharepoint Server

Exchange Server

This section covers the following topics:

- [Prerequisites](#)
- [Configure impersonation](#)
- [Add Exchange Server data store](#)
- [Add Exchange Server scan](#)
- [Scan additional mailbox types](#)
- [Unsupported mailboxes and folders](#)

Prerequisites

Component	Description
Proxy agent	Windows agent

Component	Description
	Note ▪ Agent host architecture (32-bit or 64-bit) must match the Exchange Server. ▪ The Agent host must be able to contact the domain controller (DC). ▪ A valid LDAP over SSL (LDAPS) certificate that is trusted by the DC must be installed on the Agent host. The proxy agent is only required for LDAPS authentication.
TCP allowed connections	▪ Port 443 ▪ Port 389 for LDAP authentication ▪ Port 636 for LDAPS authentication
Version	Exchange Server 2013 and higher.

Configure impersonation

To scan a Microsoft Exchange mailbox, you have the following options:

- Use an existing service account and assign it the ApplicationImpersonation management role.
- (Recommended) Create a new service account and assign it the ApplicationImpersonation management role.

While you *can* assign the ApplicationImpersonation management role to a global administrator and use it for scanning mailboxes, we recommend using a dedicated service account instead. Service accounts are user accounts set up to perform administrative tasks only. Because of the broad permissions granted to service accounts, we recommend that you closely monitor and limit access to these accounts.

Assigning the ApplicationImpersonation role to a service account allows that account to act as the owner of any account that it is allowed to impersonate. DDC uses the permissions granted to this service account to scan those mailboxes.

To assign a service account the ApplicationImpersonation role for all mailboxes:

1. On the Exchange Server, open the Exchange Management Shell and run as administrator:

```
# <impersonationAssignmentName>: Name of your choice to describe the role assigned to the service account.
# <serviceAccount>: Name of the Exchange administrator account used to scan EWS.
```

```
New-ManagementRoleAssignment -Name:<impersonationAssignmentName> -Role:ApplicationImpersonation -User:<serviceAccount>
```

(Advanced) To assign the service account the ApplicationImpersonation role for a limited number of mailboxes, apply a management scope when making the assignment.

To assign a service account the ApplicationImpersonation role with an applied management scope:

1. On the Exchange Server, open the Exchange Management Shell as administrator.
2. Create a management scope to define the group of mailboxes the service account can impersonate:

```
New-ManagementScope -Name <scopeName> -RecipientRestrictionFilter <filter>
```

For more information on how to define management scopes, see Microsoft: [New-ManagementScope](#).

3. Apply the ApplicationImpersonation role with the defined management scope:

```
New-ManagementRoleAssignment -Name:<impersonationAssignmentName> -Role:ApplicationImpersonation -User:<serviceAccount> -CustomRecipientWriteScope:<scopeName>
```

Add Exchange Server data store

To add the Exchange Server data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery and Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Server**.

2. From **Select Server Type**, select **Exchange Server**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the Exchange Server domain:

Field	Description
Domain	Domain name to scan mailboxes that reside on that domain. This is usually the domain component of the email address, or the Windows domain. Note Using the domain IP instead of the domain name will not work.
User	Service account user name.

Field	Description
	Tip The account used to scan Microsoft Exchange mailboxes must: ▪ Have a mailbox on the target Microsoft Exchange server. ▪ Be a service account assigned the ApplicationImpersonation management role. See Configure impersonation.
Password	Service account password.

2. In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Exchange Server scan

Note

If a mailbox is a member of multiple Groups, it is scanned each time a Group it belongs to is scanned. Mailboxes that are members of multiple Groups still consume only one mailbox license, no matter how many times it is scanned as part of a separate Group.

Example: User mailbox "A" belongs to Groups "A1", and "A2". When Groups "A1" and "A2" are added to the same scan, user mailbox "A" is scanned once when Group "A1" is scanned, and a second time when Group "A2" is scanned. Mailbox "A" consumes only one mailbox license despite having been scanned twice.

To add a scan for the Exchange Server:

1. Open the **Data Discovery and Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.

3. Complete the following steps:

- a. [General Info](#)
- b. [Select Data Stores](#)
- c. [Add Targets](#)
- d. [Select Profiles](#)
- e. [Add Filters](#)
- f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:

- Add target path manually.

Under the **Add Target** field, specify the correct target path and click **Apply**. See [Target limitations](#).

- Navigate and add target paths.

- a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- b. In the left pane, navigate and select the desired target path.
-

- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an Exchange Server data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using wildcard characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify `<string>`.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example, to exclude locations that contain 'blob' in their path, use expression `*blob*`.

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and **End** dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Scan additional mailbox types

The following additional mailbox types are supported:

- **Shared mailboxes:** Shared mailboxes do not have a specific owner. Instead, user accounts that need to access the shared mailbox are assigned "SendAs" or "FullAccess" permissions.
- **Linked mailboxes:** A linked mailbox is a mailbox that resides on one Active Directory (AD) forest, while its associated AD user account (the linked master account) resides on another AD forest.

- **Mailboxes associated with disabled AD user accounts:** Disabled AD user accounts may still be associated with active mailboxes that can still receive and send email. Mailboxes associated with disabled AD user accounts are not the same as disconnected mailboxes.

To scan the above supported mailbox types, use a service account with "FullAccess" rights to the target mailbox.

Grant access rights to mailboxes

Note

Adding "FullAccess" privileges to an existing user account may cause issues with existing user configuration. To avoid this, create a new service account and use only that account for scanning Exchange shared mailboxes with DDC.

Changes may not be immediate. Wait 15 minutes before starting a scan on the exchange server.

After the service account is granted access to the target mailboxes, you can add them as targets.

Note

Linked mailboxes as service accounts

You cannot use a linked master account (the owner of a linked mailbox) to scan Exchange Targets in DDC. To successfully scan an Exchange Target, use a service account that resides on the same AD forest as the Exchange Target.

Shared Mailboxes

To grant a service account "FullAccess" rights to shared mailboxes, run the following commands in the Exchange Management Shell:

Grant full access to a specific shared mailbox:

```
Add-MailboxPermission -Identity <SHARED_MAILBOX> -User <SERVICE_ACCOUNT> -AccessRights FullAccess -Automapping $false
```

Here, <SHARED_MAILBOX> is the name of the shared mailbox, and <SERVICE_ACCOUNT> is the name of the account used to scan the mailbox.

Grant full access to all existing shared mailboxes on the Exchange server:

```
Get-Recipient -Resultsize unlimited | where {$_.RecipientTypeDetails -eq "SharedMailbox"} | Add-MailboxPermission -User <SERVICE_ACCOUNT> -AccessRights FullAccess -Automapping $false
```

Here <SERVICE_ACCOUNT> is the name of the account used to scan the mailboxes.

Linked Mailboxes

To grant a service account "FullAccess" rights to linked mailboxes, run the following commands in the Exchange Management Shell:

Grant full access to a specific shared mailbox:

```
Add-MailboxPermission -Identity <LINKED_MAILBOX> -User <SERVICE_ACCOUNT> -AccessRights FullAccess -Automapping $false
```

Here <LINKED_MAILBOX> is the name of the shared mailbox, and <SERVICE_ACCOUNT> is the name of the account used to scan the mailbox.

Grant full access to all existing shared mailboxes on the Exchange server:

```
Get-Recipient -Resultsize unlimited | where {$_.RecipientTypeDetails -eq "LinkedMailbox"} | Add-MailboxPermission -User <SERVICE_ACCOUNT> -AccessRights FullAccess -Automapping $false
```

Here <SERVICE_ACCOUNT> is the name of the account used to scan the mailboxes.

Mailboxes associated with disabled AD user accounts

To grant a service account "FullAccess" rights to mailboxes associated with disabled AD user accounts, run the following commands in the Exchange Management Shell:

Grant full access to a specific mailbox:

```
Add-MailboxPermission -Identity <USER_DISABLED_MAILBOX> -User <SERVICE_ACCOUNT> -AccessRights FullAccess -Automapping $false
```

Here, <USER_DISABLED_MAILBOX> is the name of the mailbox associated with a disabled AD user account, and <SERVICE_ACCOUNT> is the name of the account used to scan the mailbox.

Unsupported mailboxes and folders

Currently, DDC doesn't support the following mailbox types and folders for the Exchange Server target:

- **Disconnected mailboxes:** Disconnected mailboxes are mailboxes that have been:
 - **Disabled:** Disabled mailboxes are rendered inactive and retained until the retention period expires, while leaving associated user accounts untouched. Disabled mailboxes can only be accessed by reconnecting the owner user account to the mailbox.
 - **Removed:** Removing a mailbox deletes the associated AD user account, renders the mailbox inactive, and retains it until its retention period expires. Removed mailboxes can only be accessed by connecting it to another user account.
 - **Moved to a different mailbox database:** Moving a mailbox from one mailbox database to another leaves the associated user account untouched, but sets the state of the mailbox to "SoftDeleted". "SoftDeleted"

mailboxes are left in place in its original mailbox database as a backup, in case the destination mailbox is corrupted during the move. To access a "SoftDeleted" mailbox, connect it to a different user account or restore its contents to a different mailbox.

- **Resource mailboxes:** Resource mailboxes are mailboxes that have been assigned to meeting locations (room mailboxes) and other shared physical resources in the company (equipment mailboxes). These mailboxes are used for scheduling purposes.
- **Remote mailboxes:** Mailboxes that are set up on a hosted Exchange instance, or on Microsoft 365, and connected to an email user on an on-premises Exchange instance.
- **System mailboxes**
- **Legacy mailboxes**

Note

The following are not mailboxes, and are not supported as scan locations:

- All distribution groups.
- Mail users or mail contacts.
- Public folders.

Target limitations

Scanning the entire data store is not supported, hence you must specify a scan target path.

Sharepoint Server

This section covers the following topics:

- [Prerequisites](#)
- [SharePoint Server resources](#)
- [Add Sharepoint Server data store](#)
- [Add Sharepoint Server scan](#)

Prerequisites

Component	Description
Proxy agent	Windows agent

Component	Description
TCP allowed connections	All TCP ports used by the SharePoint web applications.
Version	SharePoint Server 2013 and higher.

SharePoint Server resources

Sharepoint Server is a web-based collaborative platform integrating natively with Microsoft Office that is provided to organizations seeking greater control over Sharepoint's behavior or design. A SharePoint Server host consists of web applications, site collections, sites, files, and folders as shown below. When a SharePoint Server is added as a scan target, DDC returns all root-level site collections for the SharePoint Server.

Example structure of SharePoint Server Host:

```
SharePoint Server Host (host name: SharePointX)
+- SharePoint Server
  +- Web Application 1 (https://sharepoint.example.com)
    +- Site Collection 1 (https://sharepoint.example.com/)
    +- Site Collection 2 (https://sharepoint.example.com/operations)
    +- Site Collection 3 (https://sharepoint.example.com/marketing)
  +- Web Application 2 (https://sharepoint.example.com:100)
    +- Site Collection 1 (https://sharepoint.example.com:100/)
    +- Site Collection 2 (https://sharepoint.example.com:100/engineering)
```

Example of structure of Web application:

```
Web Application 1 (https://sharepoint.example.com)
+- Site Collection 1 (https://sharepoint.example.com/)
+- Site Collection 2 (https://sharepoint.example.com/operations)
  +- Sub-site 1 (https://sharepoint.example.com/operations/sub-site.aspx)
  +- Folder 1 (https://sharepoint.example.com/operations/myFolder)
    +- File 1 (https://sharepoint.example.com/operations/myFolder/myFile
e.txt)
  +- Lists (https://sharepoint.example.com/operations/Lists)
    +- List 1 (https://sharepoint.example.com/operations/Lists/myList)
      +- Item 1 https://sharepoint.example.com/operations/Lists/myList/
myFile.pptx)
```

Add Sharepoint Server data store

To add the Sharepoint Server data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery and Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
4. Complete the following steps:
 - a. **Select Type & Category**

- b. General Info
- c. Configure Connection
- d. Add Access Control & Tags

Select Type & Category

1. Under **Select Data Store Category**, select **Server**.
2. From **Select Server Type**, select **Sharepoint Server**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the Sharepoint Server domain:

Field	Description
Hostname	Hostname of the Microsoft SQL Server where the SharePoint Server is hosted.
User	A SQL user configured in the SharePoint and has access to the sites that you want to scan.
Password	Password used for the SQL user.
API Passwords	If multiple credentials are required to access the different site collections or sites, click the Browse

Field	Description
	button and upload the text file containing granular access credentials for sites and site collections. Text file format: Each line in the file defines a credential set for a URL path formatted as <code><url_path> <username> <password></code>. Here, ▪ <code><url_path></code> is the URL path to a Site Collection or Site. If it is left blank, the credentials will be used to access all content in the SharePoint Server. ▪ <code><username></code> is the user name that has access to the URL path. ▪ <code><password></code> is the password for the corresponding user. Note DDC will default to the credentials provided in the User and Password fields for sites or site collections that are not specified in the text file.

Note

Use credentials that have the minimum required privileges to access all the web applications and site collections on the Sharepoint Server for successfully scanning all resources for a Sharepoint Server target. For example, to scan all the Sharepoint site collections in "Sharepoint X", use a credential set that has access to "Web Application 1" and "Web Application 2".

Tip

Windows Authentication for Microsoft SQL

For Windows authentication, enter Windows account credentials as follows:

- **User:** Windows domain and username in the `<domain_name\user_name>` format.

- **Password:** Windows password.

For more information on Windows or SQL Server authentication modes, see [Choose An Authentication Mode](#).

2. In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this

option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.

- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Sharepoint Server scan

To add a scan for the Sharepoint Server:

1. Open the **Data Discovery and Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.

- Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
- Click **Next**.

Select Data Stores

- Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
- Click **Next**.

Add Targets

- To add a scan target, do one of the following:

- Add target paths manually.

Under the **Add Target** field, specify the correct target path and click **Apply**. See [SharePoint Server resources](#) to understand the structure of targets.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Target Path to scan	Syntax	Example
Complete data store	Empty path	
Site collection	<code><organization>.sharepoint.com/<site_collection></code>	<code>https://example.sharepoint.com/operations</code>
All lists in a site collection	<code><organization>.sharepoint.com/<site_collection>/<site></code>	<code>https://example.sharepoint.com/operations/my-site</code>
Specific list in a site collection	<code><organization>.sharepoint.com/<site_collection>/:site/:list/<list></code>	<code>https://example.sharepoint.com/operations/:site/:list/my-list</code>
All folders and files in a site collection.	<code><organization>.sharepoint.com/<site_collection>/:site/:file</code>	<code>https://example.sharepoint.com/operations/:site/:file</code>
Specific folder in a site collection.	<code><organization>.sharepoint.com/<site_collection></code>	<code>https://example.sharepoint.com/</code>

Target Path to scan	Syntax	Example
	<code>>/:site/:file/<folder></code>	<code>operations/:site/:file/documents</code>
Specific file in a site collection.	<code><organization>.sharepoint.com/<site_collection>/:site/:file/<file></code>	<code>https://example.sharepoint.com/operations/:site/:file/example-file.txt</code>
Specific file within a folder in a site collection.	<code><organization>.sharepoint.com/<site_collection>/:site/:file/<folder>/<file></code>	<code>https://example.sharepoint.com/operations/:site/:file/documents/example-file.txt</code>

Note

- Target paths are case-sensitive.
- A list item in a specific list cannot be individually added and scanned. You can only scan the entire list.
- While scanning lists, attachments within SharePoint Server lists are not scanned, resulting in no sensitive data objects being detected. To scan the attachments within list, use the following target path: `<organization>.sharepoint.com/sites/<site_collection>/:site/:file/Lists/list_name/attachments`

- Navigate and add target paths.

- Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- In the left pane, navigate and select the desired target path.
- Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan a Sharepoint Server data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using wildcard characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Syntax
Site collection	<code><organization>.sharepoint.com/<site_collection></code>
Site	<code><organization>.sharepoint.com/<site_collection>/<site></code>
List	<code><organization>.sharepoint.com/<site_</code>

Filter Item	Syntax
	collection>/:site/:list/<list>
Files	<organization>.sharepoint.com/<site_collection>/:site/:file/<file>
Folders	<organization>.sharepoint.com/<site_collection>/:site/:file/<folder>

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify <string>.

Filter Item	Example
Site collection	<site_collection>/*
Site	<site>/*
List	<list>/*
Files	<file>/*
Folders	<folder>/*

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify <string>.

For example, to exclude locations that contain 'blob' in their path, use expression *blob*.

Filter Item	Example
Site collection	*<site_collection>*
Site	*<site>*
List	*<list>*
Files	*<file>*
Folders	*<folder>*

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and **End** dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual**: This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled**: Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Clouds

- [AWS S3](#)
- [Azure Blobs](#)
- [Azure Table](#)
- [Google Workspace: GDrive](#)
- [Google Workspace: Gmail](#)
- [Office 365: Exchange Online](#)
- [Office 365: OneDrive for Business](#)
- [Office 365: Sharepoint Online](#)
- [Salesforce](#)

AWS S3

This section covers the following topics:

- Prerequisites
- Encryption Methods
- Get AWS S3 Access Keys
- Add AWS S3 Data Store
- Add AWS S3 Scan

Prerequisites

Component	Description
Proxy Agent	• Direct Internet Access • Cloud service-specific access key • Recommended Proxy Agents: • Windows Agent • Linux Agent
TCP Allowed Connections	Port 443

Encryption methods

DDC supports Amazon S3 Buckets that use the following encryption methods:

- Server-side encryption with Amazon S3-managed encryption keys (**SSE-S3**)
- Server-side encryption with AWS KMS-managed keys (**SSE-KMS**)

Get AWS S3 access keys

You require AWS S3 account access key (and associated secret access key) to add AWS S3 data store in DDC for scanning.

To get AWS S3 access key:

1. Log on to [AWS Console](#).
2. On the top left corner, click **Services > IAM > Users**.
3. Search and select the IAM user with the following access permissions:
 - ListAllMyBuckets
 - ListBucket
 - GetBucketLocation

-
- `GetObject`

You can check the user's permission under **Permissions** tab. If you cannot find a user with the required permissions, [create a new user](#).

4. Go to **Security Credentials > Access keys** and note down the access key ID.

Each access key has an associated secret access key. The secret access key is provided to the user at the time of key creation.

If there are no access key listed, [create a new access key](#).

Create new user

1. Click **IAM > Users > Create user**.
2. Enter the user name and click **Next**.
3. Under **Permissions options**, select **Attach policies directly** option.
4. Under **Permissions policies**, select an existing policy having required access permissions to the Amazon S3 Buckets. Or, [create a new permission policy](#).
5. Under **Review and create**, click **Create user**.

Create access key

You can have a maximum of two access keys at a time.

To create an access key:

1. In the **Access keys** section, click **Create access key**.
2. Under **Access key best practices & alternatives**, select **Other** and click **Next**.

You can choose any option as per your requirements. In this use case, **Others** is selected for long-term access key.

3. (Optional) Under **Set description tag**, add a description for the access key.
4. Click **Create access key**.

Access and Secret Access keys are created.

5. Click **Download .csv file** to download and save the access key and secret access key.

Caution

Once the **Retrieve access key** page is closed, you cannot access the secret access keys.

6. Click **Done**.

Create policy

1. Click **Create policy**.
2. On the **Specify permissions** page, use the Visual or JSON editor to add the following permissions:
 - ListAllMyBuckets
 - ListBucket
 - GetBucketLocation
 - GetObject

Example:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetObject",
        "s3:GetBucketLocation"
      ],
      "Resource": "*"
    }
  ]
}
```

3. Click **Next**.
4. On **Review and create** page, enter the policy name and click **Create policy**.

The new user policy is created and will display under **Permissions policies**.

Delete/Deactivate access key

You can deactivate or delete AWS S3 access keys for a user account.

1. In the AWS console, click **Services > IAM service > Users**.
2. Select the desired user and go to **Security Credentials** tab.

The **Access keys** section displays the existing keys.

3. Do one of the following:

- a. To deactivate an access key:

- Select **Deactivate** from the **Actions** drop-down of the desired access key.

Note

A deactivated access key still counts toward your limit of two access keys.

- b. To delete an access key:

- a. Choose **Delete** from the **Actions** drop-down of the desired access key. Follow the instructions in the dialog to first **Deactivate** the key before deleting it.
- b. In the **Delete <Access Key>** dialog, enter the access key ID.
- c. Click **Delete**.

The access key is deleted. This enables the **Create access key** button for adding new access key.

Add AWS S3 online data store

To configure the AWS S3 Online data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)

d. Add Access Control & Tags

Note

It is recommended to use IAM access keys instead of AWS root credentials.

Select Type & Category

1. Under **Select Data Store Category**, select **Cloud**.
2. From **Select Cloud Type**, select **AWS S3**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the AWS S3:

Field	Description
Access Key ID	Access Key ID obtained from your storage account administrator.
Secret Access Key	Secret Access Key ID obtained from your storage account administrator.

2. (Optional) In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add AWS S3 scan

To add a scan for SharePoint Online:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the **Add Scan** screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Full Data Store	<Empty_Path>	
Whole Bucket	<BucketName>	ddc-data
Specific folder in Bucket	<BucketName/folder_name>	ddc-data/test1
Specific file in Bucket	<BucketName/folder_name/ filename>	ddc-data/test1/emp-info.txt

Note

Target paths are case-sensitive, except for the bucket name. Failure to use the correct case will result in a completed scan, but the report will not contain any match count for data objects and sensitive data.

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.
 - c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
 - d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store.
You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an AWS S3 data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Syntax	Example
Bucket	<BucketName>	ddtestdata Wildcard usage ddtestdata* — Applies all paths starting with 'ddtestdata' as prefix.
Folder	<BucketName>/<FolderName>	ddtestdata/tests

Filter Item	Syntax	Example
Sub folder	<BucketName>/<FolderName>/<SubFolderName>	ddctestdata/tests/data
File	<BucketName>/<FolderName>/<SubFolderName>/<File>	ddctestdata/tests/data/file.txt

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify **<string>**.

Filter Item	Syntax	Example
Bucket	<BucketName>*	ddctestdata* — Applies all paths starting with 'ddctestdata' as suffix
Folder	<BucketName>/<FolderName>*	ddctestdata/tests* — Applies all paths starting with 'ddctestdata/tests' as suffix.
Sub folder	<BucketName>/<FolderName>/<SubFolderName>*	ddctestdata/tests/data* — Applies all paths starting with 'ddctestdata/tests/data' as suffix.
File	<FileName>	file.txt Wildcard usage *.txt — Applies all paths ending with '.txt' as suffix.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify **<string>**.

For example, to exclude locations that contain 'blob' in their path, use expression *blob*.

Filter Item	Syntax	Example
Bucket	*<BucketName>*	*ddctestdata*
Folder	*<FolderName>*	*tests*
Sub folder	*<FolderName>/<SubFolderName>*	*tests/data*
File	*<BucketName>/<FolderName>/	*ddctestdata/tests/data/file1* or

Filter Item	Syntax	Example
	<FileName>* or *<FileName>*	*file1*

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify the **Start** and **End** dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule run

1. Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Azure Blobs

This section covers the following topics:

- [Prerequisites](#)
- [Get Azure Account Access Keys](#)
- [Add Azure Blobs Data Store](#)
- [Add Azure Blobs Scan](#)

Prerequisites

Component	Description
Proxy Agent	• Direct Internet Access • Cloud service-specific access key • Recommended Proxy Agents: • Windows Agent • Linux Agent
TCP Allowed Connections	Port 443

Get Azure Account access keys

You require Azure account access keys to add Azure Blobs data store in DDC for scanning.

To get Azure Account Access keys:

1. Log on to your Azure account.
2. Select **All resources**.
3. Use the search explorer to search for *storage account*.
4. Select the desired storage account.
5. Select **Access keys**.
6. Click **Show** to display the key values.

Note down the key values for **key1 (Primary)** and **key2 (Secondary)**.

Note

Only one access key can be active at a time. Ask your Azure Storage account administrator which access key is currently active, and use that key to connect DDC to your Azure Storage account.

Add Azure Blobs data store

To configure the Azure Blobs data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
4. Complete the following steps:

-
- a. **Select Type & Category**
 - b. **General Info**
 - c. **Configure Connection**
 - d. **Add Access Control & Tags**

Note

It is recommended to use IAM access keys instead of AWS root credentials.

Select Type & Category

1. Under **Select Data Store Category**, select **Cloud**.
2. From **Select Cloud Type**, select **Azure Blobs**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the Azure Blobs:

Field	Description
Account Name	Azure account name.
User	Azure storage account name.
Active Access Key	Azure storage account access keys (key1 or key2). See Get Azure Account Access keys .

- In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

- (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
- Click **Next**.

Add Access Control & Tags

- (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
- Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - All groups:** All groups of users can access the data store through reports. This is the default setting.

- **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Azure Blobs scan

Azure Blobs data store support multi-agent scans.

To add a scan for Azure Blobs:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Apply Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the **Add Scan** screen.

General Info

1. Specify a **Name** for the scan.

2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Full Data Store	<Empty>	
Specific folder	<foldername>	test-dataset
Specific file	<folder_name/file_name.txt>	test-dataset/american-express.txt

Note

Target paths are case-sensitive. Incorrect casing will lead to a failed scan.

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.
 - c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.

- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store.
You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Apply Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an Azure Blobs data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. Specify `<string>`. It can be used to exclude entire directory trees.

Filter Item	Syntax	Example
Container	<ContainerName>	ddctest Wildcard usage ddtest* — Applies all paths starting with 'ddtest' as prefix.

Filter Item	Syntax	Example
Blob	<ContainerName>/<BlobName>	ddctest/reuters/reut2-012.sgm Wildcard usage *reut2-012.sgm — Applies all paths ending with 'reut2-012.sgm' as prefix.

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify <string>.

Filter Item	Syntax	Example
Container	<ContainerName>* or *<ContainerName>*	my_data* — Applies all paths starting with 'my_data' as suffix. *my_data* — Applies all paths containing 'my_data' as suffix.
Blob	<ContainerName>/<BlobName> or <BlobName>	ddctestblob/reut2 or my_data Wildcard usage my_data* — Applies all paths starting with 'my_data' as suffix.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify <string>.

For example, to exclude locations that contain 'blob' in their path, use expression *blob*.

Filter Item	Syntax	Example
Container	*<ContainerName>* or <ContainerName>*	ddctestdata* or *test*
Blob	*<BlobName>*	*reut2*

- **Include DO modified recently**

Includes data object modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and **End** dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Azure Table

This section covers the following topics:

- [Prerequisites](#)
- [Get Azure Account Access Keys](#)
- [Add Azure Table Data Store](#)
- [Add Azure Table Scan](#)

Prerequisites

Component	Description
Proxy Agent	• Direct Internet Access • Cloud service-specific access key • Recommended Proxy Agents: • Windows Agent • Linux Agent
TCP Allowed Connections	Port 443

Get Azure Account access keys

You require Azure account access keys to add Azure Table data store in DDC for scanning.

To get Azure Account Access keys:

1. Log on to your Azure account.
2. Select **All resources**.
3. Use the search explorer to search for *storage account*.
4. Select the desired storage account.
5. Select **Access keys**.
6. Click **Show** to display the key values.

Note down the key values for **key1(Primary)** and **key2(Secondary)**.

Note

Only one access key can be active at a time. Ask your Azure Storage account administrator which access key is currently active, and use that key to connect DDC to your Azure Storage account.

Add Azure Table data store

To configure the Azure Table data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Note

It is recommended to use IAM access keys instead of AWS root credentials.

Select Type & Category

1. Under **Select Data Store Category**, select **Cloud**.
2. From **Select Cloud Type**, select **Azure Table**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the Azure Table:

Field	Description
Account Name	Azure account name.
Authentication User	Azure storage account name.
Password	Azure storage account access keys (key1 or key2). See Get Azure Account Access keys .

2. In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store.

Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups**: All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s**: Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Azure Table scan

Azure table data store support multi-agent scans.

To add a scan for Azure Table:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Apply Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of tabs of the **Add Scan** screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Full Data Store	<empty_path>	
Specific table	<tablename>	testdata

Note

Target paths are case-sensitive, except for the bucket name. Failure to use the correct case will result in a successful scan, but the report will not contain any match count for data objects and sensitive data.

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.
 - c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
 - d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store.
You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Apply Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an Azure table data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. Specify `<string>`. It can be used to exclude entire directory trees.

Filter Item	Syntax	Example
Table	<code><TableName></code>	ddcdata

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify `<string>`.

Filter Item	Syntax	Example
Table	<TableName>*	ddcdata* — Applies all paths starting with 'ddcdata' as suffix.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example, to exclude locations that contain 'blob' in their path, use expression `*blob*`.

Filter Item	Syntax	Example
Table	<TableName>*	ddcdata*

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Note

Unsupported filters for Azure tables data store:

- Include DO's within modification date
- Exclude DO greater than size
- Include DO modified recently

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Google Workspace: GDrive

This section covers the following topics:

- [Prerequisites](#)
- [Configure Google Workspace Account](#)
- [Add Drive Data Store](#)
- [Add Drive Scan](#)

Prerequisites

Component	Description
Proxy Agent	Proxy Agent host with direct internet access. Recommended Proxy Agents: • Windows Agent • Linux Agent
TCP Allowed Connections	Port 443

Configure Google Workspace account

Before you add Drive targets, you must have:

- A Drive administrator account for the target Drive domain.
- A Drive account. DDC only supports Drive accounts. Personal Google accounts are not supported.

To configure your Google Workspace account for scanning:

1. [Select a Project](#)
2. [Enable APIs](#)
3. [Create a Service Account](#)
4. [Set up Domain-Wide Delegation](#)
5. [Generate a New Key for the Service Account](#)

Select a project

1. Log on to the [Google API Console](#).
2. From the projects list, select the project that you want to scan with DDC. If no project exists, create a new project.

Enable APIs

To scan a Drive account, enable the APIs for it in your selected project.

To enable the APIs:

1. In the left pane, click **Enabled APIs & services**.
2. On the **APIs & Services**, click **+ ENABLE APIS AND SERVICES**.
3. On the **API Library** page, search for the following APIs:

Target Drive	API Library
All	Admin SDK API
Google Drive	Drive API

4. Click **ENABLE** for both the APIs.

Create a service account

Before adding Drive as a target, you must create a Google service account for use with DDC. The service account must have the required permissions to allow DDC to authenticate and access (scan) the resources in your Drive account.

1. Log on to the [Google Cloud Console](#).
2. Select the desired project from the project list.
3. Click the hamburger icon (≡) to expand the navigation menu.
4. Go to **IAM & Admin > Service Accounts**.
5. Click **+CREATE SERVICE ACCOUNT**.
6. In the **Service account details** section, specify the following details:

Field	Description
Service account name	Enter a descriptive name for the service account, for example, ddc-sa .
(Optional) Service account ID	Edit the default ID for the service account, or click the refresh button to generate a service account ID. For example, ddc-sa@project-id.iam.gserviceaccount.com . This will be required when configuring connection for the Drive data store.
(Optional) Description	Provide a description for the new service account.

7. Click **CREATE AND CONTINUE**.
8. In the **Grant this service account access to project** section, from the **Select a role** drop-down list, select **Project > Owner**.
9. Click **CONTINUE** and then **Done**. Your service account is created.
10. Navigate back the **Service accounts** page.
11. Click the newly created service account.
12. Click the **DETAILS** tab.
13. Note down the **Unique ID** (or **OAuth 2 Client ID**) for the service account (for example, **123456789012345678901**). This is required when [setting up domain-wide delegation](#).

Set up domain-wide delegation

After creating a service account, you need to set up and enable domain-wide delegation to allow DDC to access your Drive domain with the service account.

To delegate domain-wide authority to a service account:

1. Log on to [Google Admin Console](#) as a super administrator of the Drive domain.
2. Click the hamburger icon (≡) to expand the navigation menu.
3. Go to **Security > Access and data control > API controls**.
4. Click **MANAGE DOMAIN WIDE DELEGATION**.
5. Click **Add New**. The Add a new customer ID dialog box is displayed.
6. In the **Customer ID** field, enter the service account's **Client ID** or **Unique ID** (for example, **123456789012345678901**).
7. In the **OAuth scopes (comma delimited)** field, enter a comma-separated list of Google API scopes for every Drive service that you want to scan with DDC.

Google Service	Google API OAuth 2.0 Scope
All (required)	https://www.googleapis.com/auth/admin.directory.user.readonly
Google Drive	https://www.googleapis.com/auth/drive.readonly

For example, specify <https://www.googleapis.com/auth/admin.directory.user.readonly>, <https://www.googleapis.com/auth/drive.readonly>.

8. Click **AUTHORIZE**.

Generate a new key for the service account

Credentials for a service account are set to expire in the year 10000, making expiration unlikely. However, a contingency plan exists for regenerating a new key, if needed. If the key is not obtained or expires, it's acceptable to create a new key, although not an ideal practice.

1. Log on to the Google Cloud Platform (GCP) with your credentials.
2. Go to the **DASHHBOARD**.
3. Click the hamburger icon (☰) to expand the navigation menu.
4. Click **IAM & Admin > Service Accounts**.
5. Search for the project in which the required service account is available.
6. Click the ellipses icon (⋮) and click **Manage keys**. The Keys page is displayed.
7. Click **ADD KEY**, then **Create new key** to generate a new key. The Create private key for "<Your service account>" dialog box is displayed.
8. Select **P12** as the **Key type** and click **CREATE**.

The key will be automatically downloaded to your system. Save the created P12 private key file to a secure location on your computer. This will be required when [configuring connection](#) for the Drive data store.

Add Drive data store

To add the Drive data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Cloud**.
2. From **Select Cloud Type**, select **G-Drive**.

3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name:** Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name:** Location of the data store.
 - **Add Location:** Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store:** Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the Google domain.

Field	Description
Google Domain	Google domain that you want to scan. For example, if your Drive administrator email address is <code>admin@example.com</code> , your Google domain is <code>example.com</code> .
User	Email address of the Drive administrator account. For example, <code>admin@example.com</code> .
Service Account ID	Your service account ID or email address. For example, <code>ddc-sa@project-id.iam.gserviceaccount.com</code> . Refer to Create a Service Account to get the service account ID.
Private Key	Private key (*.p12) associated with the Google Workspace service account. Refer to Generate a New Key for the Service Account for the key details.

2. In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Drive scan

To add a scan for Drive:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the **Add Scan** screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Complete data store	<Empty_Path>	
Specific user account	<User_Name>	testuser
Specific folder in user account	<user_name/My Drive/ folder_name>	testuser/My Drive/sensitivedata
Specific file inside a folder	<user_name/My Drive/ folder_name/file_name>	testuser/My Drive/sensitivedata/ foo.txt

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.
 - c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
 - d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan a Drive data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

▪ Exclude Path/DO by prefix

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Syntax	Example
User	<code><UserName></code>	datastoreiciduser
Folder	<code><UserName>/<FolderName></code>	datastoreiciduser/my drive/sensitivedata
File	<code><UserName>/<FolderName>/<Filename></code>	datastoreiciduser/my drive/sensitivedata/file.txt

▪ Exclude Path/DO by suffix

Excludes paths or data objects that end with a given string. Specify `<string>`.

Filter Item	Syntax	Example
User	<code><UserName>*</code>	datastoreiciduser* — Applies all paths starting with

Filter Item	Syntax	Example
		'datastoreicduser' as suffix.
Folder	<FolderName>*	folder* — Applies all paths starting with 'folder' as suffix.
File	<FileName>	file1 Wildcard usage file1* — Applies all paths starting with 'file1' as suffix.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify <string>.

For example, to exclude locations that contain 'blob' in their path, use expression *blob*.

Filter Item	Syntax	Example
User	<UserName>*	datastoreicduser*
Folder	<UserName>/<FolderName>* or *<FolderName>*	datastoreicduser/my drive/some_folder* or *folder*
File	<UserName>/<FolderName>/<FileName>* or *<FileName>*	datastoreicduser/my drive/some/folder/file1* or *file1*

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter,specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and **End** dates.

2. Click **Apply**.

3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.

4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Google Workspace: Gmail

This section covers the following topics:

- Prerequisites
- [Configure Google Workspace Account](#)
- Add Gmail Data Store
- Add Gmail Scan

Prerequisites

Component	Description
Proxy Agent	Proxy Agent host with direct internet access. Recommended Proxy Agents: • Windows Agent • Linux Agent
TCP Allowed Connections	Port 443

Configure Google Workspace account

Before you add Gmail targets, you must have:

- A Gmail administrator account for the target Gmail domain.
- A Gmail account. DDC only supports Gmail accounts. Personal Google accounts are not supported.

To configure your Google Workspace account for scanning:

1. [Select a Project](#)

2. [Enable APIs](#)
3. [Create a Service Account](#)
4. [Set up Domain-Wide Delegation](#)
5. [Generate a New Key for the Service Account](#)

Select a project

1. Log on to the [Google API Console](#).
2. From the projects list, select the project that you want to scan with DDC. If no project exists, create a new project.

Enable APIs

To scan a Gmail account, enable the APIs for it in your selected project.

To enable the APIs:

1. In the left pane, click **Enabled APIs & services**.
2. On the **APIs & Services**, click **+ ENABLE APIS AND SERVICES**.
3. On the **API Library** page, search for the following APIs:

Target Gmail	API Library
All	Admin SDK API
Google Mail	Gmail API

4. Click **ENABLE** for both the APIs.

Create a service account

Before adding Gmail as a target, you must create a Google service account for use with DDC. The service account must have the required permissions to allow DDC to authenticate and access (scan) the resources in your Gmail account.

1. Log on to the [Google Cloud Console](#).
2. Select the desired project from the project list.
3. Click the hamburger icon (≡) to expand the navigation menu.
4. Go to **IAM & Admin > Service Accounts**.
5. Click **+CREATE SERVICE ACCOUNT**.

6. In the **Service account details** section, specify the following details:

Field	Description
Service account name	Enter a descriptive name for the service account, for example, <code>ddc-sa</code> .
(Optional) Service account ID	Edit the default ID for the service account, or click the refresh button to generate a service account ID. For example, <code>ddc-sa@project-id.iam.gserviceaccount.com</code> . This will be required when configuring connection for the Gmail data store.
(Optional) Description	Provide a description for the new service account.

7. Click **CREATE AND CONTINUE**.
8. In the **Grant this service account access to project** section, from the **Select a role** drop-down list, select **Project > Owner**.
9. Click **CONTINUE** and then **Done**. Your service account is created.
10. Navigate back the **Service accounts** page.
11. Click the newly created service account.
12. Click the **DETAILS** tab.
13. Note down the **Unique ID** (or **OAuth 2 Client ID**) for the service account (for example, `123456789012345678901`). This is required when [setting up domain-wide delegation](#).

Set up domain-wide delegation

After creating a service account, you need to set up and enable domain-wide delegation to allow DDC to access your Gmail domain with the service account.

To delegate domain-wide authority to a service account:

1. Log on to [Google Admin Console](#) as a super administrator of the Gmail domain.
2. Click the hamburger icon (≡) to expand the navigation menu.
3. Go to **Security > Access and data control > API controls**.
4. Click **MANAGE DOMAIN WIDE DELEGATION**.
5. Click **Add New**. The Add a new customer ID dialog box is displayed.
6. In the **Customer ID** field, enter the service account's **Client ID** or **Unique ID** (for example,

123456789012345678901).

7. In the **OAuth scopes (comma delimited)** field, enter a comma-separated list of Google API scopes for every Gmail service that you want to scan with DDC.

Google Service	Google API OAuth 2.0 Scope
All (required)	https://www.googleapis.com/auth/admin.directory.user.readonly
Google Mail	https://mail.google.com/

For example, specify `https://www.googleapis.com/auth/admin.directory.user.readonly,https://mail.google.com/`.

8. Click **AUTHORIZE**.

Generate a new key for the service account

Credentials for a service account are set to expire in the year 10000, making expiration unlikely. However, a contingency plan exists for regenerating a new key, if needed. If the key is not obtained or expires, it's acceptable to create a new key, although not an ideal practice.

1. Log on to the Google Cloud Platform (GCP) with your credentials.
2. Go to the **DASHHBOARD**.
3. Click the hamburger icon (≡) to expand the navigation menu.
4. Click **IAM & Admin > Service Accounts**.
5. Search for the project in which the required service account is available.
6. Click the ellipses icon (⋮) and click **Manage keys**. The Keys page is displayed.
7. Click **ADD KEY**, then **Create new key** to generate a new key. The Create private key for "<Your service account>" dialog box is displayed.
8. Select **P12** as the **Key type** and click **CREATE**.

The key will be automatically downloaded to your system. Save the created P12 private key file to a secure location on your computer. This will be required when [configuring connection](#) for the Gmail data store.

Add Gmail data store

To add the Gmail data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.

3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Cloud**.
2. From **Select Cloud Type**, select **Google-workspace: Gmail**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the Google domain.

Field	Description
Google Domain	Google domain that you want to scan. For example, if your Gmail administrator email address is <code>admin@example.com</code> , your Google domain is

Field	Description
	example.com.
User	Email address of the Gmail administrator account. For example, admin@example.com.
Service Account ID	Your service account ID or email address. For example, enterprise-recon-sa@project-id.iam.gserviceaccount.com. Refer to Create a Service Account to get the service account ID.
Private Key	Private key (*.p12) associated with the service account. Refer to Generate a New Key for the Service Account for the key details.

- In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

- (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
- Click **Next**.

Add Access Control & Tags

- (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
- Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Gmail scan

To add a scan for Gmail:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)

f. Schedule Run

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax
Complete data store	<Empty_Path>
User account	<User_Name>
Folder in user account	<user_name/folder_name>

Note

- Target paths are not case-sensitive.
- To scan the mailbox `user_name@example.com`, enter `user_name`. To scan a specific folder in the mailbox, specify `user_name/<folder>`. For example:

- To scan the **Inbox** folder, enter `user_name/inbox`.
- To scan the **Sent Mail** folder, enter `user_name/sent`.

- Navigate and add target paths.

- a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- b. In the left pane, navigate and select the desired target path.
- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan a Gmail data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. Can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Syntax	Example
User/Account	<UserName>	datastoreicduser
Folder	<UserName>/<FolderName>	datastoreicduser/inbox Wildcard usage *inbox — Applies all paths ending with 'inbox' as prefix Note Labels are not supported. To filter labels, label ID is required.
File	<UserName>/<FolderName>/<FileNameName>	▪ datastoreicduser/inbox/mydata filter test data/Mon, 23 Aug 2021 10:54:40 +0530/50-contacts.csv ▪ datastoreicduser/inbox/test filter test data ▪ datastoreicduser/Label_4800715244570918733/test filter test data/ Wildcard usage ▪ datastoreicduser/inbox/mydata filter test data/*/50-contacts.csv — Applies all paths starting with 'datastoreicduser/inbox/mydata

Filter Item	Syntax	Example
		filter test data/' and ending with '/50-contacts.csv' as prefix. datastoreicduser/*/mydata filter test data — Applies all paths starting with 'datastoreicduser/' and ending '/mydata filter test data' as prefix

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify `<string>`.

Filter Item	Syntax	Example
User/Account	<code><UserName>*</code>	datastoreicduser* — Applies all paths starting with 'datastoreicduser' as suffix.
Folder	<code><FolderName>*</code>	inbox* — Applies all paths starting with 'inbox' as suffix.
File	<code><FileName></code>	50-contacts.csv Wildcard usage datastoreicduser/inbox/mydata filter test data/*/50-contacts.csv — Applies all paths starting with 'datastoreicduser/inbox/mydata filter test data/' and ending with '/50-contacts.csv' as suffix. 50-contacts.csv* — Applies all paths starting with '50-contacts.csv' as suffix.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example, to exclude locations that contain 'blob' in their path, use expression `*blob*`.

Filter Item	Syntax	Example
User/Account	<UserName>*	datastorecicduser*
Folder	*<FolderName>*	*folder*
File	<UserName>/<FolderName>/<Subject>/<Date>/<FileName> or <UserName>/<FolderName>/<Subject>*/<FileName>	- datastorecicduser/inbox/mydata filter test data/Mon, 23 Aug 2021 10:54:40 +0530/50-contacts.csv - datastorecicduser/inbox/mydata filter test data/*/50-contacts.csv

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and **End** dates.

- **Exclude DO greater than size**

This filter is not supported in this release.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Note

API request default quota for Gmail is **15,000 per minute**. If this limit is exceeded, API request will fail and scan run may encounter different issues.

Office 365: Exchange Online

This section covers the following topics:

- Prerequisites
- Configure Microsoft 365 Account
- Add Exchange Online Data Store
- Add Exchange Online Scan
- Unsupported Mailbox Types and Folders

Prerequisites

Component	Description
Proxy Agent	Proxy Agent host with direct internet access. Recommended Proxy Agents: • Windows Agent • Linux Agent
TCP Allowed Connections	Port 443

Configure Microsoft 365 account

This topic describes how to generate the client ID, tenant key ID, and client secret key. These are required to register DDC with Microsoft 365 account. The topic also describes how to grant API permissions to DDC to access specific resource APIs.

1. Generate Client ID and Tenant ID
2. Generate Client Secret Key
3. Grant API Access

Generate client ID and tenant ID

To generate the client ID and tenant ID:

1. Log on to the [Azure app registration portal](#) as an administrator account. After you log on successfully, you are redirected to the home page.

2. In the left pane, select the **Microsoft Entra ID** service. Alternatively, you can search for **Microsoft Entra ID** in the search bar.
3. Under **Manage**, click **App registrations**. The App registrations page is displayed.
4. Click **+ New registration**. The Register an application page is displayed.
5. Specify the following:
 - **Name:** Display name for your application.
 - **Supported account types:** Select **Accounts in this organizational directory only**.
6. Click **Register**. The Overview page of the newly registered application is displayed.
7. Note down the values of **Application (client) ID** and **Directory (tenant) ID**. These values will be required when [configuring a connection](#) for the data store.

Generate client secret key

1. Under **Manage**, click **Certificates & secrets**.
2. On the **Client secrets** tab, click **+ New client secret**. The Add a client secret dialog box is displayed.
3. Specify the following:
 - **Description:** Description for the client secret key.
 - **Expires:** Validity for the client secret key. The recommended validity is 180 days (6 months).
4. Click **Add**. The client secret key is added. The Value column displays the client secret key.
5. Copy the client secret key and save it to a secure location. This secret key will be required when [configuring a connection](#) for the data store.

Warning

Save the client secret key in a secure location. You cannot access this client secret key after you navigate away from the page.

Grant API access

To scan Exchange Online targets, you need to grant permissions to DDC to access specific resource APIs.

1. Under **Manage**, click **API permissions**.
2. In the **Configured permissions** section, click **+ Add a permission**. The Request API permissions dialog box

is displayed.

3. On the **Microsoft APIs** tab, click **Microsoft Graph > Application permissions**.
4. Select the following API permissions for your app:

Permissions	Description
• Group.Read.All • User.Read.All • Directory.Read.All • Mail.Read • Contacts.Read • Calendars.Read	Permissions required for scanning Exchange Online targets.
• Group.ReadWrite.All • User.ReadWrite.All • Directory.ReadWrite.All • Mail.ReadWrite • Contacts.ReadWrite • Calendars.ReadWrite	Permissions required for remediating Exchange Online targets.

Tip

To add permissions:

- a. In the search box, type the permission you want to add.
- b. Select the check box to the left of the permission. Similarly, select all the required permissions.
- c. Click **Add permissions**.

5. In the **Configured permissions** section, click **Grant admin consent for <organization name>**. The Grant admin consent confirmation dialog box is displayed.
6. Click **Yes**.

The Status column for all the newly added API permissions is updated to **Granted for <organization name>**.

Add Exchange Online data store

To add the Exchange Online data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.

3. Click **Data Stores > Data Stores > Add Data Store**. The **Add Data Store** screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Cloud**.
2. From **Select Cloud Type**, select **Office 365: Exchange Online**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the Exchange Online domain:

Field	Description
Exchange Online Domain	Microsoft 365 domain to scan (for example, <code>example.onmicrosoft.com</code>). Only accounts where the user principal name (UPN)

Field	Description
	shares the same domain as specified in the Exchange Online Domain field will be scanned. For example, if Exchange Online Domain is set to <code>example.onmicrosoft.com</code>, <code>user1@example2.onmicrosoft.com</code> will not be scanned even if the user belongs to a group in the <code>example.onmicrosoft.com</code> domain. To scan multiple domains within your organization's Microsoft 365 environment, add these domains as separate Exchange Online targets.
Client ID	Client ID (application ID) of Exchange Online. For example, <code>clientid-1234-5678-abcd-6d05bf28c2bf</code>. Refer to Generate Client ID and Tenant ID for the client ID.
Client Secret Key	Exchange Online client secret key. For example, <code>client~secret.key-CHvV1B5YQfr~6zDjEyv</code>. Refer to Generate Client Secret Key for the client secret key.
Tenant ID	Tenant ID of Exchange Online. Your Microsoft 365 tenant ID is a globally unique identifier (GUID) that is different than your organization name or domain. For example, <code>tenantid-1234-abcd-5678-02011df316f4</code>. Refer to Generate Client ID and Tenant ID for the tenant ID.

- In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

- (Optional) In the **Add Label** field, enter a to restrict the agents that can scan this specific data store instance.

You can also remove an existing label. See [Labels](#) for more details.

4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups**: All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s**: Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Note

- Only accounts where the user principal name (UPN) shares the same domain as specified in the Exchange Online Domain field will be scanned.
- To scan multiple domains, you need to add the domains in a new data store and add this data store in the scan setting.
- A group can have an email associated with it. If the group has a domain similar to the configured Exchange Online Domain, then only the group will be scanned.
- A group may not have an email configured. The group will be scanned if the users in the group have the same domain as configured in Exchange Online Domain. Users with a different domain won't be scanned.

For example, security groups don't need an email, they will still be scanned. However, the users that belong to this group should have the same domain as the group domain to be scanned. Users that belong to this group, but have a different domain, will not be scanned.

- A group, which has the same email as the group domain, may contain users with different domains. In this case, only the users that share the same domain as the group domain will be scanned.

Add Exchange Online scan

To add a scan for the Exchange Online:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The **Add Scan** screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Complete data store	<Empty_Path>	
All user accounts in a specific group	<Group Display Name>	TEST_GROUP
Specific user account in group	<Group Display Name>/<User Principal Name>	TEST_GROUP/ user1@example.onmicrosoft.com
Specific folder for user account in group (e.g. Calendar, Contacts, Notes etc.)	<Group Display Name>/<User Principal Name>/<Mailbox Folder>	TEST_GROUP/ user1@example.onmicrosoft.com /Inbox
All user accounts	<All Users>	All Users
Specific user account (Recommended for user accounts that do not belong to any group.)	< All Users/User Principal Name	All Users/ user1@example.onmicrosoft.com
Specific folder for user account	<All Users/<User Principal	All Users/

Target Path to Scan	Syntax	Example
(e.g. Calendar, Contacts, Notes etc.) (Recommended for user accounts that do not belong to any group.)	Name>/<Mailbox Folder>>	user1@example.onmicrosoft.com /Inbox

Note

- Target paths are case-sensitive. Target paths can be groups user principal name and specific folders and files.
- If multiple Microsoft 365 groups with the same display name exist in your domain, then only the first instance of the group will be retrieved. For example, if three groups have the same display name, "TESTGROUP", then the first "TESTGROUP" group for the Exchange Online target will be scanned.

- Navigate and add target paths.

- a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- b. In the left pane, navigate and select the desired target path.
- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification profiles.

2. Click **Next**.

Add Filters

This step is optional.

Note

Filters only apply to the scan entities found until the specified folder name in the target path. It doesn't include anything inside that folder or beyond it.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an Exchange Online data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Example
Group	All Users
User/Account	All Users/sample@sjcpl.onmicrosoft.com Wildcard usage *sample@sjcpl.onmicrosoft.com — Applies all paths ending with 'sample@sjcpl.onmicrosoft.com' as prefix.
Folder	All Users/sample@sjcpl.onmicrosoft.com/inbox Wildcard usage *inbox — Applies all paths ending with 'inbox' as prefix. Note: Folder name is not case-sensitive.

Filter Item	Example
Attachment	- All Users/sample@sjcpl.onmicrosoft.com/Inbox/Mail a/2021-02-22T06:40:18Z/maildir-a.zip - All Users/sample@sjcpl.onmicrosoft.com/folder_name/subject — This filters out a specific mail and all its content with a corresponding subject name. - All Users/sample@sjcpl.onmicrosoft.com/Inbox/Mail a/2021-02-22T06:40:18Z Wildcard usage - All Users/sample@sjcpl.onmicrosoft.com/Inbox/Mail a/*/maildir-a.zip — Applies all paths starting with 'All Users/sample@sjcpl.onmicrosoft.com/Inbox/Mail a/' and ending with '/maildir-a.zip' as prefix. It is recommended to the use string to avoid manually checking email's date and time and converting it to required format. *maildir-a.zip — Applies all paths ending with 'maildir-a.zip' as prefix. This filters out all data objects with attachment maildir-a.zip. *subject — Applies all paths ending with 'subject' as prefix.

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify `<string>`.

Filter Item	Example
Group	All Users* — Applies all paths starting with 'All Users' as suffix.
User/Account	sample@sjcpl.onmicrosoft.com* — Applies all paths starting with 'sample@sjcpl.onmicrosoft.com' as suffix *sample@sjcpl.onmicrosoft.com* — Applies all paths containing 'sample@sjcpl.onmicrosoft.com' as suffix.
Folder	inbox* — Applies all paths starting with 'inbox' as suffix.
Attachment	maildir-a.zip Wildcard usage All Users/sample@sjcpl.onmicrosoft.com/Inbox/

Filter Item	Example
	Mail a/*/maildir-a.zip* — Applies all paths starting with 'All Users/sample@sjcpl.onmicrosoft.com/Inbox/Mail a/' and containing '/maildir-a.zip' anywhere in the followed path as suffix *maildir-a.zip* — Applies all paths containing 'maildir-a.zip' as suffix.

▪ Exclude Path/DO by expression

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example, to exclude locations that contain 'blob' in their path, use expression `*blob*`.

Filter Item	Example
Group	All Users* (Use a trailing * to exclude a given location.)
User/Account	All Users/sample@sjcpl.onmicrosoft.com* or *sample@sjcpl.onmicrosoft.com*
Folder	All Users/sample@sjcpl.onmicrosoft.com/inbox* or *inbox*
Attachment	- All Users/sample@sjcpl.onmicrosoft.com/Inbox/Mail a/2021-02-22T06:40:18Z/maildir-a.zip or - All Users/sample@sjcpl.onmicrosoft.com/Inbox/Mail a/*\maildir-a.zip* or *maildir-a.zip*

▪ Include DO modified recently

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

▪ Exclude DO greater than size

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

▪ Include DO's within modification date

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and **End** dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual**: This is the default option. Select this option run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled**: Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Note

API request default quota for SharePoint Exchange Online is **800 per minute**. If this limit is exceeded, API request will fail and scan run may encounter different issues.

Unsupported mailboxes and folders

Currently, DDC doesn't support the following mailbox types and folders for the Exchange Online target:

- Inactive mailboxes
- Disabled mailboxes
- Unlicensed mailboxes

Office 365: OneDrive for Business

This section covers the following topics:

- [Prerequisites](#)
- [Configure Microsoft 365 Account](#)
- [Add OneDrive for Business Data Store](#)
- [Add OneDrive for Business Scan](#)
- [Unsupported Types and Folders](#)

Prerequisites

Component	Description
Proxy Agent	Proxy Agent host with direct internet access. Recommended Proxy Agents: • Windows Agent • Linux Agent
TCP Allowed Connections	Port 443

Configure Microsoft 365 account

This topic describes how to generate the client ID, tenant key ID, and client secret key. These are required to register DDC with Microsoft 365 account. The topic also describes how to grant API permissions to DDC to access specific resource APIs.

1. [Generate Client ID and Tenant ID](#)
2. [Generate Client Secret Key](#)
3. [Grant API Access](#)

Generate client ID and tenant ID

To generate the client ID and tenant ID:

1. Log on to the [Azure app registration portal](#) as an administrator. After you log on successfully, you are redirected to the home page.
2. In the left pane, select the **Microsoft Entra ID** service. Alternatively, you can search for **Microsoft Entra ID** in the search bar.
3. Under **Manage**, click **App registrations**. The App registrations page is displayed.
4. Click **+ New registration**. The Register an application page is displayed.
5. Specify the following:
 - **Name:** Display name for your application.
 - **Supported account types:** Select **Accounts in this organizational directory only**.
6. Click **Register**. The Overview page of the newly registered application is displayed.
7. Note down the values of **Application (client) ID** and **Directory (tenant) ID**. These values will be required when [configuring a connection](#) for the data store.

Generate client secret key

1. Under **Manage**, click **Certificates & secrets**.
2. On the **Client secrets** tab, click **+ New client secret**. The Add a client secret dialog box is displayed.
3. Specify the following:
 - **Description:** Description for the client secret key.
 - **Expires:** Validity for the client secret key. The recommended validity is 180 days (6 months).
4. Click **Add**. The client secret key is added. The Value column displays the client secret key.
5. Copy the client secret key and save it to a secure location. This secret key will be required when [configuring a connection](#) for the data store.

Warning

Save the client secret key in a secure location. You cannot access this client secret key after you navigate away from the page.

Grant API access

To scan OneDrive for Business targets, you need to grant permissions to DDC to access specific resource APIs.

1. Under **Manage**, click **API permissions**.
2. In the **Configured permissions** section, click **+ Add a permission**. The Request API permissions dialog box is displayed.
3. On the **Microsoft APIs** tab, click **Microsoft Graph > Application permissions**.
4. Select the following API permissions for your app:

Permissions	Description
• Group.Read.All • GroupMember.Read.All • Directory.Read.All • Files.Read.All • Sites.Read.All	Permissions required for scanning OneDrive for Business targets.

Tip

To add permissions:

- a. In the search box, type the permission you want to add.
- b. Select the check box to the left of the permission. Similarly, select all the required permissions.
- c. Click **Add permissions**.

5. In the **Configured permissions** section, click **Grant admin consent for <organization name>**. The Grant admin consent confirmation dialog box is displayed.
6. Click **Yes**.

The Status column for all the newly added API permissions is updated to **Granted for <organization name>**.

Add OneDrive for Business data store

To add the OneDrive for Business data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Cloud**.
2. From **Select Cloud Type**, select **Office 365: OneDrive for Business**.
3. Click **Next**.

General Info

1. Specify the following details:

- **Data Store Name:** Name for the data store.
- **Description** (Optional): Description for the data store.
- **Location Name:** Location of the data store.
- **Add Location:** Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
- **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
- **Enable Data Store:** Whether to enable the newly added data store. Select the check box to enable the data store.

2. Click **Next**.

Configure Connection

1. Specify the credentials of the OneDrive for Business domain:

Field	Description
OneDrive for Business Domain	Microsoft 365 domain to scan (for example, <code>example.onmicrosoft.com</code>). Only accounts where the user principal name (UPN) shares the same domain as specified in the OneDrive for Business Domain field will be scanned. For example, if OneDrive for Business Domain is set to <code>example.onmicrosoft.com</code>, <code>user1@example2.onmicrosoft.com</code> will not be scanned even if the user belongs to a group in the <code>example.onmicrosoft.com</code> domain. To scan multiple domains within your organization's Microsoft 365 environment, add these domains as separate OneDrive for Business targets.
Client ID	Client ID (application ID) of OneDrive for Business. For example, <code>clientid-1234-5678-abcd-6d05bf28c2bf</code>. Refer to Generate Client ID and Tenant ID for the client ID.
Client Secret Key	OneDrive for Business client secret key. For example, <code>client~secret.key-</code>

Field	Description
	CHvV1B5YQfr~6zDjEyv. Refer to Generate Client Secret Key for the client secret key.
Tenant ID	Tenant ID of OneDrive for Business. Your Microsoft 365 tenant ID is a globally unique identifier (GUID) that is different than your organization name or domain. For example, <code>tenantid-1234-abcd-5678-02011df316f4</code> . Refer to Generate Client ID and Tenant ID for the tenant ID.

- In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

- (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
- Click **Next**.

Add Access Control & Tags

- (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
- Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Note

- Only accounts where the user principal name (UPN) shares the same domain as specified in the OneDrive for Business Domain field will be scanned.
- To scan multiple domains, you need to add the domains in a new data store and add this data store in the scan setting.
- A group can have an email associated with it. If the group has a domain similar to the configured OneDrive for Business Domain, then only the group will be scanned.
- A group may not have an email configured. The group will be scanned if the users in the group have the same domain as configured in OneDrive for Business Domain. Users with a different domain won't be scanned.

For example, security groups don't need an email, they will still be scanned. However, the users that belong to this group should have the same domain as the group domain to be scanned. Users that belong to this group, but have a different domain, will not be scanned.

- A group, which has the same email as the group domain, may contain users with different domains. In this case, only the users that share the same domain as the group domain will be scanned.

Add OneDrive for Business scan

To add a scan for the Exchange Online:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Complete data store	<Empty_Path>	
All user accounts	<All Users>	All Users
All user accounts in a specific group	<Group Display Name>	TEST_GROUP
Specific user account in group	<Group Display Name>/<User Principal Name>	TEST_GROUP/ user1@example.onmicrosoft.com
Specific folder for user account in group	<Group Display Name>/<User Principal Name>/<Folder>	TEST_GROUP/ user1@example.onmicrosoft.com /ProjectA
Specific file for user account in group	<Group Display Name>/<User Principal Name>/<Folder>/<File>	TEST_GROUP/ user1@example.onmicrosoft.com /ProjectA/file.txt

Note

- Target paths are case-sensitive. A target path can be groups user principal name and specific folders and files.
- If multiple Microsoft 365 groups with the same display name exist in your domain, then only the first instance of the group will be retrieved. For example, if three groups have the same display name, "TESTGROUP", then the first "TESTGROUP" group for the OneDrive for Business target will be scanned.

See [Target limitations](#) for more details.

- Navigate and add target paths.

- Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- In the left pane, navigate and select the desired target path.
- Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan a OneDrive for Business data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. Can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Syntax
Group	<code><Group_Name></code>
User	<code><Group_Name>/<user@example.onmicrosoft.com></code>
Folder	<code><Group_Name>/<user@example.onmicrosoft.com>/<foldername></code>

Filter Item	Syntax
File	<Group_Name>/<user@example.onmicrosoft.com>/<foldername>/<filename>

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify <string>.

Filter Item	Syntax
Group	<Group_Name>
User	<Group_Name>/<user@example.onmicrosoft.com>
Folder	<Group_Name>/<user@example.onmicrosoft.com>/<foldername>
File	<Group_Name>/<user@example.onmicrosoft.com>/<foldername>/<filename>

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify <string>.

For example, to exclude locations that contain 'blob' in their path, use expression *blob*.

Filter Item	Syntax
Group	<Group_Name>* (You have to use a trailing * to exclude a given location.)
User	<All Users>/<sample@sjcpl.onmicrosoft.com>* or *<sample@sjcpl.onmicrosoft.com>*
Folder	<All Users>/<sample@sjcpl.onmicrosoft.com>/<folder>* or *<folder>*
File	<All Users>/<sample@sjcpl.onmicrosoft.com>/<folder>/<file>* or *<file>*

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and **End** dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Note

API request default quota for OneDrive is **1000 per minute**. If this limit is exceeded, API request will fail and scan run may encounter different issues.

Unsupported types and folders

Currently, DDC doesn't support the following types and folders for the OneDrive for Business target:

- Notebooks
- OneNote file types and folders stored in OneDrive for Business but outside the default Notebooks folder.

Target limitations

It is not possible to scan all groups from the root location. This is because accounts are located in multiple groups, and scanning all from the root would result in scanning many locations multiple times. For this reason, the user is required to at least specify groups for the scan.

Office 365: SharePoint Online

This section covers the following topics:

- Prerequisites
- Configure SharePoint Online
- Add SharePoint Online Data Store
- Add SharePoint Online Scan
- Deleted SharePoint Online Sites
- Troubleshooting

Prerequisites

Component	Description
Proxy Agent	Recommended Proxy Agents: • Windows Agent • Linux Agent • FreeBSD Agent
TCP Allowed Connections	Port 443

Configure SharePoint Online

These steps need to be performed on Microsoft SharePoint Online.

Before you can add SharePoint Online as a target, you need to register and configure it for use with DDC. Also, grant the necessary permissions to the SharePoint Add-in to authenticate and access (scan) resources within your SharePoint Online environment.

1. Register the SharePoint Add-in
2. Grant Permissions to the SharePoint Add-in

Note

Microsoft has announced the retirement of SharePoint Add-ins as detailed in their [Retirement Announcement for Add-ins](#). From CipherTrust Manager 2.23 onwards, the existing SharePoint Online data store configuration method will no longer be supported, and you must reconfigure the data store according to Microsoft's updated prerequisites and permissions model.

Register the SharePoint Add-in

To register the SharePoint Add-in:

1. Log on to SharePoint Online.
2. Go to the AppRegNew form at `<site collection url>/_layouts/15/AppRegNew.aspx`. For example, `https://mycompany.sharepoint.com/_layouts/15/AppRegNew.aspx`.
3. In the **AppRegNew** form, specify the following:

Field	Description
Client Id	Enter a unique lowercase string. For example, <code>1234abcd-56ef-78gh-90ij-1234clientid</code> . Alternatively, you can click Generate to generate a client ID.
Client Secret	Click Generate to generate a client secret. For example, <code>abcdefghijklmnopqrstuvwxyz0123456789klmnopqrst0clientsecret</code> .
Title	Enter a descriptive name for the add-in. For example, <code>DDC SPO add-in</code> .
App Domain	The hostname of the remote component of the SharePoint Add-in. For example, <code>www.example.com</code> . This field is mandatory when registering the SharePoint Add-in. However, this field is not required for scanning the SharePoint Online Targets.
Redirect URI	The endpoint in the remote application or service to which Azure Access Control service (ACS) sends an authentication code. For example, <code>https://www.example.com/default.aspx</code> . This field is mandatory when registering the SharePoint Add-in. However, this field is not required for scanning the SharePoint Online Targets.

4. Click **Create**. The page reloads and displays the details of the newly registered SharePoint Add-in.
5. Note down the **Client ID** and **Client Secret Key**. These values will be required when configuring a connection for the data store.

Grant permissions to the SharePoint Add-in

1. Log on to SharePoint Online as an administrator.
2. Go to the tenant administration site at `<tenant>-admin.sharepoint.com/_layouts/15/appinv.aspx`. For example, `https://mycompany-admin.sharepoint.com/_layouts/15/appinv.aspx`.

3. In the **App ID** field, enter the Client ID of the registered SharePoint Add-in. This is the Client ID you noted down in [Register the SharePoint Add-in](#).
4. Click **Lookup**. It should fetch and populate the Title, App Domain, and Redirect URL fields.
5. In the **Permission Request XML** field, enter the following permissions:

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
  <AppPermissionRequest Scope="http://sharepoint/content/tenant" Right="FullControl"/>
  <AppPermissionRequest Scope="http://sharepoint/content/sitecollection" Right="Read"/>
  <AppPermissionRequest Scope="http://sharepoint/content/sitecollection/web" Right="Read"/>
</AppPermissionRequests>
```

6. Click **Create**. A permission consent dialog box is displayed.
7. Click **Trust It** to grant permissions to the SharePoint Add-in.
8. Go to the **Site App Permissions** page at `<tenant>-admin.sharepoint.com/_layouts/15/appprincipals.aspx?Scope=Web`. For example, `https://xyz-admin.sharepoint.com/_layouts/15/appprincipals.aspx?Scope=Web`.
9. Under the **App Display Name** column, look for the registered SharePoint Add-In.
10. Note down the value of **Tenant ID** from Application Identifier. This value will be required when configuring SharePoint Online as a Target.

```
# App Identifier format: i:0i.t|ms.sp.ext|<Client ID>@<Tenant ID>
i:0i.t|ms.sp.ext|1234abcd-56ef-78gh-90ij-1234clientid@12345678-abcd-9012-efgh-ijkltenantid
```

Where:

- Client ID = `1234abcd-56ef-78gh-90ij-1234clientid`
- Tenant ID = `12345678-abcd-9012-efgh-ijkltenantid`

Add SharePoint Online data store

To add the SharePoint Online data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.

3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Cloud**.
2. From **Select Cloud Type**, select **Office 365: Sharepoint Online**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the SharePoint Online domain:

Field	Description
Domain	Name of the SharePoint Online organization. For example, if you access SharePoint Online at https://mycompany.sharepoint.com , then

Field	Description
	mycompany is the domain.
Client ID	Client ID of the registered SharePoint Add-in. For example, 1234abcd-56ef-78gh-90ij-1234clientid. Refer to Register the SharePoint Add-in for the client ID.
Client Secret Key	Client Secret key of the registered SharePoint Add-in. For example, abcdefghij0123456789klmnopqrst0clientsecret. Refer to Register the SharePoint Add-in for the client secret key.
Tenant ID	Tenant ID of the registered SharePoint Add-in. For example, 12345678-abcd-9012-efgh-ijkltenantid. Refer to Grant Permissions to the SharePoint Add-in for the tenant ID.

- (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.

Note

DDC doesn't support selection of multiple agents for the SharePoint Online data store.

- Click **Next**.

Add Access Control & Tags

- (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
- Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add SharePoint Online scan

To add a scan for SharePoint Online:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Scan all site collections, sites, lists, list items, folders and files.	<Empty_Path>	
Scan a site collection. Includes all sites, lists, list items, folders, and files for the site collection.	<organization>.sharepoint.com/sites/<site_collection>	https://example.sharepoint.com/sites/operations
Scan a site in a site collection.	<organization>.sharepoint.com/sites/<site_collection>/<site>	https://example.sharepoint.com/sites/operations/my-site
Scan all lists in a site collection.	<organization>.sharepoint.com/sites/<site_collection>/:site/:list	https://example.sharepoint.com/sites/operations/:site/:list
Scan a specific list in a site collection.	<organization>.sharepoint.com/sites/<site_collection>/:site/:list/<list>	https://example.sharepoint.com/sites/operations/:site/:list/my-list
Scan all folders and files in a site collection.	<organization>.sharepoint.com/sites/<site_collection>/:site/:file	https://example.sharepoint.com/sites/operations/:site/:file

Target Path to Scan	Syntax	Example
Scan a specific folder in a site collection.	<organization>.sharepoint.com/sites/<site_collection>/:site:/file/<folder>	https://example.sharepoint.com/sites/operations/:site:/file/Shared Documents/documents
Scan a specific file in a site collection.	<organization>.sharepoint.com/sites/<site_collection>/:site:/file/<file>	https://example.sharepoint.com/sites/operations/:site:/file/Shared Documents/my-file.txt
Scan a specific file within a folder in a site collection.	<organization>.sharepoint.com/sites/<site_collection>/:site:/file/<folder>/<file>	https://example.sharepoint.com/sites/operations/:site:/file/Shared Documents/documents/my-file.txt

Note

While scanning lists, attachments within SharePoint Online lists are not scanned, resulting in no sensitive data objects being detected. To scan the attachments within list, use the following target path: <organization>.sharepoint.com/sites/<site_collection>/:site:/file/Lists/list_name/attachments

- Navigate and add target paths.

- a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- b. In the left pane, select the desired target path.

To view subfolders within the folder hierarchy, select the desired folder and click **List**.

- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an Office365 SharePoint Online data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

▪ Exclude Path/DO by prefix

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Syntax
Site collection	<code><organization>.sharepoint.com/sites/<site_collection></code>
Site	<code><organization>.sharepoint.com/sites/<site_collection>/<site></code>
List	<code><organization>.sharepoint.com/sites/<site_collection>/<site>/<list></code>
File	<code><organization>.sharepoint.com/sites/<site_collection>/<site>/<list>/<file></code>
Folder	<code><organization>.sharepoint.com/sites/<site_collection>/<site>/<list>/<folder></code>

▪ Exclude Path/DO by suffix

Excludes paths or data objects that end with a given string. Specify `<string>`.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes search locations and nested locations that matches the given expression. Specify **<string>**.

For example, to exclude locations that contain 'blob' in their path, use expression *blob*.

Filter Item	Syntax
Site collection	<organization>.sharepoint.com/<site_collection>* or *<site_collection>*
Site	<organization>.sharepoint.com/<site_collection>/<site>* or *<site>*
List	<organization>.sharepoint.com/<site_collection>/<site>/<list>* or *<list>*
File	<organization>.sharepoint.com/<site_collection>/<site>/<list>/<file>* or *<file>*
Folder	<organization>.sharepoint.com/<site_collection>/<site>/<list>/<folder>* or *<folder>*

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and **End** dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:

- **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
- **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Note

API request default quota for SharePoint Online is **600 per minute**. If this limit is exceeded, API request will fail and scan run may encounter different issues.

Deleted SharePoint Online sites

In SharePoint Online, deleted sites or site collections are retained for 93 days in the site Recycle Bin, unless deleted permanently. However, if you try to scan a deleted site, it will result in following error when attempting to scan them:

The target for Data Store Sharepoint Online has not been found.

Troubleshooting

While adding the SharePoint Online data store to DDC, you might encounter the following error:

The target for Data Store SharePoint scan for sites does not have access permissions.

Cause

This error occurs because the `grant app permission` is disabled by default on SharePoint Online.

Solution

For the SharePoint Add-in to work, the `DisableCustomAuthenticationApp` setting for the tenants needs to be set to `false`, as described below:

1. Open PowerShell.
2. Run `Install-Module -Name Microsoft.Online.SharePoint.PowerShell`.
3. Run `$adminUPN="<full email address of a SharePoint administrator account>"`.

For example:

```
$adminUPN="example@democompany.onmicrosoft.com"
```

4. Run `$orgName="<name of your Office 365 organization>"`.

For example:

```
$orgName="democompany"
```

5. Run `$userCredential = Get-Credential -UserName $adminUPN -Message "<password>"`.

For example:

```
$userCredential = Get-Credential -UserName $adminUPN -Message "demopassword@123"
```

6. Run `Connect-SPOService -Url https://$<orgName>-admin.sharepoint.com -Credential $userCredential`.

For example:

```
Connect-SPOService -Url https://$democompany-admin.sharepoint.com -Credential $userCredential
```

Note

The `Connect-SPOService -Url https://$<orgName>-admin.sharepoint.com -Credential $userCredential` command might return the following error. This error occurs when Multifactor Authentication (MFA) is enabled.

```
Connect-SPOService : Identity Client Runtime Library (IDCRL) did not get
a response from the Login server.
At line:1 char:1
+ Connect-SPOService -Url https://trial8349-admin.sharepoint.com -Crede
...
+ CategoryInfo          : NotSpecified: ([Connect-SPOService], IdcrLException
+ FullyQualifiedErrorId : Microsoft.SharePoint.Client.IdcrLException,Mic
rosoft.Online.SharePoint.PowerShell.ConnectSPOService
```

To work around this issue:

- a. Rerun `Connect-SPOService -Url https://$<orgName>-admin.sharepoint.com` (without `-Credential $userCredential`). You will be prompted for the Office 365 authentication.

- b. Enter the Office 365 credentials.

7. Run `set-spotenant -DisableCustomAppAuthentication $false`.

Salesforce

This section covers the following topics:

- [Prerequisites](#)
- [Configure Salesforce Account](#)
- [Add Salesforce Data Store](#)
- [Add Salesforce Scan](#)
- [Salesforce API Limits](#)
- [Unsupported Salesforce Standard Objects](#)

Prerequisites

Component	Description
Proxy Agent	Proxy Agent host with direct internet access. Recommended Proxy Agents: • Windows Agent • Linux Agent
TCP Allowed Connections	Port 443

Configure Salesforce account

This topic describes how to generate a digital certificate and the private key. These are required to register DDC with a Salesforce account.

1. [Generate Certificate and Private Key](#)
2. [Create Connected App](#)

Generate certificate and private key

To scan the Salesforce targets, you need a digital signature associated with a digital certificate and the private key.

To generate the digital certificate and the private key:

1. Open a terminal or the Windows Command prompt.
2. Install the OpenSSL package and run the following command:

```
openssl req -x509 -sha256 -nodes -newkey rsa:2048 -days <number of days> -keyout <*.key private key file> -out <*.crt certificate file> openssl req -x509 -sha256 -nodes -newkey rsa:2048 -days 365 -keyout ddc-salesforce.key -out ddc-salesforce.crt
```

Parameter	Description
(Optional) days	Number of days to certify the certificate for. The default is 30 days.
keyout	Name of the output file to write the private key to. For example, <code>ddc-salesforce.key</code> .
out	Name of the file to write the digital certificate to. For example, <code>ddc-salesforce.crt</code> .

You will be prompted to enter the required details.

3. Specify the following details:

Prompt	Answer
Country Name (2 letter code) [AU]	Your country's two letter country code (ISO 3166-1 alpha-2).
State or Province Name (full name) [Some-State]	State or province name.
Locality Name (for example, city) []	City name or name of region.
Organization Name (for example, company) [Internet Widgits Pty Ltd]	Name of organization.
Organizational Unit Name (for example, section) []	Name of organizational department.
Common Name (for example, server FQDN or YOUR name) []	Fully qualified domain name of the Master Server.
Email Address []	Email address of organization's contact person.

OpenSSL generates two output files:

- The private key (for example, `ddc-salesforce.key`) required to [set up](#) and scan a Salesforce target.
- The digital certificate (for example, `ddc-salesforce.crt`) required to [create a connected app](#) for DDC.

Create connected app

Instructions for the Salesforce Lightning and Salesforce Classic interfaces are different. Click the desired tab to view the instructions.

Salesforce Lightning

To create a connected app for DDC in Salesforce:

1. Log on to your organization's Salesforce site as an administrator.
2. Got to **Setup**.
3. On the **Home** tab, in the **Quick Find** field, enter **App Manager** and select **App Manager**.
4. On the **Lightning Experience App Manager** page, click **New Connected App**.
5. In the **Basic Information** section, specify the following details:

Field	Description
Connected App Name	Descriptive display name for DDC, for example, DDC .
API Name	Unique identifier for referring to the app programmatically, for example, DDC .
Contact Email	Email address that Salesforce can use to contact you about the connected app.

6. In the **API (Enable OAuth Settings)** section, select the **Enable OAuth Settings** check box.
7. In the **Callback URL** field, enter the URL to redirect to after successful authorization of the connected app.

Note

The **Callback URL** is a compulsory field for setting up a connected app but is not required for scanning the Salesforce targets with DDC.

8. Select **Use digital signatures** and click **Choose File** to upload a digital certificate, for example, `er-salesforce.crt`. Refer to [Generate Certificate and Private Key](#) for details.
9. Under **Select OAuth Scopes**, select and **Add** the following permissions for the **DDC** connected app:

Available OAuth Scopes	Description
• Access the identity URL service (id, profile, email,	Required for probing, scanning, and remediating

Available OAuth Scopes	Description
address, phone) • Manage user data via APIs (api) • Perform requests at any time (refresh_token, offline_access)	Salesforce targets.

10. Click **Save** and click **Continue**.

11. On the **Manage Connected Apps** page, go to **API (Enable OAuth Settings) > Consumer Key** and click **Copy**.

A sample consumer key is

1234567890.ThisIsTheConsumerKeyForTheDDCConnectedAppForSalesforce_1234567.

Save the consumer key at a secure location. This key will be required when [configuring the connection](#) for the Salesforce data store.

12. Click **Manage > Edit Policies**.

13. Under **OAuth Policies > Permitted Users**, select **Admin approved users are pre-authorized**.

14. Click **Save**.

15. Return to the **App Manager** page, go to the **Profiles** section, and click **Manage Profiles**.

16. On the **Application Profile Assignment** page, select the profile(s) (for example, **System Administrator**) that you want to allow to access the **DDC** connected app.

Note

The Salesforce account that is specified when you Set Up and Scan a Salesforce Target must be assigned to at least one of the profiles that has:

- Access to the DDC connected app (for example, "DDC"), and
- Minimum "Read" permissions for the Salesforce Objects to be scanned

Refer to [Salesforce Help - Object Permissions](#) for details.

17. Click **Save**.

18. On the **Setup > Home** tab, in the **Quick Find** field, enter **Profiles** and select **Profiles**.

19. Go to the profile you [selected](#) earlier (for example, **System Administrator**) and click **Edit**.

20. In the **Administrative Permissions** section, select the following:

- API Enabled
- Query All Files

Note

Enabling the **Query All Files** permission is an optional step. This permission allows the Salesforce account to scan all files in your organization's Salesforce site, including those owned / managed by other user accounts. You will specify this Salesforce account when [configuring a connection](#) for the data store.

For more information about Salesforce behavior when **Query All Files** is enabled, refer to [ContentVersion](#).

Without the **Query All Files** permission, **DDC** will can only scan the files that are owned by / shared to the specified Salesforce account.

21. Click **Save**.

Salesforce Classic

To create a connected app for DDC in Salesforce:

1. Log on to your organization's Salesforce site as an administrator.
2. Got to **Setup**.
3. In the left pane, under the **Build** section, expand **Create** and click **Apps**.
4. In the **Connected Apps** section, click **New**.
5. In the **Basic Information** section, specify the following details:

Field	Description
Connected App Name	Descriptive display name for DDC, for example, DDC .
API Name	Unique identifier for referring to the app programmatically, for example, DDC .
Contact Email	Email address that Salesforce can use to contact you about the connected app.

6. In the **API (Enable OAuth Settings)** section, select the **Enable OAuth Settings** check box.
7. In the **Callback URL** field, enter the URL to redirect to after successful authorization of the connected app.

Note

The **Callback URL** is a compulsory field for setting up a connected app but is not required for scanning the Salesforce targets with DDC.

8. Select **Use digital signatures** and click **Choose File** to upload a digital certificate, for example, `er-salesforce.crt`. Refer to [Generate Certificate and Private Key](#) for details.
9. Under **Select OAuth Scopes**, select and **Add** the following permissions for the **DDC** connected app:

Available OAuth Scopes	Description
• Access the identity URL service (id, profile, email, address, phone) • Manage user data via APIs (api) • Perform requests at any time (refresh_token, offline_access)	Required for probing, scanning, and remediating Salesforce targets.

10. Click **Save** and click **Continue**. You will be redirected to app details page.
11. Go to **API (Enable OAuth Settings) > Consumer Key and Secret >** and click **Manage Consumer Details > Copy Consumer Key**.

A sample consumer key is
`1234567890.ThisIsTheConsumerKeyForTheDDCConnectedAppForSalesforce_1234567`.
Save the consumer key at a secure location. This key will be required when [configuring the connection](#) for the Salesforce data store.
12. Click **Back to Manage Connected Apps > Manage > Edit Policies**.
13. Under **OAuth Policies > Permitted Users**, select **Admin approved users are pre-authorized**.
14. Click **Save**.
15. Go to the **Profiles** section, and click **Manage Profiles**.
16. On the **Application Profile Assignment** page, select the profile(s) (for example, **System Administrator**) that you want to allow to access the **DDC** connected app.

Note

The Salesforce account that is specified when you set up and scan a Salesforce target must be

assigned to at least one of the profiles that has:

- Access to the DDC connected app (for example, "DDC"), and
- Minimum "Read" permissions for the Salesforce Objects to be scanned

Refer to [Salesforce Help - Object Permissions](#) for details.

17. Click **Save**.
18. Navigate to **Setup**. In the left pane, under the **Administer** section, expand **Manage Users** and select **Profiles**.
19. Go to the profile you [selected](#) earlier (for example, **System Administrator**) and click **Edit**.
20. In the **Administrative Permissions** section, select the following:
 - API Enabled
 - Query All Files

Note

Enabling the **Query All Files** permission is an optional step. This permission allows the Salesforce account to scan all files in your organization's Salesforce site, including those owned / managed by other user accounts. You will specify this Salesforce account when [configuring a connection](#) for the data store.

For more information about Salesforce behavior when **Query All Files** is enabled, refer to [ContentVersion](#).

Without the **Query All Files** permission, **DDC** will can only scan the files that are owned by / shared to the specified Salesforce account.

21. Click **Save**.

Add Salesforce data store

To add the Salesforce data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.

3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Cloud**.
2. From **Select Cloud Type**, select **Salesforce**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials of the Salesforce domain:

Field	Description
Account Name	Email address linked with the Salesforce account to scan. Production

Field	Description
	• Syntax: <code><email_address></code> • Example: <code>admin@example.com</code> Sandbox • Syntax: <code>sandbox:<email_address></code> • Example: <code>sandbox:admin@example.com</code>
Consumer Key	Key obtained when creating the connected app. For example, <code>1234567890.ThisIsTheConsumerKeyForTheDDCConnectedAppForSalesforce_1234567</code> . This key was created when creating the connected app .
Private Key	Cloud service-specific access keys, for example, <code>ddc-salesforce.key</code> . Refer to Generate Certificate and Private Key for the private key.

2. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.

Note

DDC doesn't support selection of multiple agents for the Salesforce data store.

3. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:

- **All groups:** All groups of users can access the data store through reports. This is the default setting.
- **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Salesforce scan

To add a scan for the Salesforce:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.

2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under the **Add Target** field, specify the correct target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists target paths and syntax to specify them with examples.

Salesforce Object Type	Syntax	Example
Complete data store	<Empty_Path>	
All standard objects	s	s
All custom objects	c	c
All big objects	b	b
Specific standard object	s/<object API name>	s/Account
Specific custom object	c/<object API name>	c/Account__c
Specific big object	b/<object API name>	b/Account__b

Note

Target paths are case-sensitive.

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level.

Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.

- b. In the left pane, navigate and select the desired target path.
- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple data stores, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an Salesforce data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects that begin with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Syntax	Example
Folder	<FolderName>	C
Object	<FolderName>/<ObjectName>	c/usecase_data

- **Exclude Path/DO by suffix**

Excludes paths or data objects that end with a given string. Specify **<string>**.

Note

You cannot exclude targets, folders, and objects using suffix filter. To exclude items within any object, provide the complete path or ending string of the item path.

Example - If target path is *c/Usecase_data3__c* and actual path of the object item is *c/Usecase_data3__c/a0H5G00000R5GygUAF/AMX 346761705951424*, then provide **AMX 346761705951424** as suffix to exclude this object item.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify **<string>**.

For example, to exclude locations that contain 'blob' in their path, use expression ***blob***.

Filter Item	Syntax	Example
Folder	*<FolderName>*	*C*
Object	*<FolderName>/<ObjectName>*	*c/ddc_c*

- **Include DO modified recently**

Includes data objects modified within **N** number of days from the current date, where the value of **N** ranges from 1 to 99 days. After selecting this filter, specify **Days from current date**.

- **Exclude DO greater than size**

Excludes data objects that are larger than a given file size (in MB). After selecting this filter, specify the file size in MB.

- **Include DO's within modification date**

Includes data objects modified within a given range of dates. After selecting this filter, specify **Start** and

End dates.

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual**: This is the default option. Select this option run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled**: Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Salesforce API limits

Salesforce imposes a limit on the total number of inbound API calls that can be made per 24-hour period for an organization. For every API call to Salesforce, DDC queries and retrieves:

- Up to 2000 records (including Big Objects), or
- A single attachment or file.

If an organization reaches its daily API request limit:

- A critical error is flagged for the Salesforce domain (or location) with the HTTP 403 error - **REQUEST_LIMIT_EXCEEDED. TotalRequest Limit Exceeded**.
- Running Salesforce scans stop executing with the Failed status. A critical error is reflected on the last Object that was scanned when the limit was reached.
- Probing a Salesforce target results in the HTTP 403 error - **REQUEST_LIMIT_EXCEEDED. TotalRequest Limit Exceeded**.

Refer to [Salesforce - API Request Limits and Allocations](#) for details.

Unsupported Salesforce Standard Objects

Currently, DDC doesn't support the following Salesforce Standard Objects:

- AccountUserTerritory2View
 - AppTabMember
-

-
- ColorDefinition
 - ContentDocumentLink
 - ContentFolderItem
 - ContentFolderMember
 - DataStatistics
 - DataType
 - DatacloudAddress
 - EntityParticle
 - FieldDefinition
 - FlexQueueItem
 - FlowVariableView
 - FlowVersionView
 - IconDefinition
 - IdeaComment
 - ListViewChartInstance
 - NetworkUserHistoryRecent
 - OutgoingEmail
 - OutgoingEmailRelation
 - OwnerChangeOptionInfo
 - PicklistValueInfo
 - PlatformAction
 - RelationshipDomain
 - RelationshipInfo
 - SearchLayout
 - SiteDetail
 - UserEntityAccess
-

- UserFieldAccess
- UserRecordAccess
- Vote

Databases

- IBM Db2
- MongoDB
- Microsoft SQL
- MySQL
- Oracle
- PostgreSQL
- SAP HANA

IBM Db2

This section covers the following topics:

- Prerequisites
- Add IBM Db2 Data Store
- Add IBM Db2 Scan

Prerequisites

Component	Description
Proxy Agent	Windows Agent Note Windows Agent built-in drivers are required to connect to a Db2 data store.
Version	IBM Db2 11.1 onwards

Add IBM Db2 data store

To add IBM Db2 data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Database**.
2. From **Select Database Type**, select **IBM DB2**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

- Specify the credentials for the IBM Db2 data store:

Field	Description
Hostname/IP	Specify a valid hostname or IP address of the database server. Hostname must be more than 2 characters long.
Port	Specify a valid port for connecting to the database server. It must be a valid integer between 1 and 65535. Default port: 50000
Database	Specify the database name. <i>Case sensitivity is applicable.</i>
User	Specify the username of database user.
Password	Specify the user password.

- (Optional) In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

- (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
- Click **Next**.

Add Access Control & Tags

- (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
- Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is

properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add IBM Db2 scan

To add a scan for IBM Db2:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)

-
- d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under **Add Target Path** field, specify the complete target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists different target paths, their syntax, and examples.

Target Path to Scan	Syntax	Example
Full Data Store	<Empty>	
Schema	<SchemaName>	DB2INST1
Table inside schema	<SchemaName>/<TableName>	DB2INST1/TEST1

Note

- Table names are case sensitive but schema names are not case sensitive.
- It is recommended to set the path in **uppercase** if the database is configured as **case-insensitive**.

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level. Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.
 - c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
 - d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store.
You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an IBM Db2 data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using](#)

Wildcard Characters to learn how wildcards work.

▪ Exclude Path/DO by prefix

Excludes paths or data objects beginning with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Syntax	Example
Database	<code><DatabaseName>:<Port></code>	testdb:50000
Schema	<code><DatabaseName>:<Port>/<SchemaName></code>	testdb:50000/DB2INST1
Table	<code><DatabaseName>:<Port>/<SchemaName>/<TableName></code>	testdb:50000/DB2INST1/TEST1
Column	<code><DatabaseName>:<Port>/<SchemaName>/<TableName>/<ColumnName></code>	testdb:50000/DB2INST1/TEST1/ DATA Wildcard usage *DATA — Applies all paths ending with 'DATA' as prefix.

▪ Exclude Path/DO by suffix

Excludes paths or data objects ending with a given string. Specify `<string>`.

Filter Item	Syntax	Example
Database	<code><DatabaseName>:<Port>*</code>	testdb:50000* — Applies all paths starting with 'testdb:50000' as suffix.
Schema	<code><SchemaName></code>	testdb Wildcard usage testdb* — Applies all paths starting with 'testdb' as suffix.
Table	<code><TableName></code>	TEST1 Wildcard usage TEST1* — Applies all paths starting with 'TEST1' as suffix.

Filter Item	Syntax	Example
Column	<ColumnName>	DATA Wildcard usage DATA* — Applies all paths starting with 'DATA' as suffix.

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify **<string>**.

For example, to exclude locations that contain 'blob' in their path, use expression *blob*.

Filter Item	Syntax	Example
Database	<DatabaseName>:<Port>*	testdb:50000*
Schema	<DatabaseName>:<Port>/<SchemaName>/*	testdb:50000/DB2INST1/*
Table	*<TableName>*	*TEST1*
Column	*<ColumnName>*	*DATA*

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Note

List of unsupported filters for IBM Db2 data stores:

- Include DO modified recently
- Exclude DO greater than size
- Include DO's within modification date

Schedule Run

1. Specify the scan run frequency. The two options are:

- **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
- **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

MongoDB

This section covers the following topics:

- [Prerequisites](#)
- [Add MongoDB Data Store](#)
- [Add MongoDB Scan](#)

Prerequisites

Component	Description
Proxy Agent	▪ Windows Agent ▪ Linux Agent
Version	MongoDB 6.0 and above

Add MongoDB data store

To configure MongoDB data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Database**.
2. From **Select Database Type**, select **MongoDB**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials for the MongoDB data store:

Field	Description
Hostname/IP	Specify a valid hostname or IP address of database server. Hostname must be more than 2 characters long.
Port	Specify a valid port for connecting to database server. It must be a valid integer between 1 and 65535. Default port: 27017
User	Specify the username.
Password	Specify the password.

2. (Optional) In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add MongoDB scan

To add a scan for MongoDB:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:

- Under **Add Target Path** field, specify the complete target path and click **Apply**.

If no specific target is added, the entire data store will be scanned. *Target paths are case-sensitive.*

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Full Data Store	<EmptyPath>	
Database	<DatabaseName>	admin
Collection	<DatabaseName>/<CollectionName>	admin/test

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level. Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.
 - c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
 - d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.

2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan a MongoDB data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

▪ Exclude Path/DO by prefix

Exclude paths or data objects beginning with a given string. It can be used to exclude entire directory trees.

Filter Item	Syntax	Example
Database	<DatabaseName>	admin
Collection	<DatabaseName>/<CollectionName>*	▪ admin/test* — applies all locations that start with 'admin\test' as prefix ▪ *test* —applies all locations that contains "test" as prefix

▪ Exclude Path/DO by suffix

Exclude paths or data objects ending with a given string, specify <string>.

Filter Item	Syntax	Example
Database	<DatabaseName>*	admin* — applies all locations that start with 'admin' as prefix.
Collection	<DatabaseName>/<CollectionName>*	admin/test* — applies all locations that start with 'admin/test' as suffix. *test* — applies all locations that contain 'test' as suffix.

▪ Exclude Path/DO by expression

This filter is majorly used with wildcard characters.

Exclude paths or data objects that matches a given expression.

Filter Item	Syntax	Example
Database	<DatabaseName>*	admin* — excludes all locations that start with 'admin'.
Collection	<DatabaseName>/<CollectionName>*	admin/test* — excludes all locations that start with 'admin\test'. *test* — excludes all locations that contain 'test'.

- Click **Apply**.
- Repeat above steps to apply multiple filters. Click **Remove** to remove any applied filter.
- Click **Next**.

Note

Unsupported filters in MongoDB data stores:

- Include DO modified recently
- Exclude DO greater than size
- Include DO's within modification date

Schedule Run

- Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

- Click **Save**.

Potential problems while scanning MongoDB data store with GridFS

Following are the several known issues specific to MongoDB while scanning GridFS database.

- You cannot specify a GridFS database collection in the scan path, even if the default scan path for MongoDB is of the format `<database/collection>`. If specified, the scan will fail with the error: "Wrong database collection in target path". You should only use the `<database>` in the scan path.
- A scan on a GridFS database only accepts two default collections with a bucket named fs: **fs.files** and **fs.chunks**. If you use another prefix/bucket name, it will not be scanned.
- Scanning with multiple agents will not work.
 - If you run a scan with multiple agents to scan complete data store, the GridFS database will be skipped and will not appear in the report under the data object list.
 - If you run scan with a specific path of the GridFS database, the scan will fail with the error: "Scan results could not be found".
- During a scan of a GridFS database, if multiple files share the same name, the scan report will combine the total number of matches for all those files and list them under that single filename, even if the file contents differ.
- In a report for a scan on a GridFS database, the list of data objects can contain both collections and files.

Microsoft SQL

This section covers the following topics:

- [Prerequisites](#)
- [Add Microsoft SQL Data Store](#)
- [Add Microsoft SQL Scan](#)

Prerequisites

Component	Description
Proxy Agent	Windows Agent Note Microsoft ODBC drivers are required for SQL Server installation on the Windows Proxy Agent, to establish connection between DDC and database. ODBC driver version 17 is required to support TLS 1.2 connections.
Database Credentials	Use the correct syntax for the Username and Password fields according to your Microsoft SQL

Component	Description
	Server authentication method. SQL Server Authentication - Username: <database_user_name> - Password: <database_user_password> Windows Authentication - Username: <Windows_domain>\<Windows_user_name> Note Windows Authentication is supported only when the Windows Proxy Agent and the Microsoft SQL database server reside on the same host.
Version	Microsoft SQL 2012 and above

Add Microsoft SQL data store

To add Microsoft SQL data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. **Select Type & Category**
 - b. **General Info**
 - c. **Configure Connection**
 - d. **Add Access Control & Tags**

Select Type & Category

1. Under **Select Data Store Category**, select **Database**.

2. From **Select Database Type**, select **Microsoft SQL**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials for the Microsoft SQL data store:

Field	Description
Hostname/IP	Specify a valid hostname or IP address of the database server. Hostname must be more than 2 characters long. <i>Case sensitivity is not applicable.</i>
Port	Specify a valid port for connecting to the database server. It must be a valid integer between 1 and 65535. Default port: 1433
Database	Specify the database name.
User	Specify the username.
Password	Specify the password.

2. (Optional) In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Microsoft SQL scan

To add a scan for Microsoft SQL:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under **Add Target Path** field, specify the complete target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists different target paths, their syntax, and examples.

Target Path to Scan	Syntax	Example
Full Data Store	<Empty>	
Schema	<SchemaName>	dbo
Table inside schema	<SchemaName>/<TableName>	dbo/sensitive_data

Note

- Table names are case sensitive but schema names are not case sensitive.

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level. Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.
 - c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
 - d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store.
You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan a Microsoft SQL data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

▪ Exclude Path/DO by prefix

Excludes paths or data objects beginning with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Syntax	Example
Schema	<code><DatabaseName>:<port>/<SchemaName></code>	ddctest:1433/dbo Wildcard usage */dbo — Applies all paths ending with '/dbo' as prefix.
Table	<code><DatabaseName>:<port>/<SchemaName>/<TableName></code>	ddctest:1433/dbo/test_table Wildcard usage */test_table — Applies all paths ending with '/test_table' as prefix. This filters out all tables with name 'test_table'.
Column	Not supported	

▪ Exclude Path/DO by suffix

Excludes paths or data objects ending with a given string. Specify `<string>`.

Filter Item	Syntax	Example
Schema	<code><SchemaName>*</code>	<code>dbo*</code> — Applies all paths starting with 'dbo' as suffix.
Table	▪ Table from specific schema: <code><SchemaName>/<TableName></code> ▪ Table from all schemas: <code><TableName></code>	<code>dbo/test_table</code> , <code>test_table</code>

▪ Exclude Path/DO by expression

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example to exclude locations that contain 'replica' in their path, use expression `*replica*`.

Filter Item	Syntax	Example
Schema	<code><DatabaseName>:<port>/<SchemaName>*</code>	<code>testddc:1433/dbo*</code>
Table	<code>*<TableName>*</code>	<code>*my_table*</code>

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Note

List of unsupported filters for Microsoft SQL data stores:

- Include DO modified recently
- Exclude DO greater than size
- Include DO's within modification date

Schedule Run

1. Specify the scan run frequency. The two options are:

- **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
- **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

MySQL

This section covers the following topics:

- [Prerequisites](#)
- [Add MySQL Data Store](#)
- [Add MySQL Scan](#)

Prerequisites

Component	Description
Proxy Agent	▪ Windows Agent ▪ Linux Agent Note Built-in drivers for Windows and Linux Agents are required to connect with MySQL data store. Built-in drivers do not support password authentication with 'caching_sha2_password' method.
Database Credentials	Use credentials that have minimum required privileges to access the databases, schemas, and tables you want to scan. Example: To scan a MySQL database, use credentials that have SELECT permissions (read permissions).
Version	MySQL 5.0 and above

Add MySQL data store

To configure MySQL data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Database**.
2. From **Select Database Type**, select **MySQL**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials for the MySQL data store:

Field	Description
Hostname/IP	Specify valid hostname or IP address of database server. Hostname must be more than 2 characters long. <i>Case sensitivity is not applicable.</i>
Port	Specify a valid port for connecting to database server. It must be a valid integer between 1 and 65535. Default port: 3306
User	Specify the username.
Password	Specify the password.

2. (Optional) In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.

4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add MySQL scan

To add a scan for MySQL:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)

f. Schedule Run

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under **Add Target Path** field, specify the complete target path and click **Apply**.

If no specific target is added, the entire data store will be scanned. *Target paths are case-sensitive.*

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Full Data Store	<EmptyPath>	
Database	▪ Pagination enabled (default): <DatabaseName> ▪ Pagination disabled: <DatabaseName(paged=false)>	test-ddc, test-ddc(paged=false)
Table inside database	<DatabaseName>/<TableName>	test-ddc/sensitive_data

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level. Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.

- c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
- d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan a MySQL data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

▪ Exclude Path/DO by prefix

Excludes paths or data objects beginning with a given string. It can be used to exclude entire directory trees. Specify `<string>`.

Filter Item	Syntax	Example
Database/Schema	<code><DatabaseName></code>	testddc
Table	<code><DatabaseName>/<TableName></code>	testddc/sensitive_data Wildcard usage *sensitive_data — Applies all

Filter Item	Syntax	Example
		paths ending with 'sensitive_data' as prefix.
Column	<DatabaseName>/<TableName>/<ColumnName>	testddc/sensitive_data/email Wildcard usage *email — Applies all paths ending with 'email' as prefix.

▪ Exclude Path/DO by suffix

Excludes paths or data objects ending with a given string. Specify `<string>`.

Filter Item	Syntax	Example
Database	<DatabaseName>*	testddc* — Applies all paths starting with 'testddc' as suffix.
Table	▪ Specific table: <TableName> ▪ Table from specific database: <DatabaseName>/<TableName>	sensitive_data, testddc/sensitive_data
Column	▪ Specific column: <ColumnName> ▪ Specific column within table: <DatabaseName>/<TableName>/<ColumnName>	email, test-ddc/sensitive_data/email

▪ Exclude Path/DO by expression

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example, to exclude locations that contain 'blob' in their path, use expression *blob*.

Filter Item	Syntax	Example
Database/Schema	<DatabaseName>*	testddc*
Table	▪ Specific table: *<TableName> ▪ Table from specific database: <DatabaseName>/<TableName>	*sensitive_data, testddc/sensitive_data

Filter Item	Syntax	Example
	<DatabaseName>/<TableName>	
Column	*<DatabaseName>/<TableName>/<ColumnName>	*testddc/sensitive_data/email

- Click **Apply**.
- Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
- Click **Next**.

Note

Unsupported filters in MySQL data stores:

- Include DO modified recently
- Exclude DO greater than size
- Include DO's within modification date

Schedule Run

- Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

- Click **Save**.

Oracle

This section covers the following topics:

- [Prerequisites](#)
- [Add Oracle Data Store](#)
- [Add Oracle Scan](#)

Prerequisites

Component	Description
Proxy Agent	- Windows Agent - Linux Agent Note Windows and Linux Agent built-in drivers are required to connect to an Oracle data store.
Libraries	Requires <code>libaio1</code> and <code>libaio-dev</code> libraries installed on Linux 3 Agent host. Command: <code>sudo apt-get install libaio1 libaio-dev</code>
Version	Oracle Database 11g and above

Add Oracle data store

To add Oracle data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. **Select Type & Category**
 - b. **General Info**
 - c. **Configure Connection**
 - d. **Add Access Control & Tags**

Select Type & Category

1. Under **Select Data Store Category**, select **Database**.
2. From **Select Database Type**, select **Oracle**.

3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name:** Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name:** Location of the data store.
 - **Add Location:** Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store:** Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials for the Oracle data store:

Field	Description
Hostname/IP	Specify a valid hostname or IP address of the database server. Hostname must be more than 2 characters long.
Port	Specify a valid port for connecting to the database server. It must be a valid integer between 1 and 65535. Default port: 1521
Schema	Specify the schema name with service name. Example: Schema with service name: HR(SERVICE_NAME=XE).
User	Specify the username of the database.
Password	Specify the password.

2. (Optional) In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add Oracle scan

To add a scan for Oracle:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under **Add Target Path** field, specify the complete target path and click **Apply**.

If no specific target is added, the entire data store will be scanned.

The following table lists different target paths, their syntax, and examples.

Target Path to Scan	Syntax	Example
Full data store	<Empty>	
Table	<TableName>	TEST1

Note

- Table names are case sensitive.
- It is recommended to set the path in **uppercase** if the database is configured as **case-insensitive**.

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level. Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.
 - c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
 - d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store.
You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan an Oracle data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

- **Exclude Path/DO by prefix**

Excludes paths or data objects beginning with a given string. Specify `<string>`.

Filter Item	Syntax	Example
Table	<code><SchemaName>(SERVICE_NAME=<sid>:<Port>)/<TableName></code>	<code>DDCADMIN(SERVICE_NAME=DDCTEST):1521/TEST</code>

- **Exclude Path/DO by suffix**

Excludes paths or data objects ending with a given string. Specify `<string>`.

Filter Item	Syntax	Example
Table	<TableName> or end characters of table name	TEST1, T1 (excludes tables ending with 'T1')

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. Specify `<string>`.

For example, to exclude locations that contain 'replica' in their path, use expression `*replica*`.

Filter Item	Syntax	Example
Table	*<TableName>*	*TEST1*

2. Click **Apply**.
3. Repeat the above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Note

List of unsupported filters for Oracle data stores:

- Include DO modified recently
- Exclude DO greater than size
- Include DO's within modification date

Schedule Run

1. Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

Known issues

Issue	Cause
You might encounter following error while configuring Oracle database that requires service name: Cannot find minimum agent for the data store	The service name is missing or incorrect. Solution: Retry configuring the database using correct service name.
Issue when scanning text files as BLOBs in Oracle	DDC can't scan any text file stored as BLOB in Oracle if the file size is greater than 4 KB.

PostgreSQL

This section covers the following topics:

- Prerequisites
- Configure PostgreSQL Server
- Add PostgreSQL Data Store
- Add PostgreSQL Scan

Prerequisites

Component	Description
Proxy Agent	▪ Windows Agent ▪ Linux Agent
Version	PostgreSQL 13 and above

Configure PostgreSQL server

By default, remote connections to the PostgreSQL server are blocked.

To establish connection with PostgreSQL data store, do either of the following:

- Install the DDC agent on the PostgreSQL database server, or
- Configure the PostgreSQL server to allow remote connections (**recommended**).

To configure the PostgreSQL server:

1. Locate the `pg_hba.conf` configuration file on the PostgreSQL database server.

On a Unix-based server, this file is usually found at this location: `/var/lib/postgresql/data directory`.

2. As a root user, open *pg_hba.conf* in a text editor.
3. Append the following lines in the configuration file.

```
# Syntax:  
# host <database_name> <postgresql_user_name> <agent_host_address> <auth-method>  
host all all all md5
```

Above configuration allows any remote client with a valid username and password to connect with the PostgreSQL server. For a more secure configuration, use specific parameter values for database, user, and IP address. For example:

```
host database_A scan_user 172.17.0.0/24 md5
```

4. Save the changes and restart PostgreSQL server.
5. (Optional) Open the *postgresql.conf* file and set the value of `#listen_addresses` parameter to `*` in the Connections and Authentication sections.

```
#listen_addresses = '*'
```

You can also assign a specific IP address of a PostgreSQL server to `listen_addresses` instead of global value (*).

Add PostgreSQL data store

To add PostgreSQL data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. **Select Type & Category**
 - b. **General Info**
 - c. **Configure Connection**
 - d. **Add Access Control & Tags**

Select Type & Category

1. Under **Select Data Store Category**, select **Database**.
2. From **Select Database Type**, select **PostgreSQL**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name**: Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name**: Location of the data store.
 - **Add Location**: Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store**: Whether to enable the newly added data store. Select the check box to enable the data store.
2. Click **Next**.

Configure Connection

1. Specify the credentials for the PostgreSQL data store:

Field	Description
Hostname/IP	Specify a valid hostname or IP address of the database server. Hostname must be more than 2 characters long.
Port	Specify a valid port for connecting to database server. It must be a valid integer between 1 and 65535. Default port: 5432
Database	Specify the database name. <i>Case sensitivity is applicable.</i>
User	Specify the username.
Password	Specify the password.

2. (Optional) In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

3. (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.
4. Click **Next**.

Add Access Control & Tags

1. (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
2. Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add PostgreSQL scan

To add a scan for PostgreSQL:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)
 - d. [Select Profiles](#)
 - e. [Add Filters](#)
 - f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:

- Under **Add Target Path** field, specify the complete target path and click **Apply**.

If no specific target is added, the entire data store will be scanned. *Target paths are case-sensitive.*

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Full Data Store	<EmptyPath>	
Schema	<SchemaName>	public
Table	<SchemaName>/<TableName>	public/test_table

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level. Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.
 - c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
 - d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store. You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan a PostgreSQL data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcards work.

▪ Exclude Path/DO by prefix

Exclude paths or data objects beginning with a given string. It can be used to exclude entire directory trees.

Filter	Syntax	Example
Database	<DatabaseName>	ddc_test
Schema	<DatabaseName>:<Port>/<SchemaName>	ddc_test:5432/public Wildcard usage ddc_test:5432/* — Applies all paths starting with 'ddc_test:5432/' as prefix.
Table	<DatabaseName>:<Port>/<SchemaName>/<TableName>	ddc_test:5432/public/test
Column	*<DatabaseName>:<Port>/<SchemaName>/<TableName>/<ColumnName>*	*ddc_test:5432/public/test/data* — Applies all paths that contains '5432/public/test' as prefix.

▪ Exclude Path/DO by suffix

Exclude paths or data objects ending with a given string, specify <string>.

Filter	Syntax	Example
Database	<DatabaseName>*	ddc_test* — Applies all locations starting with 'ddc_test' as suffix.
Schema	*<Schema>*	*public* — Applies all paths that contain 'public' as suffix.

Filter	Syntax	Example
Table	Table from a specific schema: <DatabaseName>:<Port>/<SchemaName>/<TableName> Table from complete database: <table>	ddc_test:5432/public/test, test
Column	Column from a specific table: <DatabaseName>:<Port>/<SchemaName>/<TableName>/<ColumnName> Column from complete database: <ColumnName>	ddc_test:5432/public/test/data, data

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression.

For example to exclude locations that contain 'blob' in their path, use expression *blob*.

Filter	Syntax	Example
Database	<DatabaseName>	ddc_test*
Schema	<DatabaseName>:<Port>/<SchemaName>/*	ddc_test:5432/public/*
Table	<DatabaseName>:<Port>/<SchemaName>/<TableName>	ddc_test:5432/public/test
Column	*<DatabaseName>:<Port>/<SchemaName>/<TableName>/<ColumnName>*	*ddc_test:5432/public/test/data*

2. Click **Apply**.
3. Repeat above steps to apply multiple filters. Click **Remove** to remove any applied filter.
4. Click **Next**.

Note

Unsupported filters in PostgreSQL data stores:

- Include DO modified recently
- Exclude DO greater than size
- Include DO's within modification date

Schedule Run

1. Specify the scan run frequency. The two options are:

- **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
- **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

2. Click **Save**.

SAP HANA

This section covers the following topics:

- [Prerequisites](#)
- [Add SAP HANA Data Store](#)
- [Add SAP HANA Scan](#)

Prerequisites

Component	Description
Proxy Agent	Windows Agent Note Windows Agent built-in drivers are required to connect to a SAP HANA data store. If SAP HANA ODBC drivers are installed on host agent, the Agent will use these drivers instead of built-in drivers.

Component	Description
Version	SAP HANA 2.0 SPS04 and above

Add SAP HANA data store

To add SAP HANA data store:

1. Log on to the **CipherTrust Manager** GUI.
2. Open the **Data Discovery & Classification** application.
3. Click **Data Stores > Data Stores > Add Data Store**. The Add Data Store screen is displayed.
4. Complete the following steps:
 - a. [Select Type & Category](#)
 - b. [General Info](#)
 - c. [Configure Connection](#)
 - d. [Add Access Control & Tags](#)

Select Type & Category

1. Under **Select Data Store Category**, select **Database**.
2. From **Select Database Type**, select **SAP HANA**.
3. Click **Next**.

General Info

1. Specify the following details:
 - **Data Store Name:** Name for the data store.
 - **Description** (Optional): Description for the data store.
 - **Location Name:** Location of the data store.
 - **Add Location:** Click **Add Location** to add new locations to the **Location Name** drop-down. Refer to [Adding Locations](#) for detailed steps.
 - **Sensitivity Level** (Optional): Sensitivity level for the data store. Refer to [Sensitivity Levels](#) for details.
 - **Enable Data Store:** Whether to enable the newly added data store. Select the check box to enable the data store.

- Click **Next**.

Configure Connection

- Specify the credentials for the SAP HANA data store:

Field	Description
Hostname/IP	Specify valid hostname or IP address of database server. Hostname must be more than 2 characters long. <i>Case sensitivity is applicable.</i>
Port	Specify a valid port for connecting to database server. It must be a valid integer between 1 and 65535. Default port: 30015
Database	Specify the database name.
User	Specify the username.
Password	Specify the password.

- (Optional) In the **Select Number of Agents** field, set the minimum and maximum number of agents for the data store. Refer to [Agents](#) for more information.

Warning

- As there is no limit on the number of minimum and maximum agents that you can set, you should exercise caution so that you do not impact the system performance by using too many resources for a single scan.
- You will not be able to add a datastore if the minimum number of agents cannot be assigned.
- A scan will fail if the assigned agent is unavailable after adding the datastore.
- The minimum number of agents must be less than or equal to the maximum number of agents.

- (Optional) In the **Add Label** field, enter a label to restrict the agents that can scan this specific data store instance. You can also remove an existing label. See [Labels](#) for more details.

- Click **Next**.

Add Access Control & Tags

- (Optional) Grant the **All groups (default)** access for reports. Alternatively, select a group.
- Click **Save**.

The data store is added to the Data Stores page. If the **Ready to Scan** column shows **Ready**, then data store is properly configured.

For more information on Access control and Tags, expand the section below.

Access Control & Tags

The **Access Control & Tags** tab on the **Add Data Store** screen allows you to grant access rights to your data store and add tags. More details below:

- **ACCESS CONTROL** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups:** All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- **TAGS** - Select a tag from the **Add Tag** drop-down. See the list of prebuilt tags in [Predefined tags](#) section.

Tip

- New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down.
- Add as many tags as needed.
- To remove a tag, click the close icon in the tag name.

Add SAP HANA scan

To add a scan for SAP HANA data store:

1. Open the **Data Discovery & Classification** application.
2. Click **Scans > Add Scan**. The Add Scan screen is displayed.
3. Complete the following steps:
 - a. [General Info](#)
 - b. [Select Data Stores](#)
 - c. [Add Targets](#)

- d. [Select Profiles](#)
- e. [Add Filters](#)
- f. [Schedule Run](#)

Refer to [Scans](#) for the description of sections of the Add Scan screen.

General Info

1. Specify a **Name** for the scan.
2. (optional) Add a **Description** for the scan.
3. Expand **Advanced Configuration** and specify advanced configurations such as Scan Priority, Memory Usage Limit, and Amount of Data Object Volume. Refer to [Advanced Configuration](#) for details.
4. Click **Next**.

Select Data Stores

1. Under **Data Store Name**, select the desired data store that is **Ready** for scanning. You can select multiple data stores, if required.
2. Click **Next**.

Add Targets

1. To add a scan target, do one of the following:
 - Under **Add Target Path** field, specify the complete target path and click **Apply**.

If no specific target is added, the entire data store will be scanned. *Target paths are case-sensitive.*

The following table lists target paths and syntax to specify them with examples.

Target Path to Scan	Syntax	Example
Schema	<SchemaName>	system
Table	<SchemaName>/<TableName>	system/test_table

Note

Entire data store scan is not supported for SAP HANA. You must provide at least one path to initiate a scan.

- Navigate and add target paths.
 - a. Click **Browse** to navigate target paths from the root level. Alternatively, provide an initial path in the **Add Target Path** field and click **Browse** to navigate targets from that point onward.
 - b. In the left pane, navigate and select the desired target path.
 - c. Click **Add Path** to add the target path to the right pane. Similarly, add other target paths.
 - d. Click **Add**.

Tip

Either navigate the target paths from the root level (without specifying any path in the **Add Target Path** field) or make sure you provide the correct path to navigate further locations within it.

2. Click **Next**.

Select Profiles

1. Under **Classification Profile Name**, select the desired classification profiles to search for in the data store.
You can select multiple profiles, if required. Refer to [Classification Profiles](#) for details on classification profiles.
2. Click **Next**.

Add Filters

This step is optional.

1. Select the desired filter from the **Select Filter** drop-down list.

To filter the locations to scan a SAP HANA data store, consider the following syntax.

Note

Exclude Path/DO by prefix, suffix, and expression filters support wildcard characters. See [Using Wildcard Characters](#) to learn how wildcard characters function.

- **Exclude Path/DO by prefix**

Exclude paths or data objects beginning with a given string. It can be used to exclude entire directory trees.

Filter Item	Syntax	Example
Database	<DatabaseName>	ddc_test

Filter Item	Syntax	Example
Schema	<DatabaseName>:<Port>/<SchemaName>	ddc_test:5432/system
Table	<DatabaseName>:<Port>/<SchemaName>/<TableName>	ddc_test:5432/system/table Wildcard usage ddc_test:5432/*table — Applies all locations starting with 'ddc_test:5432/' and ending with '/table' as prefix
Column	*<DatabaseName>:<Port>/<SchemaName>/<TableName>/<ColumnName>*	*ddc_test:5432/system/table/col* — Applies all paths containing 'ddc_test:5432/system/table/col' as prefix. *col* — Applies all paths containing 'col' as prefix.

- **Exclude Path/DO by suffix**

Exclude paths or data objects ending with a given string, specify `<string>`.

Filter Item	Syntax	Example
Database	<DatabaseName>*	ddc_test* —Applies all locations starting with 'ddc_test' as suffix.
Schema	*<SchemaName>*	*system* — Applies all locations containing 'system' as suffix.
Table	▪ Specific table: <TableName> ▪ Table from specific schema: <Schema>/<TableName>	test, public/test
Column	▪ Specific column: <ColumnName> ▪ Specific column in specific table: <DatabaseName>:<Port>/<SchemaName>/<TableName>/<ColumnName>	data, test/data,ddc_test:5432/public/test/data

- **Exclude Path/DO by expression**

This filter is majorly used with wildcard characters.

Excludes paths or data objects that matches the given expression. For example to exclude locations that

contain 'blob' in their path, use expression `*blob*`.

Filter Item	Syntax	Example
Database	*<DatabaseName>*	*testddc*
Schema	*<SchemaName>*	*system*
Table	*<TableName>*	*table*
Column	*<ColumnName>*	*email*, *table/col*

- Click **Apply**.
- Repeat above steps to apply multiple filters. Click **Remove** to remove any applied filter.
- Click **Next**.

Note

Unsupported filters in SAP HANA data stores:

- Include DO modified recently
- Exclude DO greater than size
- Include DO's within modification date

Schedule Run

- Specify the scan run frequency. The two options are:
 - **Manual:** This is the default option. Select this option to run the scan manually. Select the **Run Now** check box to start the scan run after you save the changes.
 - **Scheduled:** Select this option to configure the scan to run automatically at the specified time.

Refer to [Schedule Scan](#) for more details on scheduling scan runs.

- Click **Save**.

Logging

Data Discovery and Classification prints its log messages to the CipherTrust Manager logs. DDC logs are located at `/opt/keysecure/logs`. The CipherTrust Manager System administrator (**ksadmin**) can log in using SSH to retrieve CipherTrust Manager logs. Also, the DDC application administrators have access to the logs.

For more details on collecting DDC logs, refer to [Troubleshooting issues in conjunction with customer support](#).

Default logging level

By default, log level setting for DDC is INFO. With this log level set DDC prints out the INFO and ERROR level messages to the log. Among the various messages that DDC prints to the logs, the error messages and security audit messages are the most useful for troubleshooting DDC issues and securing the deployment.

Identifying DDC log messages

The microservices behind DDC are `oleander` and `sundew` and the messages coming to the CipherTrust Manager log from DDC can be identified by those names.

Additionally, `oleander` has these three modules:

- Clustering
- Agent_Selection
- Scan_watcher

Each of these modules will generate its own error messages, each in its separate log.[] log file.

The logging service responsible for collecting and processing these messages is FLUENTD. It is capable of displaying those messages to the terminal through the log command. Here's an example of such a command:

```
log | grep oleander | grep "clustering"
```

This command would display all messages coming from the `oleander`'s Clustering module.

For a complete list of error messages that DDC sends to the CipherTrust Manager logs, see the [Error Log Messages](#).

Modifying DDC ML agent log levels

The DDC ML agent generates the following logs:

Log Type	Description
<code>MLAgentReg.log</code>	ML agent registration logs
<code>agent.{timestamp}</code>	ML library logs
<code>ddc_agent_spark.log</code>	Spark logs generated during scan
<code>service.log</code>	ML Agent service logs

You can change the log level of the DDC ML agent by adding following entries in the `<INSTALL_DIR>/conf/service.conf` file and restarting the DDC ML Agent service.

- To set log level to Warning, add the `AGENT_LOG_CONFIG='{ "level": "WARN", "levels": "*=WARN", "development": false }'` entry in the file.

- To set log level to Error, add the `AGENT_LOG_CONFIG={'level':"ERR", "levels":"*=ERR", "development":false}` entry in the file.
- To set log level to Info, add the `AGENT_LOG_CONFIG={'level':"INF", "levels":"*=INF", "development":false}` entry in the file.

Security audit log messages

The DDC security audit messages can be identified by the `Oleander | INFO [security]` bit that they contain. The full format of such a log message (or log line) is:

```
<date> | Oleander | INFO [security] <event> <error (if any)> <details (if any)>
```

For example:

```
2020-06-29 | Oleander | INFO | [security] DDCScanClientUnexpectedErrorProbe "error: error probing scan client" "details: [scan_id:5432-5432-543254-2-5432]"
```

Usually, only the event type is printed out to the log (in the example above, it would be `DCScanClientUnexpectedErrorProbe`).

Enabling Syslog logging

Audit records are logged to a local database by default. This is suitable for production systems and clusters with a limited load. However, for clusters that support a large number of transactions, it is recommended to configure the CM to disable logging to a local database and enable logging using a remote Syslog server. This significantly reduces cluster traffic and disk usage. For more information, refer to the following sections in the [Thales CipherTrust Manager Administrator Guide](#):

- "Disabling local database audit logging"
- "Configuring remote Syslog server"
- [Configuring Connection to a Syslog Server](#)

Logging troubleshooting information

In addition to including INFO and ERROR log messages in CipherTrust Manager logs, DDC can generate local logs to record interactions between different components during scan run. These logs contain useful technical information that can help [Thales Customer Support](#) in identifying and resolving scan-related issues.

In case the scan fails, is stuck in the running state, or experiences any unexpected issues, you can download and share these logs with [Thales Customer Support](#) to address scan-related issues.

Note

Before running the scan, enable the [Trace logs](#) scan parameter to capture detailed scan execution information for each file.

You can download the logs starting from the scan initiation, throughout its execution, and after it finishes. These log files are encrypted and require decryption before you can share them with [Thales Customer Support](#).

To download and decrypt the logs files:

1. In the **Scans** screen, click the ellipses icon (...) corresponding to the desired scan name.
2. Select the **Download Logs** option.
3. Click **Download**.

An encrypted .tar file containing all the necessary logs will be downloaded.

4. Run the following `ksctl` command to decrypt the .tar file. See [CLI Toolkit](#) to learn how to install and run `ksctl`.

```
ksctl ddc raw-data decrypt [flags]
```

Examples

```
ksctl ddc raw-data decrypt -i /tmp/ScanReport1.tar
ksctl ddc raw-data decrypt -i /tmp/ScanReport1.tar -o /tmp/DecryptedScanReport1.zip
```

Scan logs will be decrypted to a .zip file containing readable files.

Operations

This section provides information on operations that the DDC administrator performs on the CipherTrust Manager.

Reconfiguring DDC agents

In some situations, for example, if the hostname or IP address of the CipherTrust Manager appliance changes, Agents' connection with DDC must be reconfigured with the new hostname or IP address.

Reconfiguring DDC agents on Windows

To reconfigure a DDC Agent:

1. Log on to the host machine as administrator.

2. Open Enterprise Recon Configuration Tool (**er2_config_cmd.exe**).

By default, the tool is available at `C:\Program Files (x86)\Ground Labs\Enterprise Recon 2\`.

3. In the **Master server IP address or host name** field, specify the new hostname or IP address of the CipherTrust Manager.
4. Click **Test Connection**. A message stating "Connectivity test is successful" confirms successful reconfiguration.
5. Click **Finish**.

Reconfiguring DDC agents on Debian

To reconfigure a DDC Agent:

1. Log on to the host machine as a user with root privileges.
2. Reconfigure connection with DDC on the CipherTrust Manager appliance.

```
sudo er2-config -i <hostname|ip_address>
```

Here, `<hostname|ip_address>` represents the new IP address or hostname of the CipherTrust Manager appliance.

3. Restart the Agent service. Configuration settings will be effective after the Agent restarts.

```
sudo /etc/init.d/er2-agent -restart
```

Reconfiguring DDC agents on RHEL

To reconfigure a DDC Agent:

1. Log on to the host machine as a user with root privileges.
2. Reconfigure connection with DDC on the CipherTrust Manager appliance.

```
er2-config -i <hostname|ip_address>
```

Here, `<hostname|ip_address>` represents the new IP address or hostname of the CipherTrust Manager appliance.

3. Restart the Agent service. Configuration settings will be effective after the Agent restarts.

```
sudo /etc/init.d/er2-agent restart
```

Tuning scan settings

You can customize system parameters for all the DDC Agents by using the `ksctl` tool. The following system parameters can be modified this way:

- Agent Memory (in MB)
- CPU used
- Throughput

The `ksctl` command to use to this end is `ksctl ddc settings scan modify`.

Command syntax:

```
ksctl ddc settings scan modify [flags]
```

Flags:

Flag	Usage
<code>--agent-memory int</code>	Setting for the maximum memory usage that the scanner service can use on the agent host, in MB. Default value: 1024.
<code>-h, --help</code>	Help on the command's usage
<code>--jsonfile string</code>	JSON format to create resources in DDC endpoints.
<code>--max-scan-throughput int</code>	Max I/O rate the scanner service will use to read data from the data store, in MBps. Set to 0 for unlimited. Default value is 0.
<code>--scan-cpu string</code>	CPU priority set for the agent used in the scan. The possible values are 'low' and 'normal'. Default value is 'low'.

Global Flags:

Flag	Usage
<code>--configfile string</code>	Full path and name to a file that contains the configuration parameters (optional).
<code>--connection string</code>	The friendly name of the server you want to authenticate against. (default "local_account")
<code>--domain string</code>	The CipherTrust Manager domain that the command will operate in. Can be used only with user/password and not with token. By default the command will operate in the root domain or the domain the user is logged-in.
<code>--jwt string</code>	The JSON Web Token (JWT). It is the access token that can be passed instead of user/password (optional). The command <code>ksctl tokens create</code> creates a JWT.

Flag	Usage
<code>--nossllverify</code>	Do not verify the certificate for SSL/HTTPS authentication (not recommended).
<code>--password string</code>	CipherTrust Manager server user password. Do not use this flag to enter the password (masked) from terminal.
<code>--respfmt string</code>	Response Output format (json is the only supported value at present, optional) (default "json")
<code>--timeout int</code>	Timeout in seconds for TCP connection attempts.
<code>--token string</code>	The refresh token returned from the login command to be passed instead of user/password (optional). The command <code>ksctl login</code> creates a token and writes it to the config file.
<code>--url string</code>	CipherTrust Manager server URL.
<code>--user string</code>	CipherTrust Manager server user name.
<code>-v, --verbose</code>	Provide verbose output while executing command (optional).

Examples:

```
ksctl ddc settings scan modify --agent-memory 1024 --scan-cpu normal --max-sc
an-throughput 2
ksctl ddc settings scan modify --jsonfile scansettings.json
```

Warning

The `ksctl ddc settings scan` command will be deprecated in the coming release.

Restarting DDC agents

Restarting agents on Windows

To restart a DDC Agent, run the following commands:

```
net stop "Enterprise Recon 2 Agent (&lt;ARCH>)"
net start "Enterprise Recon 2 Agent (&lt;ARCH>)"
```

Here, <ARCH> represents the Windows architecture - x32 or x64.

Restarting agents on Debian

To restart a DDC Agent, run:

```
sudo /etc/init.d/er2-agent restart
```

Alternatively, restart the Agent service by stopping it and again starting it manually. Run the following commands:

```
sudo /etc/init.d/er2-agent stop
sudo /etc/init.d/er2-agent start
```

Restarting agents on RHEL

To restart a DDC Agent, run:

```
sudo /etc/init.d/er2-agent restart
```

Alternatively, restart the Agent service by stopping it and again starting it manually. Run the following commands:

```
sudo /etc/init.d/er2-agent stop
sudo /etc/init.d/er2-agent start
```

Renewing DDC ML agent certificate

DDC ML agents are linked with CipherTrust Manager either through client certificates or server certificates for web interface. Certificate renewal is necessary under the following scenarios:

- When client certificate is near expiry
- When signer CA of client certificate is renewed or updated in CipherTrust Manager
- When server certificate is renewed or updated

Note

Only local CAs are supported for issuing certificate for agents. See [Certificate Authority](#) to learn more about local CA certificates.

Renewing client certificate

Client certificate is renewed when the DDC ML agent certificate is about to expire or when its signing CA (Certificate Authority) is modified in the CipherTrust Manager. *The DDC ML agent automatically handles renewal of client certificates.*

- **Agent certificate or its signing CA is near expiry**

When the agent certificate (*with a valid signing CA*) is about to expire, the agent initiates the renewal process sixty days before the expiry. It requests the CipherTrust Manager to issue new certificates. The new certificates are automatically downloaded and saved on the agent host machine.

If the *agent certificate's signing CA* is about to expire, the agent requests CipherTrust Manager to issue a new certificate. This process continues until the expiration or update of the agent certificate's signing CA.

When the signing CA is renewed or updated, CipherTrust Manager sends renewal notifications to the agent, as described in the next section. However, if the signing CA is expired, you must re-install the agent to reestablish communication with CipherTrust Manager.

▪ **Signing CA of agent certificate is updated**

In CipherTrust Manager, each DDC client is associated with a Client Profile. Further, each Client Profile has a CA associated with it, which acts as the signing CA for the agent certificate.

When the signing CA is updated or renewed, CipherTrust Manager sends renewal notification to all the clients associated with the Client Profile.

Upon receiving the notification, DDC ML agents set a flag in their configuration file and within the next 12 hours, the agent certificates are renewed.

When certificate renewal is successful, a message *Updated Agent and signer CA certificates* is added in the log files and the agent continues to function normally.

Note

- If the agent is down or unavailable, CipherTrust Manager will continue to send notifications of CA changes until the agent is accessible again.
- In the event of agent certificate expiring unexpectedly, you need to reinstall the agent to restore communication with CipherTrust Manager.

Renewing server certificate

When a server certificate or one of its signing CAs is renewed or updated, a notification is sent to all the agents registered with the CipherTrust Manager for renewal.

You can renew server certificate either through the CipherTrust Manager API or GUI.

See [Get certificate for NAE, KMIP, and WEB interfaces](#) to issue web interface certificate and use them during renewing.

API

1. Add the new certificate to the list of upcoming sever certificates using the below API.

```
put /v1/configs/interfaces/web/renewal-certificate
```


Parameter	Description
<code>certificate</code>	Specify the certificate chain having the primary certificate, the CA, and its private key.
<code>format</code>	Specify the certificate format.

- Restart the web interface services using the below API.

```
post /v1/system/services/restart
```

Parameter	Description
<code>delay</code>	Specify the delay vale. Default value is 5 seconds.
<code>services</code>	Specify the interface service you want to restart. Enter 'web' as the parameter value.

After upcoming renewal certificate is updated successfully, following events occur at the agent side:

- Active agents receive notification about the update from CipherTrust Manager and updates the trust store with the new certificate chain.

Note

If the agent is down or unavailable, CipherTrust Manager continues to send notifications until the agent is accessible again.

- Old and new certificates are stored on the agent side at `$INSTALL_DIR/Certificates/trustStore.pem` and `$INSTALL_DIR/Certificates/trustStore.0` locations, respectively.
- Agent log displays the *Server CA certificate renewal completed successfully* message.
- Agent trusts both old and new certificates until the former one expires.

- Fetch the upcoming renewal certificate using the below API.

```
get /v1/configs/interfaces/web/renewal-certificate
```

The API returns the public key and CA chain of the upcoming renewal certificate in the PEM format.

- When the current web interface certificate is about to expire, apply the new certificate using the below API.

```
post /v1/configs/interfaces/{interface}/renewal-certificate/apply
```

Note

If the new certificate is not applied manually, it will be automatically applied on expiration of the current certificate.

GUI

1. Upload the upcoming renewal certificate
 - a. Go to **Admin Settings > Interfaces**.
 - b. Click the ellipses icon (...) next to the web interface.
 - c. Select **Renewal Certificate Options**.
 - d. Click **Upload/Generate** to upload a new server certificate.
 - e. Click **OK**.
 - f. Select an upload option.
 - **File Upload:** Use this option to upload the certificate file.
 - **Text:** Use this option to paste the signed certificate and its private key content.
 - g. Select PEM as Format.
 - h. Click **Upload Certificate**.
 - i. Go to **Admin Settings > Services** and click the **Restart** button next to web services.

After the upcoming renewal certificate is uploaded successfully, following events occur at the agent side:

- Active agents receive notification of this update from the CipherTrust Manager and updates the trust store with the new certificate chain.

Note

If the agent is down or unavailable, CipherTrust Manager continues to send notifications until the agent is accessible again.

- Old and new certificates are stored on the agent side at `$INSTALL_DIR/Certificates/trustStore.pem` and `$INSTALL_DIR/Certificates/trustStore.0` locations, respectively.

- Agent log displays the *Server CA certificate renewal completed successfully* message
- Agent trusts both old and new server certificates until the former one expires.

2. Apply the upcoming renewal certificate

When the current interface certificate is about to expire, you need to apply the newly uploaded upcoming renewal certificate for the interface.

- Click the ellipses icon (...) next to the web interface.
- Click **Renewal Certificate Options**.
- Select **Apply** and click **Ok** to apply the upcoming renewal server certificate.

The existing server certificate is replaced with the newly uploaded server certificate.

Note

If the new certificate is not applied manually, it will be automatically applied on expiry of the current certificate.

DDC ML agent configuration parameters

The configuration parameters are available in the `<INSTALL_DIR>/conf/ddc_agent.conf` file.

Performance log settings

Parameters	Default	Description
<code>ddc_agent_extraction_log_threshold</code>	None	This is the elapsed time threshold in milliseconds to log text extraction call. A call will be logged if the elapsed time is equal to or greater than the threshold.
<code>ddc_agent_prediction_log_threshold</code>	None	This is the elapsed time threshold in milliseconds to log ML model prediction call. A call will be logged if the elapsed time is equal to or greater than the threshold.
<code>ddc_agent_archive_log</code>	None	This is the boolean flag on whether to log messages for archive extraction.

Request configuration settings

Parameters	Default	Description
<code>ddc_prediction_worker</code>	(host cpu number) / 4	This is the number of threads to call the ML model prediction interface concurrently.
<code>ddc_prediction_queue</code>	20	This is the maximum number of texts that can be queued for prediction.
<code>ddc_prediction_batch_memory</code>	1 megabytes	ML model has prediction call to accept an array of texts, and this parameter just hints the maximum memory size of the text array in bytes.
<code>ddc_prediction_batch_number</code>	10	ML model has prediction call to accept an array of texts, and this parameter specifies the maximum number of texts in the array.
<code>ddc_parse_worker</code>	(host cpu number) / 4	This is the number of threads to call Apache Tika parser interface concurrently.
<code>ddc_parse_queue</code>	20	This is the maximum number of files can be queued for parsing.
<code>ddc_parser_chunk_size</code>	10 megabytes	When the ML Agent calls Apache Tika to parse a big file, it can receive result texts chunk by chunk and call the ML model prediction interface with those chunks. This parameter specifies the chunk size in bytes.
<code>ddc_prediction_report_threshold</code>	100	This is the frequency for the ML agent to report prediction progress. The default value is 100, which means the ML agent reports progress when every 100 number of files are processed and predicted.
<code>ddc_scan_report_threshold</code>	10000	This is the frequency for the ML agent to report progress of scan file list generation. The default value is 10000, which means the ML agent reports progress when every 10000 files are written into the scan file list.
<code>ddc_java_options</code>	No default	This is the JVM options list separated by comma, for example <code>-Xmx8g, -XX:+UseG1GC</code> .

Model driver

Parameters	Default	Description
<code>ddc_ml_log_level</code>	info	Represent the log level for the model driver. Possible values: info, warn, or error.

TDPaaS retention policy

A retention policy allows you to store data on the cloud TDP (TDPaaS) for a definite amount of time. This feature is useful if there is a need to delete the stored information after a definite amount of time to comply with some regulations and policies.

After the retention policy is enabled, information removal is automatically handled by DDC through scrub jobs. The scrub jobs operate automatically at regular intervals, and you can view the complete details on the **Operations** tab. The scrubbing process clears the scan executions and reports from the cloud storage and CipherTrust Manager database, ensuring they don't show up in the list of executions.

After the scrub job is completed, scrubbed data is no longer accessible for report generation or scan executions, and it can't be restored.

Note

A retention policy is only available when TDPaaS is provisioned. Refer to [TDPaaS Deployment Guide](#) to learn how to provision TDPaaS.

Key features

- **Enable/Disable Flexibility:** Activate or deactivate the Retention Policy at any time.
- **Automatic Scrubbing Jobs:** Scrubbing jobs operate automatically at regular intervals to effectively manage data.
- **Retention Period:** Flexibility to select the retention duration in the range of 3 to 60 months. If the retention period exceeds 60 months, it's recommended to avoid enabling the retention policy.
- **Data Management:** Scrubbing job eliminates scan executions and reports from the cloud storage and the CipherTrust Manager database, ensuring they are no longer visible in the execution list.
- **View Job Details:** Access information about scrub jobs, including:
 - Execution ID - Displays the execution ID of the scrub job.
 - Execution Started - Displays the timestamp when scrub job execution started.
 - Duration - Displays the duration of the scrub job execution.

- **Status** - Displays the current status.
- **Space Freed** - Displays the amount of space freed from cloud storage and CipherTrust Manager databases in MBs.
- **Retention Period** - Displays the retention period in months.
- **Filtering Options**: Filter scrub job data based on the status of job executions and chosen date ranges.

Enabling retention policy

1. Log on to CipherTrust Manager.
2. Go to **Settings > Cloud Management**.
3. Select the **Retention Policy** tab.
4. Turn on the **Retention Policy** toggle.
5. Enter **Retention Period** in the range of 3 to 60 months. By default, it is set to 12 months.
6. Click **Apply**.

Note

If you enable retention policy and switch back to on-premises TDP, then retention policy doesn't disable automatically but remains ineffective.

Viewing Scrub job executions

After enabling the retention policy:

1. Go to the **Operations** tab.

This tab displays all scrub job executions up to the current date. Scrub jobs run automatically at a reasonable frequency.
2. Use the date filter to narrow down job executions to a specific date range.
3. Use the status filter to filter job executions as per Running, Completed, or Failed statuses.

Troubleshooting

Error messages

This section lists the various error messages that the system can display, with explanations and solutions (if

available).

Locations

Error Message	Explanation
"Branch Location name already exists" (message on toast)	You tried to create a location with a name that is already taken by another location. SOLUTION: Choose another name.

Data stores

Error Message	Explanation
"Data Store name already exists" (message on toast)	You tried to create a data store with a name that is already taken by another data store. SOLUTION: Choose another name.
"A valid agent could not be found" (Agent selection - on mouse-over on the data store)	There is no active agent for this data store that the automatic agent selection process has been able to detect. SOLUTION: This requires additional research, such as checking if the agent is installed on the data store, if it has the right type (local/proxy), if it is of the right OS "flavor" (Linux, Windows), or of the right type (e.g. database). Refer to the Data Discovery and Classification Deployment Guide for more information on troubleshooting this issue.
"One of the Oracle Data Stores has been configured with a wrong schema."	SOLUTION: Check the Oracle schema that you used and change it if necessary. For more information, see Configure Connection .
"The target for Data Store SharePoint scan for sites does not have access permissions"	If DDC returns this error while adding a SharePoint Online data store, it is because "grant app permission" is disabled by default on SharePoint Online. SOLUTION: For the SharePoint Add-In to work, the DisableCustomAuthenticationApp setting for the tenants needs to be set to "false". Follow the steps in Configure SharePoint Online .

Scans

Error Message/Issue	Explanation
"Scan name already exists" (message on toast)	You tried to create a scan with a name that is already taken by another scan. SOLUTION: Choose another name.
"All Data Stores are disabled" (message on toast)	You attempted to run a scan that has all data stores disabled. SOLUTION: Enable at least one data store for the scan. Refer to the Scans for instructions.
"The following Data Stores are not accessible: xyz" (message on toast)	You tried to scan a data store that is not accessible. The scan is marked as Failed, and includes a warning

Error Message/Issue	Explanation
	icon with the message "The data store xyz included in the scan is not accessible" on mouse-over. SOLUTION: Verify the connectivity from the agent to the data store. Verify the data store configuration.
"One or more Data Stores are not accessible." (on mouse-over on the scan fail icon)	The scan failed because the data store that is configured is inaccessible. The data store failed after the scan was launched. SOLUTION: There may be a number of reasons for this. To troubleshoot a failed data store refer to Discovering Sensitive Information.
"The following Data Stores have no agent available: xyz" (on mouse-over on the scan fail icon)	You tried to scan a data store that had no agent available when the scan was executed. There is a problem with the agent. The data stores that failed are listed. SOLUTION: This requires additional research, such as checking if the agent is installed on the data store, if it has the right type (local/proxy), if it is of the right OS "flavor" (Linux, Windows), or of the right type (e.g. database). Refer to the Data Discovery and Classification Deployment Guide for more information on troubleshooting this issue.
"Data Store has incorrect credentials" (on mouse-over on the scan fail icon)	Data store credentials provided are incorrect so the scan cannot be executed. These data stores are listed. SOLUTION: Update the server credentials for the data store.
"One or more Data Stores have incorrect credentials" (message on toast)	The credentials for one or more data stores are no longer valid (credentials modified, user deleted, and so on) preventing the scan from completing. SOLUTION: Reconfigure the data store and re-launch the scan.
"The scanner service is not available" (message on toast)	You tried to run a scan with the scan engine unavailable. SOLUTION: Check the status of the scan engine (the CipherTrust Manager server).
"The following Data Stores have missing agents: xyz" (message on toast)	This happens when an agent was assigned to the listed data store(s) and then when a scan was launched, for some reason the assigned agent could not be found on the server. SOLUTION: Try to re-assign the agent in the data stores screen. If this does not work, check the agent assigned to the xyz data store.
"The following Data Stores have agent errors: xyz" (message on toast)	This happens when a management request for an agent fails (for example, at verification or when setting it as a proxy) during the scan execution. SOLUTION: It is usually a transient issue. Wait a few minutes and run the scan again. If it still fails, check the agent status.
"Error processing scan" (message on toast)	This happens when the scan fails in the processing stage, that is when the scan results are being processed by DDC. SOLUTION: Run the scan again. If the error persists,

Error Message/Issue	Explanation
	contact Thales Customer Support .
"Error connecting to HDFS" (message on toast)	This happens when the scan fails because there is no HDFS connectivity. SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-HDFS connection.
"Error connecting to PQS" (on mouse-over on a failed scan status)	A scan failed because there is no PQS connectivity. SOLUTION: Check the HDFS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-PQS connection.
"The Knox logs directory has been filled."	SOLUTION: Change the Knox log level or purge the logs directory. Refer to Changing Knox Log Level in the Thales Data Platform Deployment Guide for details.
"Error checking the data allowance" (message on banner)	This happens when DDC is not licensed. DDC sends a request for data allowance to the license server and the server responds that there is no license. SOLUTION: Obtain and install a valid DDC license. Refer to DDC Licensing for licensing information.
"One target path is missing" (on mouse-over on the scan fail icon)	A scan failed because one or more target paths are missing. SOLUTION: Open the scan for editing, by following the Edit link embedded in the error message, and check which target path is missing (it will be indicated by a yellow exclamation mark in the Targets section).
"One database target has incorrect schema" (on mouse-over on the scan fail icon)	A scan failed because one or more database targets have an incorrect schema. SOLUTION: Open the scan for editing, by following the Edit link embedded in the error message, and check which target has an incorrect schema (it will be indicated by a yellow exclamation mark in the Targets section).
"One database target has incorrect table" (on mouse-over on the scan fail icon)	A scan failed because one or more database targets have an incorrect table. SOLUTION: Open the scan for editing, by following the Edit link embedded in the error message, and check which target has an incorrect table (it will be indicated by a yellow exclamation mark in the Targets section).
"One target has incorrect file extension" (on mouse-over on the scan fail icon)	A scan failed because one or more targets have an incorrect file extension. SOLUTION: Open the scan for editing, by following the Edit link embedded in the error message, and check which target has an incorrect file extension (it will be indicated by a yellow exclamation mark in the Targets section).
"One target has nested paths" (on mouse-over on	A scan failed because one or more targets have

Error Message/Issue	Explanation
the scan fail icon)	nested paths. SOLUTION: Open the scan for editing, by following the Edit link embedded in the error message, and check which target has a nested path (it will be indicated by a yellow exclamation mark in the Targets section).
"The target for Data Store is a file" (on mouse-over on the scan fail icon)	A file cannot be a target of a data store. SOLUTION: Open the scan for editing, by following the Edit link embedded in the error message, and put a directory as the target path (the failing target path will be indicated by a yellow exclamation mark in the Targets section).
"The target for Data Store is not a valid directory" (on mouse-over on the scan fail icon)	The specified directory used as the target path is invalid. SOLUTION: Open the scan for editing, by following the Edit link embedded in the error message, and check the directory. The invalid directory will be indicated by a yellow exclamation mark in the Targets section.
"The target for Data Store cannot be accessed" (on mouse-over on the scan fail icon)	The specified path used as the target path in inaccessible. SOLUTION: Open the scan for editing, by following the Edit link embedded in the error message, and check the path. The inaccessible path will be indicated by a yellow exclamation mark in the Targets section.
"Data Allowance is exhausted" (on mouse-over on the scan status)	A scan was run that exceeded the Data Allowance. All scans running when the Data Allowance gets exhausted (or after that) will fail. From that moment on, you cannot launch new scans. SOLUTION: Contact Thales Customer Support and upgrade your DDC license. Refer to DDC Licensing for licensing information.
"Too many sensitive Data Objects found" (on mouse-over on the scan fail icon)	Scan results have a limit on the amount of information (Sensitive Data Object found, infotypes and matches found on each Sensitive Data Object, etc.) and the scan results exceeded this limit. MongoDB data stores are particularly prone to this error. SOLUTION: Split the original scan into multiple smaller scans, re-launch the scans and generate a single report combining the results from all the scans. To reduce the resulting scan size consider scheduling a different scan per Data Store and/or per Classification Profile and/or subpaths (such as folders and tables) in the original scan path.
"Error validating Target Paths" (on mouse-over on the scan fail icon)	The target paths validation failed for some reason. SOLUTION: Go to the scan view and verify that the target paths exist, and that the guard points exist and are enabled.
"There was a fatal error in scan service" (on mouse-over on the scan fail icon)	Unknown error condition on the scan service. Mainly seen when a Database server becomes inaccessible for the scan agent halfway through the scan due to

Error Message/Issue	Explanation
	network issues or DB server crash.
"Scan RAM exhausted" (on mouse-over on the scan fail icon)	A G-mail scan with a large amount of sensitive data fails with this error. SOLUTION: Increase the Agent's RAM using the ksctl tool. For details refer to Tuning Scan Settings .
"Scan results could not be found" (on mouse-over on the scan fail icon)	The scan failed because two independent sub-scans completed in the same second. DDC schedules sub-scans for every path added to the scan. If multiple agents scan a single path, each of them will perform an independent sub-scan. SOLUTION: 1) If the scan failed because of multiple scans finishing in the same second - re-launch the failing scan. 2) If the scan includes multiple agents - reduce the number of agents assigned to Data Stores in the scan. 3) If the scan includes multiple scan paths for a single Data Store - reduce the number of scan paths by scheduling independent scans or by scanning a parent folder that includes multiple scan paths.
"Status of one or more Data Stores is failed, go to Data Store list to resolve the issue" (on mouse-over on the failed scan icon)	The agent assigned to one of the data stores used in the scan has failed or could not be found and no alternative agent was found. SOLUTION: Follow the "Data Store list" link embedded in the error message, locate and fix the agent that has failed. Then, go back to the scans page and relaunch the scan.
Scan run fails after few minutes with error: "Scan result could not be found" Scan run gets stalled for hours or days	Agent is disconnected. SOLUTION: Identify the agent running the scan and try reconnecting the agent. API request quota limit for the data store is exhausted. SOLUTION: Verify quota usage from cloud provider logs. See API Request Quota Limit to check quota limit for different DS.
data store path not accessible	This error indicates an issue with Hadoop file access permissions. It occurs when scanning a Hadoop data store and the configured scan target is a Hadoop file for which you don't have access rights.
table or schema not accessible	This error occurs when scanning an IBM, Oracle, or MS-SQL data store that contains an empty table.
Scanning a Gmail label did not find any results	You can scan emails in a Gmail label if you move the emails to the label, otherwise, the emails will be kept in inbox folder. For the default system labels, Gmail creates some folders that do not match the label name. Solution: Refer to the Gmail documentation to learn the right path to scan a particular system label.
Exchange Online scan completes faster than expected without any matches	A full data store scan is executed successfully even with an incorrect domain name, without triggering any errors or matches. Solution: Verify that the provided domain name is

Error Message/Issue	Explanation
	valid.

Reports

Error Message	Explanation
"Report name already exists" (message on toast)	You tried to create a report with a name that is already used in another report. SOLUTION: Choose another name.
"The version of the scan that was used to generate the report can no longer be found." (message on toast)	The report execution information in TDP references some resource version that is no longer available inside the DDC Database, such as Data Store, Custom Infotype, Classification Profile, or other, probably after restoring DDC from a backup that was taken before the referenced resource was created or modified. SOLUTION: Please restore CipherTrust Manager from a newer backup that includes the referenced resource version, or update the report template to stop referencing the missing resource.
"The report template is configured with a scan execution that can no longer be found." (message on toast)	The report template references a concrete scan execution that is no longer found in TDP, probably after restoring TDP from a backup that was taken before the referenced scan execution was completed. SOLUTION: Please restore TDP from a newer backup that includes the information of the referenced scan execution, or edit the report template to reference a valid scan execution date.
Empty report created for large ML scan runs.	Empty reports may be created for these reasons: - The data file contains a very high number of sensitive entity objects, usually more than 4 million. - There may be constraints with cloud resources, like limited bandwidth between the Machine Learning as a Service (MLaaS) and the agents, or instability issues with Google Cloud Platform (GCP). Workaround: Check the agent log to identify the flagged data object that has high number of scanned entities. To fix the issue, divide the files to ensure match entities remain under the 4 million limit.

Classification Profiles

Error Message	Explanation
"Classification Profile name already exists" (message on toast)	You tried to create a classification profile with a name that is already taken by another classification profile. SOLUTION: Choose another name.

Licensing

Error Message	Explanation
"DDC License not found - try again in a few minutes if you recently inserted one" (message on banner)	Any action performed in the UI results in this message, because there is no valid DDC license installed. SOLUTION: Obtain and install a valid license. Refer to DDC Licensing for licensing information.
"DDC License expired" (message on banner)	Any action performed in the UI results in this message, because your DDC license has expired. SOLUTION: Obtain and install a valid license. Refer to DDC Licensing for licensing information.

DDC ML agent

This section lists various error messages related to the DDC ML agent with explanations and solutions (if available).

Error Message/Issue	Explanation
none Notice: Connection Info - new 0.0.0.0.0: Data connection failed: Connection refused	Possible Causes: ▪ CipherTrust Manager IP/hostname is not correct. ▪ There is an issue with the DDC license. Solution: ▪ Provide a correct CipherTrust Manager IP/hostname. ▪ Ensure DDC license is present on CipherTrust Manager and is in active state.
DDC ML unified agent installation fails without generating logs.	Log directory was not set up during agent installation. Solution: Ensure that the directory <code>C:\ProgramData\thales\ddc_agent</code> exists.
DDC ML unified agent registration fails.	Check error details in <code>MLAgentReg.log</code> at <code>C:\ProgramData\thales\ddc_agent</code>.
Client with name " already exists."	The agent is already registered with CipherTrust Manager. Solution: Re-register the agent using a new hostname/IP or existing hostname/IP. To use existing hostname/IP, you must perform clean up on CipherTrust Manager. See Re-register DDC ML Agent to learn how to remove the existing agent from CipherTrust Manager.
No registered agents available	Check the <code>ML_Reg.log</code> file of the agent that is expected to be selected to identify the issue details.

Error Message/Issue	Explanation
Cannot find minimum agents for the datastore	For SMB scan, there are no agents available to run the scan. Solution: Verify that the Windows agent is registered with CipherTrust Manager. If there is an issue with registration, then re-register the agent, and refresh the datastore from CipherTrust Manager GUI.
Agent status is "Not Connected"	Agent's service is down. Check the agent's service.log file for error details.
Internal error	Displayed when a scan fails. Check the agent and ddc_agent_spark.log files for error details.

TDP (On-prem)

Error Message/Issue	Explanation
"Hadoop is not active. Please go to DDC Settings --> Hadoop" (message on toast)	Problem communicating with Hadoop or DDC has not been configured with Hadoop. SOLUTION: Assuming that you have Hadoop deployed in your environment, configure DDC to use it (DDC Settings --> Hadoop in the CipherTrust UI). For a detailed procedure, refer to the Data Discovery and Classification Deployment Guide .
"Error connecting to the PQS database" (message on toast)	Problem communicating with the Phoenix Query Server database (i.e. HBase). SOLUTION: Check the PQS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-PQS connection.
"Error creating the PQS database schema" (message on toast)	Problem communicating with the Phoenix Query Server database. SOLUTION: Check the PQS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-PQS connection.
"Error using the PQS database schema" (message on toast)	Problem communicating with the Phoenix Query Server database. SOLUTION: Check the PQS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-PQS connection.
"Error connecting to HDFS" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-HDFS connection.
"Invalid HDFS directory path: Not a directory" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-HDFS connection.

Error Message/Issue	Explanation
	Classification Deployment Guide for information on configuring the DDC-HDFS connection.
"Incorrect credentials in the HDFS connection" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-HDFS connection. Check that the authentication service is up and running.
"Incorrect HDFS URI" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-HDFS connection.
"Invalid HDFS folder: the path to the folder does not exist" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-HDFS connection.
"Invalid server certificate in the HDFS request" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-HDFS connection.
"PQS incorrect credentials or authentication service not available" (message on toast)	SOLUTION: Check the PQS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-PQS connection. Check that the authentication service is up and running.
"Invalid server certificate in the PQS request" (message on toast)	SOLUTION: Check the PQS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the Data Discovery and Classification Deployment Guide for information on configuring the DDC-PQS connection.
"Your system does not meet the 16GB RAM minimum" (message displayed across the top of all DDC screens)	DDC requires at least 16GB of RAM to be able to run properly. SOLUTION: Increase the RAM memory on board to at least the required minimum of 16GB.
Knox connection is not configured (message on toast)	Displayed when the user tries to run a scan without setting the Hadoop Knox connection of Hadoop Service. Solution: Configure the Hadoop Knox connection of Hadoop Services under the Access Management > Connection Management tab of CipherTrust Manager.
Invalid Knox connection	Displayed when the user tries to run a scan with an invalid Hadoop Knox connection of Hadoop Services. Solution: Verify and reconfigure the Hadoop Knox connection of Hadoop Services under the Access Management > Connection Management tab of

Error Message/Issue	Explanation
	CipherTrust Manager.
HDFS settings don't exist	Displayed when the user tries to run a scan without configuring the HDFS settings of Hadoop Services. Solution: Configure the HDFS settings under Settings > Hadoop Services in DDC.
Livy settings don't exist	Displayed when the user tries to run a scan without configuring the Livy settings of Hadoop Services. Solution: Configure the Livy settings under Settings > Hadoop Services Settings in DDC.
--Pending-- (message on toast)	Displayed when a user tries to save the HDFS settings but the hostname or port from the connection created in Connections Management are wrong or not reachable.
Livy connection refused (message on toast)	Displayed when a user tries to save the Livy settings but the hostname or port from the connection created in Connections Management are wrong or not reachable.
Invalid Knox server certificate (message on toast)	Displayed when a user tries to save the HDFS or Livy settings but the certificate is invalid.
Invalid HDFS URI path (message on toast)	Wrong parameter added in the DDC Hadoop settings.
Invalid HDFS folder (message on toast)	Wrong parameter added in the DDC Hadoop settings.
Invalid Livy URI path (message on toast)	Wrong parameter added in the DDC Hadoop settings.
503 Service Unavailable (message when configuring HDFS/Livy)	If you are using the CipherTrust Manager's proxy feature, make sure it is properly configured. It should be possible to directly access TDP, to resolve its name, and to allow HTTPS connections to port 8443. Check your proxy software manual to get those rules configured.
Connection error "[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:618)." (message in ambari-agent.log) Server lost heartbeat from all nodes (message on Ambari UI)	Ambari CA certificate is corrupted or has expired. Solution: See Expired/Corrupted Ambari CA Certificates to resolve this issue.
TDP silent installation failure	Occasionally, TDP silent installation may fail when setting up node cluster. Solution: See TDP Silent Installation Failed to resolve this issue.

TDPaaS

The below sections list various error messages that you may encounter when working with TDPaaS, with their explanations and possible solutions.

Scans & Reports

Error Message	Explanation
No valid Hadoop Knox or Data Management Service connection exist	Displayed when the user tries to run a scan with an invalid Hadoop Knox connection of Hadoop Services. Solution: Verify and reconfigure the Hadoop Knox connection of Hadoop Services under the Access Management > Connection Management tab of CipherTrust Manager.
None of Hadoop Knox & Data Management Service connection exist	Displayed when the user tries to run a scan without configuring either Hadoop Knox or Data Management Service connection. Solution: Configure the Hadoop Knox connection of Hadoop Services under the Access Management > Connection Management tab of CipherTrust Manager or Provision Data Management Service settings under Settings > Cloud Management > Data Management tab in DDC.
The Data Management Service settings don't exist	Displayed when the user tries to run a scan without configuring the Data Management Service settings. Solution: Configure Data Management Service under Settings > Cloud Management > Data Management tab in DDC.
None of Data Management Service & Hadoop Services settings exist	Displayed when the user tries to run a scan without configuring the HDFS & Livy settings of Hadoop Services and Data Management Service. Solution: Configure the HDFS & Livy settings under Settings > Hadoop Services in DDC or provision Data Management Service under Settings > Cloud Management > Data Management tab in DDC.
Connection is Lost: Data Management Service connection is not responding.	Displays when the connection to the Data Management Service is interrupted. Soluton: Ensure Data Management Service are correctly configured in the Settings tab.

Data Management Service Settings & Provisioning

Error Message	Explanation
Error: URL param is invalid	Invalid URL value was mentioned inside OIDC Connection Details of Data Management Service settings used for provisioning. Solution: Try provisioning again after updating the URL field with a valid value.
Error: Data Management Service was previously provisioned with a region. It cannot be changed.	The user tried to reconfigure Data Management Service settings with a different region.

Error Message	Explanation
	Solution: This action is not allowed. Data Management Service settings are already provisioned with a region. After successful provisioning, region cannot be modified.
Error: Hdfs Settings can't be changed	Data platform was switched from Data Management Service to Hadoop Services after a successful scan execution. Solution: This action is not allowed. After executing a scan with Data Management Service, you can't switch to Hadoop Services.
Error: Livy Settings can't be changed	Data platform was switched from Data Management Service to Hadoop Services after a successful scan execution. Solution: This action is not allowed. After executing a scan with Data Management Service, you can't switch to Hadoop Services.
Error: Tdpaas Settings can't be changed	Data platform was switched from Hadoop Services to Data Management Service after a successful scan execution. Solution: This action is not allowed. After executing a scan with Hadoop Services, you can't switch to Data Management Service.
TDPaaS Provision Data can't have empty fields	Data Management Service provisioning returned empty fields in response. Solution: Retry Data Management Service provisioning.
Error: region param is invalid	Data Management Service was provisioned with an invalid or unsupported region. Solution: Try provisioning with another supported region.
Error: missing field %s	Data Management Service configuration was attempted with a missing value for a mandatory field. Solution: Specify the missing field value and reconfigure the service.
Error: Data Management Service Settings could not get provisioned	Data Management Service provisioning failed to create and fetch the Data Management Service settings. Solution: Retry Data Management Service provisioning.
Error: Operation forbidden, cannot change settings - scan is running	Switched the data platform from Hadoop Services to Data Management Service or from Data Management Service to Hadoop Services while scan execution was in progress.

Error Message	Explanation
	Solution: Switch the data platform after scan execution gets failed. After a scan is successfully completed, you can't switch the data platform type.
Error: Operation forbidden, cannot change settings - TDPaaS settings is fixed	Switched the data platform from Hadoop Services to Data Management Service after a successful scan execution. Solution: This action is not allowed. After a successful scan execution with Data Management Service, you can't switch the data platform.
Error: Operation forbidden, cannot change settings - TDP-OnPrem settings is fixed	Switched the data platform from Data Management Service to Hadoop Services after a successful scan execution. Solution: This action is not allowed. After a successful scan execution with Hadoop Services, you can't switch to Data Management Service.

Other errors

Error Message	Explanation
DDCTDPaaSDataProcLaunchJobOperation	Error relates to starting the scan job while using Data Management Service. Solution: This is an unexpected error. Details about the specific cause of error are logged in the Loki Audit records, for each failed attempt of starting the scan job. Please contact customer support to resolve this issue.
DDCTDPaaSDataProcGetJobStatusOperation	Error in performing the get scan job status while using Data Management Service. Solution: This is an unexpected error. Details about the specific cause of error are logged in the Loki Audit records, for each failed attempt of getting the scan job. Please contact customer support to resolve this issue.
Error: Both LivySettings & Data Management Service are set	Livy Settings of the Hadoop Services and Data Management Service settings are set concurrently. Solution: This is an unexpected error. Please contact customer support to resolve this issue.
Error: Neither Livy nor Data Management Service is set	Neither Livy Settings of Hadoop Services nor Data Management Service settings are set. Solution: Ensure Hadoop Services or Data Management Service is correctly configured on the DDC Settings tab.
Error: cannot initialize StorageConnector, Hdfs & Data Management Service are set	Both HDFS Settings of the Hadoop Services and Data Management Service settings are set concurrently.

Error Message	Explanation
	Solution: This is an unexpected error. Please contact customer support to resolve this issue.
Error: Neither HDFS nor Data Management Service is set	Neither HDFS Settings of Hadoop Services nor Data Management Service settings are set. Solution: Ensure Hadoop Services or Data Management Service is correctly configured on the DDC Settings tab.

Info log messages

This section lists the various **information** messages that Data Discovery and Classification sends to CipherTrust Manager log.

OLEANDER INFO

Error Message	Explanation
"[Background-Processes] This node was NOT selected as active node. Turning off background processes"	Related to clustering, self-explanatory error.
"[Background-Processes] This node was selected as active node. Turning on background processes"	Related to clustering, self-explanatory error.
"[Background-Processes] Recovering collectors"	Oleander has received a valid license and recovers the collector processes that were stopped.
"[Background-Processes] Updating license from DMV"	Oleander is requesting from DMV the licenses available for DDC.
"[Background-Processes] Global license status set to nil"	Oleander is unlicensed.
"[Background-Processes] Global license status set to ", "newLicenseStatus", *newLicenseStatus"	Oleander license status is whatever <i>newLicenseStatus</i> is.
"[Background-Processes] Checking HDFS connectivity"	Oleander is sending a ping to HDFS to check the connectivity.
"[Background-Processes] Cannot connect with HDFS", "error", err	Oleander's ping against HDFS has failed. Oleander has no connectivity with HDFS.
"[Background-Processes] HDFS connectivity successful"	Oleander has successfully performed a ping against HDFS. Oleander has connectivity with HDFS.
"[Background-Processes] Checking PQS connectivity"	Oleander is sending a ping to PQS to check the connectivity.
"[Background-Processes] Cannot connect with PQS", "error", err	Oleander's ping against PQS has failed. Oleander has no connectivity with PQS.
"[Background-Processes] PQS connectivity successful"	Oleander has successfully performed a ping against PQS. Oleander has connectivity with PQS.
"[Background-Processes] License status has changed"	Oleander license status has changed since the last license cron execution.

Error Message	Explanation
"[Background-Processes] Initializing fast cron: Oleander unlicensed", "UnlicensedCronFrequency", o.Config.UnlicensedCronFrequency	Oleander is unlicensed so the license cron (which asks DMV for a DDC license) increments its frequency. The frequency is defined in the docker-compose file.
"[Background-Processes] Initializing fast cron: waiting for Hadoop connectivity", "HadoopConnectivityCronFrequency", o.Config.HadoopConnectivityCronFrequency	Oleander has a valid license but does not have connectivity with Hadoop.
"[Background-Processes] Initializing slow cron: Oleander licensed and Hadoop connectivity successful", "RunningCronFrequency", o.Config.RunningCronFrequency	Oleander has a valid license and has connectivity with Hadoop, the license cron decrements its frequency. The frequency is defined in the docker-compose file.
"[Background-Processes] About to run NO LICENSE scenario"	Oleander has no valid license so it will stop all running scan watchers, automatic agent selections and scan schedules.
"[Background-Processes] About to run VALID LICENSE scenario"	Oleander didn't have a valid license but now it has one so all scan watchers, automatic agent selections and scan schedules that were stopped are being resumed.
"[Background-Processes] Recovering automatic agent selection"	Part of the VALID LICENSE scenario above.
"[Background-Processes] Starting automatic agent selection for pending datastores", "pending datastores", strings.Join(ds, ", ")	Part of the VALID LICENSE scenario above.
"[Background-Processes] Recovering watchers"	Part of the VALID LICENSE scenario above.
"[Background-Processes] Starting watchers for ongoing scans", "ongoing scans", strings.Join(ss, ", ")	Part of the VALID LICENSE scenario above.
"[Background-Processes] Recovering scan schedules"	Part of the VALID LICENSE scenario above.
"[Background-Processes] Migrating PQS database"	Self-explanatory message.
"[Background-Processes] Deleting background process for scan", "name", sc.Scan.Name	Oleander has no valid license so all scans are being stopped as well as their corresponding background processes.
"[Background-Processes] Updating status to FAILED for scan", "name", sc.Scan.Name	Oleander has no valid license so all scans are being stopped (with status FAILED).
"Cannot connect to HDFS", "settings", hdfsSettings, "error", err	Oleander's ping against HDFS has failed. Oleander has no connectivity with HDFS.
"Cannot connect to PQS", "Settings", pqsSettings, "error", err	Oleander's ping against PQS has failed. Oleander has no connectivity with PQS.
"[Datastores] Agent selected: ", "Name", a.Name	Oleander has found and assigned a suitable agent for the mentioned data store.
"Unable to connect with datastore", "datastore name", d.Name, "error", err	A probe against the mentioned data store has failed during a test connectivity check.
"Instantiating new scrim helper"	Oleander is instantiating a new ScrimHelper object which is used for communication with Scrim, Minerva, Sallyport and DMV.
"Instantiating new hdfs scan collector"	Oleander is instantiating a new HDFSCollector object

Error Message	Explanation
	which is used for communication and processing with HDFS.
"Instantiating scheduler cron"	Oleander is instantiating a new SchedulerCron and starting the background processes.
"[WARNING] PQS connector can not be closed in GetSummaryReport service"	Oleander was unable to close the PQS connector while executing a GetSummaryReport operation.
"[WARNING] PQS connector can not be closed in GetDatastoresDetailsReport service"	Oleander was unable to close the PQS connector while executing a GetDatastoresDetailsReport operation.
"[WARNING] PQS connector can not be closed in GetReportTemplate service"	Oleander was unable to close the PQS connector while executing a GetReportTemplate operation.
"[Scan-Actions] [WARNING] PQS connector can not be closed in GetScanExecutions service"	Oleander was unable to close the PQS connector while executing a GetScanExecutions operation.
"[Scan-Watchers] Stop watcher signal received", "scan", sc.Scan.Name	A running scan watcher has received a STOP signal and the current execution is cancelled.
"[Scan-Watchers] Watcher has failed stopping scan from scanned service"	The scan has been in an INTERRUPTED status for too long so Oleander has tried to stop the scan in ER2 but the request was unsuccessful.
"[Scan-Watchers] Watcher has detected different status", "scan", sc.Scan.Name, "current status", sc.Scan.ScanProcess.Status, "scan service status", er2MappedStatus	The scan watcher has detected a change in the scan status, scan status will be updated in DDC.
"[Scan-Watchers] Interrupted status received", "MaxTimeInterruptedState", o.Config.Er2MaxInterruptedTime, "scan", sc.Scan.Name, "current status", sc.Scan.ScanProcess.Status, "scan service Mapped status", er2MappedStatus, "scan service status", er2Status, "InterruptedTimestamp", sc.InterruptedTimestamp	The scan watcher has received an INTERRUPTED status for the scan, the scan watcher will continue asking until the status changes or the timeout is exceeded.
"[Scan-Watchers] Scan has finished... collector starting", "scan", sc.Scan.Name	The scan watcher has received a COMPLETED status for the scan, the scan status is set to PROCESSING in DDC and the scan collector starts.
"DDCScanActionComplete"	Scan audit record, logged after a successful scan execution.
"DDCScanActionFail"	Scan audit record, logged after a failed scan execution.
"AggregatedReportOperationCompleteDDC"	Report audit record, logged after a successful aggregated report generation.
"ScanTrendReportOperationCompleteDDC"	Report audit record, logged after a successful scan trend report generation.

SUNDEW INFO

Error Message	Explanation
"Asking for license status"	Sundew is requesting from Oleander the DDC license status.
"Active license information received", "http ret code", c	Sundew has received a valid license status from Oleander.
"Starting scan service..."	Sundew is launching ER2.
"Connectivity test successful, scan service is up"	Sundew confirms that ER2 is up-and-running.
"Generating scan service license...", "ID", s.er2Status.lastIssuedLicenseID	Sundew is generating a license for ER2.
"Injecting license into scan service"	Sundew is injecting the generated license into ER2.
"License won't be generated/refreshed in this iteration", "ID", s.er2Status.lastIssuedLicenseID, "checks (intervals)", s.Config.GeneratingLicenseInterval-s.er2Status.lastIssuedLicense	No action is required from Sundew for this iteration regarding licensing.
"Not active license or product not licensed response received", "http ret code", c	Sundew has received an invalid license status from Oleander.
"Scan service stopped"	Sundew is stopping ER2.
"Unexpected error code in response for /license/status", "http ret code", c	Sundew asked Oleander for the license status but got an unexpected response.

Error log messages

This section lists the various **error** messages that Data Discovery and Classification sends to CipherTrust Manager log.

OLEANDER ERROR

Error Message	Comment/Explanation
Error connecting to the scan service	This error means a connectivity issue between the oleander and the sundew/ER2.
DDC Error creating the database schema.: error executing http request. Code: 500	To check the connectivity with Hadoop external data base a schema is created. So this means that there is no connectivity with Hadoop. (also, possibly related to "Error creating the PQS database schema" in the UI)
"CLIENT_CREDENTIAL_PARTITION is not set"	CLIENT_CREDENTIAL_PARTITION variable is not set in the config object.
"[Background-Processes] Error retrieving license from DMV", "error", err	Oleander GetLicenses request against DMV has failed. This error could have been caused by DMV being down.
"[Background-Processes] Error killing all agents selections"	When the oleander instance loses its license or the current license expires, all ongoing agent selections

Error Message	Comment/Explanation
	will stop. This error is caused by an internal Oleander issue while this agent selections are being shut down.
"[Background-Processes] Error killing all scan watchers"	When the oleander instance loses its license or the current license expires, all ongoing scan tracking will stop. This error is caused by an internal Oleander issue during this scan tracking shut down.
"[Background-Processes] Error removing all scan schedules due to DDC not licensed"	When the oleander instance loses its license or the current license expires, all scheduled scans will be stopped. This error is caused by an internal Oleander issue during this scheduled scans stopping.
"[Background-Processes] Error starting agent selection for datastores"	The listed Data Stores have no agent available.
"[Background-Processes] Error starting scans watchers"	When the oleander instance receives a valid license from DMV, all stopped agent selections must be resumed. This error is caused by an internal Oleander issue during this agent selections resuming.
"[Background-Processes] Error starting cron schedules"	When the oleander instance receives a valid license from DMV, all stopped scheduled scans must be resumed. This error is caused by an internal Oleander issue during this scheduled scans resuming.
"[Background-Processes] Error trying to migrate the PQS database"	During the PQS configuration, an error occurred trying to apply the changesets to update the Database, create the tables, etc...
"[Background-Processes] Error creating cron", "error", err	Background_processes service constantly creates crons for license status checking against DMV, this error is caused by an internal Oleander issue during the creation of one of this crons.
"[Background-Processes] Error with unmarshal."	Any "unmarshal/unmarshalling" error is caused by an internal Oleander issue converting a golang object to JSON format or vice versa.
"[Background-Processes] Error while trying to update status to FAILED for datastore.", "name", ds.Name, "error", err	When background_process services gets an invalid or expired license from DMV all running scans must be stopped and set as FAILED, this message indicates an internal Oleander error changing the scan status for some scan.
"[Background-Processes] Error while trying to retrieve scans from background processes table"	When the oleander instance receives a valid license from DMV, all stopped normal scans must be resumed. This error is caused by an internal Oleander issue accessing the background processes table, which contains all the information for the scans resuming.
"[Background-Processes] Watcher has failed updating scan status for scan", "name", sc.Scan.Name, "error", err	background_processes service has the responsibility to track the running scans and update their Oleander status, this error message indicates an internal Oleander issue updating the scan_process table.
"Cannot retrieve the HDFS settings", "error", err	Connectivity or internal Oleander error trying to retrieve HDFS settings from Citrus.
"Cannot retrieve the PQS settings", "error", err	Connectivity or internal Oleander error trying to

Error Message	Comment/Explanation
	retrieve PQS settings from Citrus.
"Cannot find the country", "error", err	When a user creates a location and he sets a country that is not registered in our DB.
"Cannot find the state", "error", err	When a user creates a location and he sets a state ID that is not registered in our DB.
"Trying to verify if the country has states", "error", err	When a user creates a location and he sets a state "Name" that is not registered in our DB.
"Missing tag %s for the default classification profiles", r	When inserting default classification profiles, there is no tag that matches the correct Regulation
"[Datastores] Error encrypting connection for datastore: ", "Name", d.Name	Error when calling Scrim Helper for the encryption of the Connection field for a data store.
"[Datastores] Error while trying to create background process resource for datastore: ", "Name", d.Name	When a datastore is created, a new row is inserted into the Background Processes table for further tracking, this error message indicates an internal Oleander issue inserting that row.
"[Datastores] Error selecting agent for datastore ", "Name", dsAAS.Name, "error", err	No suitable agent has been found for this data store.
"[Datastores] Error while trying to retrieve datastore for background process, agent selection might fail: ", "error", err	Error retrieving a data store from DB for background processes purposes in a data store.update operation.
"[Datastores] Error running automatic agent selection", "error", errAgentUpdate	An internal Oleander error has occurred while trying to update the status of a data store during the automatic agent selection.
"[Datastores] Error while trying to unmarshal datastore from background processes"	An internal Oleander error has occurred while trying to translate a JSON object to a golang object while recovering the automatic agent selections.
"Error closing the json file", "error", err	Oleander failed trying to close a JSON file.
"Error trying to close the families json file"	Oleander reads a families JSON file for tartup DB population. This error message indicates an internal Oleander error while closing this file.
"Error initializing the account", "error", *errPtr	Oleander failed trying to set the initialization status to the accounts map.
"Error closing oleander", "error", err	Oleander service could not be closed.
"Error starting background processes", "error", err	Some error has occurred trying to execute all background processes.
"Error connecting to HDFS", "error", err	This happens when the scan fails because there is no HDFS connectivity.
"Error trying to close the info types json file"	Oleander reads an infotypes JSON file for startup DB population. This error message indicates an internal Oleander error while closing this file.
"Error connecting to PQS", "error", err	This happens when the scan fails because there is no PQS connectivity.
"Error closing the temporary file", "FileName",	Oleander failed closing the temporary file used for

Error Message	Comment/Explanation
f.Name(), "error", err	decrypting the raw data file.
"Error deleting the temporary file", "FileName", f.Name(), "error", err	Oleander failed deleting the temporary file used for decrypting the raw data file.
"Error trying to close the zip file", "error", merry.Wrap(err).WithHTTPCode(http.StatusBadRequest)	Oleander failed closing the zip file used for decrypting the raw data file.
"Error changing the scan status", "error", err	Internal Oleander error while trying to update the scan status in the Scan Process table.
"[Scan-Launcher] Error while trying to create background process resource for datastore", "name", rsc.Name	Oleander inserts a row in the Background Processes table for further scan tracking - this message indicates an internal Oleander error while performing this insert.
"[Scan-Launcher] The scan watcher returned an error", "error", err	Generic error message for any issue during the scan watcher process.
"[Scan-Launcher] Agent not found for datastore", "DS name", s.ScanDatastores[i].Datastore.Name, "error", err	The listed Data Stores have no agent available.
"[Scan-Launcher] Error getting absolute paths", "DS name", s.ScanDatastores[i].Datastore.Name, "error", err	Oleander was unable to retrieve the absolute paths of the mentioned data store for further scan execution.
"[Scan-Launcher] Error getting connection path", "DS name", s.ScanDatastores[i].Datastore.Name, "error", err	Oleander was unable to retrieve the connection paths of the mentioned data store for further scan execution.
"[Scan-Launcher] Error retrieving the oleander context"	Error while generating the Oleander service user. (this is the context for executing actions on behalf of the itself, instead of a specific user).
"[Scan-Launcher] Error while validating the scan"	This is a generic error, one of the steps of the validation has failed. (could be that some data stores are not ready, a probe has failed, etc)
"[Scan-Launcher] Error while retrieving extensions"	The extensions necessary for the scan execution could not be retrieved from the DB.
"[Scan-Launcher] Unable to initialize background process"	The background process object could not be initialized so the scan watcher cannot start.
"[Scan-Actions] Error found while trying to delete background process resource"	When Oleander has finished tracking a scan it removes the corresponding row from background processes table - this error message indicates an internal Oleander issue removing that row.
"[Scan-Actions] Error trying to execute the Scan scheduled run", "error", err	Oleander failed trying to add a scan schedule for a scan.
"Error changing the scan status", "error", err	Generic error message for an issue during the update of the scan status.
"[Scan-Watchers] Watcher has failed getting scan status from scanned service", "scan", sc.Scan.Name, "error", err	This indicates an underlying connectivity issue with Sundew.
"[Scan-Watchers] Watcher has failed updating scan"	Error updating the scan status in the scan process

Error Message	Comment/Explanation
status", "scan", sc.Scan.Name, "error", err	table.
"[Scan-Watchers] Error processing the report: ", "error", err	Oleander scan collector has failed trying to create the scan collector background process for the mentioned scan.
"[Scan-Watchers] Scan aborted, maximum wait for scan results exceeded", "MaxTimeInterruptedState(minutes)", o.Config.Er2MaxInterruptedTime/60, "scan", sc.Scan.Name, "current status", sc.Scan.ScanProcess.Status, "scan service status", er2MappedStatus	The scan has been in INTERRUPTED er2 state for too long, when it exceeds the timeout the status is changed to FAILED in DDC.
"Error trying to parse er2 polling frequency from env"	Oleander failed trying to read the ER2 POLLING FREQUENCY env variable from the config object.
"Error while trying to retrieve scans from background processes table", "error", err	Oleander tried to retrieve all SCAN rows in the background processes table but it failed.
"Error trying to recover the scan watchers", "error", err	One of the scan watchers instantiated by the recovery system has returned an error and stopped working.
"Wrong datastore credentials"	Oleander was unable to reach a data store due to credentials failure.
"Wrong target path defined"	A scan has failed because the target path defined was not valid.
"Wrong db schema in target path"	A scan has failed because the db schema defined was not valid.
"Wrong db table in target path"	A scan has failed because the DB defined was not valid.
"Wrong defined file extension in target path"	A scan has failed because a file was specified without an extension.
"Error connecting to the scan service"	Oleander is unable to connect with ER2.
"Error processing the scan reports"	Generic error while trying to process a scan report.
"No Data Allowance Licensing detected"	Oleander does not have a Data Allowance record.
"Error reading the scan report"	Generic error while trying to read a scan report.
"Probing path can be launched only on folders. Files are not supported"	Scan failed because a file was specified as target instead of a folder.
"Probing a File or Directory that does not exist"	Scan failed because the specified directory does not exist.

SUNDEW ERROR

Error Message	Explanation
"CLIENT_CREDENTIAL_PARTITION is not set."	CLIENT_CREDENTIAL_PARTITION variable is not set in the config object.
"Error trying to ask for license status", "error", err	Sundew failed trying to request DDC status from

Error Message	Explanation
	Oleander.
"Error trying to start the scan service", "error", err	Sundew failed trying to start ER2.
"Connectivity test to the scan service failed", "error", conErr	Sundew failed trying to ping ER2. ER2 is down.
"Error during validation of received license", "error", err	Sundew failed trying to validate the DDC license retrieved from Oleander.
"Error parsing the license ID", "error", err	Sundew failed trying to parse the DDC license ID received from Oleander.
"Error while injecting license inside scan service", "error", err	Sundew failed trying to inject the generated license into ER2.
"Error stopping scan service", "error", err	Sundew failed trying to stop ER2.
"Error closing the response body", "error", err	Sundew failed trying to close the HTTP response body from a request.
"Error closing sundew", "error", err	Error when trying to close the Sundew service.

Checking the system health

If you run into a problem with DDC, use this procedure to check the system health before you report anything to [Thales Customer Support](#).

1. (if CM is clustered) Check there is a defined DDC active node and check if it is part of the cluster.

Refer to the information in [Identifying the Active DDC Node](#) for details.

2. Check the available disk space using Linux console or disk-usage API. If DDC node free disk space below 20% the scan service will not be available.

Disk usage API

Use the GET disk-usage API as described in the [CipherTrust Manager API Guide](#).

API: /v1/ddc/server-statistics/disk-usage

Description: Returns disk usage statics of scanner service.

Sample response

```
{
  "diskUsage": 82.23,
  "degradedMode": false,
  "error": ""
}
```

Linux console

- a. Access the Linux console (of the DDC active node in case of a CM cluster) as the user **ksadmin**.
- b. Run the `df -h` command and verify that the "Use%" value for the "/" partition is under 80%.

```
ksadmin@ciphertrust:~$ df -h

Filesystem Size Used Avail Use% Mounted on
udev 7.8G 0 7.8G 0% /dev
tmpfs 1.6G 3.7M 1.6G 1% /run
/dev/sda2 49G 12G 35G 26% /
tmpfs 7.9G 0 7.9G 0% /dev/shm
tmpfs 5.0M 0 5.0M 0% /run/lock
tmpfs 7.9G 0 7.9G 0% /sys/fs/cgroup
/dev/sda1 464M 130M 307M 30% /boot
tmpfs 1.6G 0 1.6G 0% /run/user/0
```

3. Check the status of the DDC License.

Refer to the information in [Viewing the license status](#) for details.

4. Check the configuration and status of the TDP connection.

Refer to the information in [Configuring TDP](#) for details.

5. Check TDP health status.

Refer to the [Thales Data Platform Documentation](#) for details.

6. Check the CipherTrust Manager logs and look for messages related to Oleander and Sundew.

- a. Access to the linux console of the DDC active node as the **ksadmin** user.

- b. To see the DDC logs in real time, run the following command:

```
tail -f /opt/keysecure/logs/keysecure.system.log | egrep "\| (oleander|sundew)"
```

- c. To copy all the ddc logs into a single file, run:

```
> /tmp/ddc.log
ls -r /opt/keysecure/logs/keysecure.system.log* | while read FILE; do ext="${FILE##*.}"; if [ "$ext" = "gz" ]; then CMD="zcat"; else CMD="cat"; fi; $CMD $FILE | egrep "\| (oleander|sundew)" >> /tmp/ddc.log; done
```

- d. To copy the ddc logs filtered by a specific date (YYYY-MM-DD), run:

```
> /tmp/ddc-YYYY-MM-DD.log
ls -r /opt/keysecure/logs/keysecure.system.log* | while read FILE; do ext="${FILE##*.}"; if [ "$ext" = "gz" ]; then CMD="zcat"; else CMD="cat"; fi; $CMD $FILE | egrep "^YYYY-MM-DD" | egrep "\| (oleander|sundew)" >> /tmp/ddc-YYYY-MM-DD.log; done
```

Tip

Replace the *YYYY-MM-DD* bit with the date that you want to filter by.

7. Agent connectivity and clock.

▪ All agents

- Go to the Agents page, by clicking the Data Discovery and Classification panel in the CipherTrust Manager dashboard, then click to expand the Data Stores menu in the navigation panel on the left, then click Agents.
- Check whether agents are listed there with right status.
- Check if agent clocks are ok.

For more details please refer to information in [Agents](#).

▪ Specific Agent

- a. Verify the logs generated by the DDC agent.
 - In Linux: `/var/lib/er2/agent.log`
 - In Windows: `C:\Program Files (x86)\Ground Labs\Enterprise Recon 2\agent.log`
- b. Verify the connection from the agent to the CM server.
 - In Linux:

Run the `er2-config -t` command:

```
root@agent1:~# er2-config -t
Master server is not specified, using the default master server: sundew
Testing connection setting...
```

```
Test SUCCESS. Saving settings
Target initial group is not specified, using the default public key [prei
nstalled]
Configuration updated, please restart agent service
The configuration has been saved. Please restart the agent for the change
s to take effect.
```

- In Windows:
 - Open the **Configure Agent** application.
 - Ensure it has the correct server IP or hostname (DDC active node in case of a CM cluster).
 - Click the **Test Connection** button.
- c. Verify the disk space.

Resolving high disk usage issues

DDC performs automatic disk clean up in the backend when a scan run is complete or stopped. However, this process is skipped for a failed scan and all other subsequent scans, whether they fail or succeed. As a result, disk consumption may increase over time, potentially exceeding threshold levels.

Tip

Use the Linux console or disk-usage API to regularly monitor disk usage, and take proactive measures to prevent disk consumption from exceeding threshold levels. See step 2 under [Checking the System Health](#) for details.

DDC continuously monitors disk usage and displays alert banners when disk consumption reaches a certain threshold value.

Warning banner

DDC displays a warning banner when disk usage reaches 60% or above.



When you see a warning banner, perform the following tasks to manage disk space:

- Start the disk clean up process via warning banner.
 - a. Click **Start Clean Up Process**.
 - b. In the consent dialog, click **Clean Up**.

This action will stop all running scans and initiate the clean-up process. After the clean up process is

complete, you can rerun the scans manually.

Alternatively, click **Cancel** and allow the scan runs to complete or stop them manually, before restarting the clean-up process.

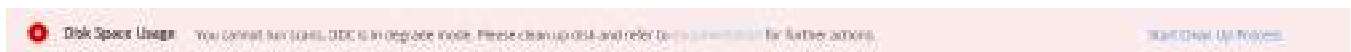
Note

After performing cleanup, it takes some time for DDC to synchronize and reflect the actual disk space, and stop displaying the warning banner.

- Delete backup files to reduce disk consumption.
- Increase the disk space.

Degraded banner

DDC enters the degraded mode when disk usage reaches 85% or above and displays a degraded banner with an option to initiate clean up process. DDC exits the degraded mode when the disk space is reduced to 70% or below.



Note

When degraded mode is reached, DDC becomes inoperable and all the APIs related to Scans, Datastores, Agents, Classification Profiles, Infotypes, and Report will not function.

When you see a degraded banner, perform the following tasks to manage disk space:

- Start the disk clean up process via degraded banner.
 - a. Click **Start Clean Up Process**.
 - b. In the consent dialog, click **Clean Up**.

During the clean up process:

- All the running, paused, and auto-paused scans will fail.
- No trace logs will be available for the failed scans.

Note

After performing cleanup, it takes some time for DDC to synchronize and reflect the actual disk space, and stop displaying the degraded banner.

- Delete backup files to reduce disk consumption.
- Increase the disk space.

Note

When disk cleanup is performed by increasing the disk space and/or deleting backup files, all running scans will continue to proceed uninterrupted. Additionally, if a scan was paused, it can be manually resumed.

Appendix

Information types

Infotype Name	Category	Region
AES Key	Personal Data	Global
American Express	Financial	Global
Artifactory Token	Personal Data	Global
Australian Bank Account Number	Financial	Oceania
Australian Business Number	Financial	Oceania
Australian Company Number	Financial	Oceania
Australian Driver License Number	Personal Data	Oceania
Australian Healthcare Identifier - Organisation	Medical	Oceania
Australian Individual Healthcare Identifier	Medical	Oceania
Australian Mailing Address	Personal Data	Oceania
Australian Medicare Card	Medical	Oceania
Australian Medicare Provider	Medical	Oceania
Australian Passport Number	Personal Data	Oceania
Australian Tax File Number	National ID	Oceania

Infotype Name	Category	Region
Australian Telephone Number	Personal Data	Oceania
Austrian Driver License Number	Personal Data	Europe
Austrian Mailing Address	Personal Data	Europe
Austrian Passport Number	Personal Data	Europe
Austrian Personalausweis	National ID	Europe
Austrian SSN	National ID	Europe
Austrian Telephone Number	Personal Data	Europe
AWS Key ID	Personal Data	Global
Azure Storage Key	Personal Data	Global
Basic Auth Secret	Personal Data	Global
Belgian Driver License Number	Personal Data	Europe
Belgian eID	National ID	Europe
Belgian National Number	National ID	Europe
Belgian Passport Number	Personal Data	Europe
Belgian Telephone Number	Personal Data	Europe
Blood Type	Medical	Global
Brazilian CPF	National ID	Americas
Brazilian Registro Geral	National ID	Americas
Bulgarian EGN	National ID	Europe
Canadian Bank Account Number	Financial	Americas
Canadian Health Service Number	Medical	Americas
Canadian Mailing Address	Personal Data	Americas
Canadian Passport Number	Personal Data	Americas
Canadian Personal Health Identification Number (PHIN)	Medical	Americas
Canadian Social Insurance Number	National ID	Americas
Canadian Telephone Number	Personal Data	Americas
Canadian Driver License Number	Personal Details	Americas
Chilean RUN	National ID	Americas
China Union Pay	Financial	Global
Cloudant Credentials	Personal Data	Global
Credentials password	Personal Data	Global
Credentials username	Personal Data	Global
Croatian OIB	National ID	Europe

Infotype Name	Category	Region
Cypriot Passport Number	Personal Data	Europe
Czech Republic RC	National ID	Europe
Danish CPR	National ID	Europe
Danish Driver License Number	Personal Data	Europe
Danish Passport Number	Personal Data	Europe
Date Of Birth (under 18)	Personal Data	Global
Date Of Birth	Personal Data	Global
DB2 Credentials	Personal Data	Global
DH Key	Personal Data	Global
Diners Club	Financial	Global
Discord Token	Personal Data	Global
Discover	Financial	Global
Drug Enforcement Agency Number	Medical	Americas
DSA Public Key	Personal Data	Global
Dutch Burgerservicenummer	National ID	Europe
Dutch Driver License Number	Personal Data	Europe
Dutch NIK	National ID	Europe
Dutch Passport Number	Personal Data	Europe
Dutch Telephone Number	Personal Data	Europe
ECC Public Key	Personal Data	Global
Email addresses	Personal Data	Global
Ethnicity (English)	Personal Data	Global
European EHIC	Medical	Europe
Finnish HETU	National ID	Europe
French Carte Vitale	National ID	Europe
French CNI	National ID	Europe
French Driver License Number	Personal Data	Europe
French INSEE	National ID	Europe
French Mailing Address	Personal Data	Europe
French Passport Number	Personal Data	Europe
French Telephone Number	Personal Data	Europe
Gambian National Identification Number	National	Africa
Gender (English)	Personal Data	Global

Infotype Name	Category	Region
Generic Bank Account Number	Financial	Global
German Driver License Number	Personal Data	Europe
German Mailing Address	Personal Data	Europe
German Passport Number	Personal Data	Europe
German Personalausweis	National ID	Europe
German Telephone Number	Personal Data	Europe
Github Token	Personal Data	Global
Greek AFM	National ID	Europe
Greek AMKA	National ID	Europe
Greek Passport Number	Personal Data	Europe
Hong Kong ID	National ID	Asia
Hungarian Personal ID	National ID	Europe
IBM Cloud IAM Key	Personal Data	Global
IBM COS HMAC Credentials	Personal Data	Global
Icelandish Kennitala	National ID	Europe
Indian Aadhaar Number	National ID	Asia
Indian Address	Personal Data	Asia
Indian Bank Account Number	Financial Data	Asia
Indian Driving License Number	Personal Data	Asia
Indian Marital Status	Personal Data	Asia
Indian MGNREGA Job Card ID	National ID	Asia
Indian Name	Personal Data	Asia
Indian PAN (Juridical) Number	National ID	Asia
Indian Passport Number	Personal Data	Asia
Indian Phone Number	Personal Data	Asia
Indian Ration Card Number	National ID	Asia
Indian Voter ID	National ID	Asia
International Bank Account Number (IBAN)	Financial	Global
IP Address	Personal Data	Global
Iranian National Identification Number	National	Asia
Irish Driver License Number	Personal Data	Europe
Irish Passport Card Number	Personal Data	Europe

Infotype Name	Category	Region
Irish Passport Number	Personal Data	Europe
Irish Personal Public Service Number	National	Europe
Irish Telephone Number	Personal Data	Europe
ISO8583 message with PAN	Financial	Global
Israeli Bank Account Number	Financial	Asia
Israeli Identity Number	National ID	Asia
Italian CARTA D'IDENTITÀ	National ID	Europe
Italian Codice Fiscale	National ID	Europe
Italian Driver License Number	Personal Data	Europe
Italian Mailing Address	Personal Data	Europe
Italian Passport	Personal Data	Europe
Italian Telephone Number	Personal Data	Europe
Japanese Bank Account Number	Financial	Asia
Japanese Driver License Number	Personal Data	Asia
Japanese Passport Number	Personal Data	Asia
Japanese Resident Registration Number	National	Asia
Japanese Social Insurance Number (SIN)	National	Asia
JCB	Financial	Global
JSON Web Token	Personal Data	Global
Laser	Financial	Global
Latvian Personas Kods	National ID	Europe
License Number	Personal Data	Global
Login credentials	Personal Data	Global
Luxembourg Driver License Number	Personal Data	Europe
Luxembourg ID	National ID	Europe
Luxembourg Passport Number	Personal Data	Europe
Luxembourg Phone Number	Personal Data	Europe
MAC Address	Personal Data	Global
Macedonian UMCN	National ID	Europe
Maestro	Financial	Global
Mailchimp Access Key	Personal Data	Global

Infotype Name	Category	Region
Malaysian NRIC	National ID	Asia
Maltese eID	National ID	Europe
Mastercard	Financial	Global
Medicare Beneficiary Identifier (MBI)	Medical	North America
Mexican CURP	National ID	Americas
Mongo DB Credentials	Personal Data	Global
MSSQL Database Credentials	Personal Data	Global
MySQL Database Credentials	Personal Data	Global
New Zealand Inland Revenue Number	National ID	Oceania
New Zealand Mailing Address	Personal Data	Oceania
New Zealand Passport Number	Personal Data	Oceania
New Zealand Telephone Number	Personal Data	Oceania
Norwegian Birth Number	National ID	Europe
Norwegian Driver License Number	Personal Data	Europe
Norwegian Passport Number	Personal Data	Europe
NPM token	Personal Data	Global
Oracle Database Credentials	Personal Data	Global
Passport Number	Personal Data	Global
Peoples Republic of China ID	National ID	Asia
Personal Names (Austrian)	Personal Data	Europe
Personal Names (Belgian)	Personal Data	Europe
Personal Names (English)	Personal Data	Global
Personal Names (French)	Personal Data	Europe
Personal Names (German)	Personal Data	Europe
Personal Names (Italian)	Personal Data	Europe
Personal Names (Netherlands)	Personal Data	Europe
Personal Names (Polish)	Personal Data	Europe
Personal Names (Portuguese)	Personal Data	Europe
PGP Public Key	Personal Data	
Polish Driver License Number	Personal Data	Europe
Polish Identity Card	National ID	Europe
Polish Mailing Address	Personal Data	Europe

Infotype Name	Category	Region
Polish Passport Number	Personal Data	Europe
Polish PESEL	National ID	Europe
Polish Telephone Number	Personal Data	Europe
Portuguese Citizen's Card	National ID	Europe
Portuguese Driver License Number	Personal Data	Europe
Portuguese Fiscal Number	National ID	Europe
Portuguese Identity Number	National ID	Europe
Portuguese Mailing Address	Personal Data	Europe
Portuguese Passport Number	Personal Data	Europe
Portuguese Phone Number	Personal Data	Europe
PostgreSQL Database Credentials	Personal Data	Global
Private Key	Personal Data	Global
Private Label Card	Financial	Global
Profanity (English)	Personal Data	Global
Redis Credentials	Personal Data	Global
Religion (English)	Personal Data	Global
Romanian Identity Card	National ID	Europe
Romanian Numerical Personal Code	National ID	Europe
RSA Public Key	Personal Data	Global
Saudi Arabia National ID	National ID	Asia
SendGrid API Key	Personal Data	Global
Serbian UMCN	National ID	Europe
Singaporean Mailing Address	Personal Data	Asia
Singaporean NRIC	National ID	Asia
Singaporean Passport Number	Personal Data	Asia
Singaporean Telephone Number	Personal Data	Asia
Singaporean Bank Account Number	Personal Data	Asia
Singaporean Driver License Number	Personal Data	Asia
Slack Token	Personal Data	Global
Slovakian RC	National ID	Europe
Slovenian EMSO	National ID	Europe

Infotype Name	Category	Region
SoftLayer Credentials	Personal Data	Global
South African Identity Number	National ID	Africa
South Korean Corporation Registration Number (법인등록번호)	Financial	Asia
South Korean Driver License Number	Personal Data	Asia
South Korean Foreigner Number	National ID	Asia
South Korean Gwangju Bank (광주은행) Account Number	Financial	Asia
South Korean Jeju Bank (제주은행) Account Number	Financial	Asia
South Korean Jeonbuk Bank (전북은행) Account Number	Financial	Asia
South Korean KB Bank (국민은행) Account Number	Financial	Asia
South Korean KEB Hana Bank (KEB하나은행) Account Number	Financial	Asia
South Korean NH Bank (농협은행) Account Number	Financial	Asia
South Korean Passport	Personal Data	Asia
South Korean Phone Number	Personal Data	Asia
South Korean RRN	National ID	Asia
South Korean Shinhan Bank (신한은행) Account Number	Financial	Asia
South Korean Taxpayer Identification Number (사업자등록번호)	Financial	Asia
Spanish DNI	National ID	Europe
Spanish Driver License Number	Personal Data	Europe
Spanish NIE	National ID	Europe
Spanish Passport Number	Personal Data	Europe
Spanish Social Security Number	National ID	Europe
Spanish Telephone Number	Personal Data	Europe
Square Oauth Secret	Personal Data	Global
Sri Lankan National Identity Card	National ID	Asia
SSH Private Key	Personal Data	Global

Infotype Name	Category	Region
SSH Public Key	Personal Data	Global
Stripe Access Key	Personal Data	Global
Swedish Driver License Number	Personal Data	Europe
Swedish Nationellt ID-kort	National ID	Europe
Swedish Passport Number	Personal Data	Europe
Swedish Personnummer	National ID	Europe
SWIFT Code	Financial	Global
Swiss Social Security Number	National ID	Europe
Taiwanese ID	National ID	Asia
TDES Key	Personal Data	Global
Thai Population Identification Code	National ID	Asia
Troy	Financial	Global
Turkish Identification Number	National ID	Europe
Turkish Tax Identification Number	National ID	Europe
Turkish Telephone Number	Personal Data	Europe
Twilio API Key	Personal Data	Global
United Arab Emirates ID	National ID	Asia
United Kingdom Community Health Index	Medical	Europe
United Kingdom Driver License Number	Personal Data	Europe
United Kingdom Electoral Roll Number	Personal Data	Europe
United Kingdom Health and Care Number	Medical	Europe
United Kingdom Mailing Address	Personal Data	Europe
United Kingdom National Health Service Number	Medical	Europe
United Kingdom NI Number	National ID	Europe
United Kingdom Passport Number	Personal Data	Europe
United Kingdom Self Assessment UTR Number	National ID	Europe
United Kingdom Telephone Number	Personal Data	Europe
United Kingdom VAT Number	Financial	Europe
United States Bank Account Number	Financial	Americas

Infotype Name	Category	Region
United States Driver License Number	Personal Data	Americas
United States Health Insurance Claim Number	Medical	Americas
United States Health Plan Identifier	Medical	Americas
United States Individual Taxpayer Identification Number (ITIN)	National ID	Americas
United States Mailing Address	Personal Data	Americas
United States National Provider Identifier	Medical	Americas
United States Passport Card Number	Personal Data	North America
United States Passport Number	Personal Data	North America
United States Routing Transit Number	Financial	Americas
United States Social Security Number	National	Americas
United States Telephone Number	Personal Data	Americas
Visa	Financial	Global
Yugoslavia UMCN	National ID	Europe

DDC ML information types

Infotype Name	Category	Region
AWS Access Key ID (ML)	Cryptographical	Global
AWS Secret Access Key (ML)	Cryptographical	Global
Mailing Address (ML)	Personal Data	Global
Email addresses (ML)	Personal Data	Global
GPS (ML)	Personal Data	Global
Nationality (ML)	Personal Data	Global
Organization (ML)	Personal Data	Global
Personal Names (ML)	Personal Data	Global
Political Party (ML)	Personal Data	Global
Religion (ML)	Personal Data	Global
RSA-2048 Private Key (ML)	Cryptographical	Global
Sexual Orientation (ML)	Personal Data	Global
URL (ML)	Personal Data	Global

Category and Subcategories Classification

Documents are classified into following categories and subcategories. These categories and subcategories are independent of each other so that each subcategory can be associated with any category.

Category

- Finance
- Healthcare
- Legal
- HR

Subcategory

- RecordsManagement
- CertificateOfEmployment
- IndemnityAgreement
- SalaryStructure
- BusinessLicense
- OfferLetter
- Diversity
- OrganizationChart
- DressCode
- HIPAA
- DoNotResuscitateForm
- PerformanceImprovementPlan
- ArticlesOfOrganization
- TimeOff
- BusinessBalanceSheet
- Immunization
- OperativeReport

-
- DataPrivacy
 - ConflictOfInterest
 - BusinessInsurance
 - Will
 - PerformanceAppraisal
 - BusinessAccountPayable
 - Telecommuting
 - MissionVisionAndStrategy
 - MissedAppointmentPolicy
 - MeetingMinutes
 - Recruitment
 - ShareHolderAgreement
 - LiabilityWaiver
 - BusinessPayroll
 - Harassment
 - EmploymentContract

Supported formats for standard agent

Files

Type	Format
Compressed	bzip2, Gzip (all types), TAR, Zip (all types)
Databases	Access, DBase, SQLite, MSSQL MDF & LDF
Images	BMP, FAX, GIF, JPG, PDF (embedded), PNG, TIF
Microsoft Backup Archive	Microsoft Binary / BKF
Microsoft Office	v5, 6, 95, 97, 2000, XP, 2003 onwards
Open Source	Star Office / Open Office / Libre Office
Open Standards	PDF, RTF, HTML, XML, CSV, TXT

Office files

WORD

- **Legacy:** Legacy filename extensions denote binary Microsoft Word formatting that became outdated with the release of Microsoft Office 2007. Although the latest version of Microsoft Word can still open them, they are no longer developed. Legacy filename extensions include:
 - .doc – Legacy Word document; Microsoft Office refers to them as "Microsoft Word 97 – 2003 Document"
 - .dot – Legacy Word templates; officially designated "Microsoft Word 97 – 2003 Template"
 - .wbk – Legacy Word document backup; referred as "Microsoft Word Backup Document"
- **OOXML:** Office Open XML (OOXML) format was introduced with Microsoft Office 2007 and became the default format of Microsoft Word ever since. Pertaining file extensions include:
 - .docx – Word document
 - .docm – Word macro-enabled document; same as docx, but may contain macros and scripts
 - .dotx – Word template
 - .dotm – Word macro-enabled template; same as dotx, but may contain macros and scripts
 - .docb – Word binary document introduced in Microsoft Office 2007

EXCEL

- **Legacy:** Legacy filename extensions denote binary Microsoft Excel formats that became outdated with the release of Microsoft Office 2007. Although the latest version of Microsoft Excel can still open them, they are no longer developed. Legacy filename extensions include:
 - .xls – Legacy Excel worksheets; officially designated "Microsoft Excel 97-2003 Worksheet"
 - .xlt – Legacy Excel templates; officially designated "Microsoft Excel 97-2003 Template"
 - .xlm – Legacy Excel macro
- **OOXML:** Office Open XML (OOXML) format was introduced with Microsoft Office 2007 and became the default format of Microsoft Excel ever since. Excel-related file extensions of this format include:
 - .xlsx – Excel workbook
 - .xlsm – Excel macro-enabled workbook; same as xlsx but may contain macros and scripts
 - .xltx – Excel template
 - .xltm – Excel macro-enabled template; same as xltx but may contain macros and scripts

POWERPOINT

- **Legacy:**

- .ppt – Legacy PowerPoint presentation
- .pot – Legacy PowerPoint template
- .pps – Legacy PowerPoint slideshow

- **OOXML:**

- .pptx – PowerPoint presentation
- .pptm – PowerPoint macro-enabled presentation
- .potx – PowerPoint template
- .potm – PowerPoint macro-enabled template
- .ppam – PowerPoint add-in
- .ppsx – PowerPoint slideshow
- .ppsm – PowerPoint macro-enabled slideshow
- .sldx – PowerPoint slide
- .sldm – PowerPoint macro-enabled slide

ACCESS

- **Legacy:**

- .ade – Protected Access Data Project (not supported in 2013)
- .adp - Access Data Project (not supported in 2013)
- .mdb - Access Database (2003 and earlier)
- .cdb - Access Database (Pocket Access for Windows CE)
- .mda - Access Database, used for addins (Access 2, 95, 97), previously used for workgroups (Access 2)
- .mdt - Access Add-in Data (2003 and earlier)
- .mdf - Access (SQL Server) detached database (2000)
- .mde - Protected Access Database, with compiled VBA and macros (2003 and earlier)
- .ldb - Access lock files (associated with .mdb)

- **Available formats since Access 2007:**

-
- .accdb – The file extension for the new Office Access 2007 file format. This takes the place of the MDB file extension
 - .accde – The file extension for Office Access 2007 files that are in "execute only" mode. ACCDE files have all Visual Basic for Applications (VBA) source code hidden. A user of an ACCDE file can only execute VBA code, but not view or modify it. ACCDE takes the place of the MDE file extension
 - .accdt – The file extension for Access Database Templates
 - .accdr – is a new file extension that enables you to open a database in runtime mode. By simply changing a database's file extension from .accdb to .accdr, you can create a "locked-down" version of your Office Access database. You can change the file extension back to .accdb to restore full functionality

OUTLOOK

- .pst - Outlook
- .ost - Outlook
- .msg - Outlook
- .dbx - Outlook

OTHER

- .pub – a Microsoft Publisher publication
- .xps – a XML-based document format used for printing (on Windows Vista and later) and preserving documents

Databases

- Microsoft SQL
- Oracle
- IBM DB2
- PostgreSQL
- SAP HANA
- MySQL
- MongoDB

Big Data

- Hadoop

- Teradata

Binary Large Objects

Database	Object Type
Oracle	BLOB, CLOB
Microsoft SQL Server	VARBINARY, Filestream
PostgreSQL	bytea, text, Large Objects(oid)
MySQL	BLOB, TINYBLOB, MEDIUMBLOB, LONGBLOB
IBM DB2	BLOBs, CLOB, DBCLOB
Teradata	BLOBs
MongoDB	GridFS
SAP HANA	BLOB, NCLOB

Supported file formats for DDC ML

- Unstructured
 - DOCX
 - PDF
 - RTF
 - TXT
 - ZIP (contains the supported file format)
- Semi-structured
 - HTML
 - DOCX (Key-value pairs)
 - PDF (Key-value pairs)
 - RTF (Key-value pairs)
 - TXT (Key-value pairs)
 - ZIP (Key-value pairs)

Unsupported file formats for DDC ML

- All structured file formats including CSV, XLSX, and database files (RDBMS).

-
- Semi-structured file formats such as JSON, XML, and YAML.
 - Unstructured file formats such as PPT, PPTX, HEIC, and MPS.
 - All image, audio, and video file formats.
 - All formats not included in [supported list](#).
 - Files with bad or inappropriate context.

Configuration backup

You can back up and restore the DDC configuration by using the Backup/Restore functionality available in CipherTrust Manager. Such a backup will include the following elements:

- Data Stores
- Locations
- Classification Profiles
- Infotypes
- Report definitions

This backup **will not include** the information about the **scan executions**.

Creating/Restoring the configuration backup

To create or restore a backup of your DDC configuration:

1. Log in to CipherTrust Manager.
2. Click the **Admin Settings** link on the dashboard.
3. Select **Backups** from the sidebar on the left. This will display the **Backups** screen.
4. To create a backup of your DDC configuration, click the **Create Backup** button.
5. To restore your DDC configuration from a backup, click the **Upload Backup** button.

For more details refer to these sections of the CipherTrust Manager documentation:

- [CipherTrust Manager backup](#)
- [CipherTrust Manager scheduling backups](#)
- [CipherTrust Manager scheduler](#)

Configuration backup limitations

- The configuration backup references the DDC Active Node. Restoring the backup to a different CipherTrust Manager cluster leaves DDC referencing an invalid node, and therefore without any valid active node.
- The configuration backup contains the definition of the DDC resources (such as the Scan or Data Store definitions). Restoring from a backup that does not contain a certain resource (for example, a Custom Classification Profile) or a resource version after a scan had been completed causes a TDP scan execution data to point to an invalid resource identifier.

If you generate a report that points to the missing resource you may display incomplete data (such as not being able to display the resource name) and/or fail.

Creating/Restoring backup of scan executions

To back up or restore the your Data Discovery and Classification scan executions data you need to access the DDC data stored in Hadoop. For details, refer to the [Thales Data Platform Hadoop backup](#) section in the [Thales Data Platform Administrator Guide](#).

Mounting an NFS share

To mount an NFS share on a Proxy agent, run this command as root:

```
sudo mount <nfs-server-hostname|nfs-server-ipaddress>:</target/directory/share-name>
```

Issues encountered while browsing target paths

You may encounter following issues while trying to navigate target paths using **Browse** button.

Datastore	Scenario	Issue
AWS	Path inside Add Target Path field contains invalid folder name within valid Bucket name Example: <valid_bucket>/<invalid_folder>	No error toast message shows up. File browser displays " No paths to display ".
GMail	Path inside Add Target Path field contains invalid folder name within valid user account Example: <valid_useraccount>/<invalid_folder>	No error toast message shows up. File browser displays " No paths to display ".
GMail, Google Drive	Add Target Path field is empty or contains a path to a valid folder	User accounts are listed correctly, but targets within it are not displayed.
NFS	Add Target Path field contains	No error toast message shows up.

Datastore	Scenario	Issue
	invalid path	File browser displays "No paths to display" .
SharePoint Online, SharePoint Server	Add Target Path field contains: - Invalid folder in valid site collection - Invalid file within valid site collection - Invalid file with valid folder in a site collection Examples: <valid_site_collection>/:site:/file/<valid_folder>/<invalid_file> <valid_site_collection>/:site:/file/<invalid_folder> <valid_site_collection>/:site:/file/<invalid_file>	No error toast message shows up. File browser displays "No paths to display" .
SharePoint Online, SharePoint Server	Add Target Path field contains incorrect site collection	Arbitrary folders and files may get displayed without any error. Selecting these results as target leads to a failed scan.
Exchange Online	Add Target Path field is empty	System default groups are listed. Upon selecting one or more default groups, the scan is executed successfully without any matches. <i>System default groups cannot undergo scanning. Targets must be valid email or email groups.</i>

API request quota limit

Datastore	Quota Limit (per minute request count)
Gmail	15000
SharePoint Online	600
Exchange Online	800
OneDrive	1000

Using wildcard characters

Asterisk (*) and question mark (?) are two most popular wildcard characters that you can use to traverse locations without providing absolute paths.

In DDC, you can use these characters while applying scan filters to scan selective locations instead of recursively scanning entire directory structures of data store.

Wildcard	Meaning	Examples
*	Matches zero or more characters	- D:/A/* will traverse all locations starting with D:/A - D:/A/*/F/E will traverse all locations starting with D:/A and ending with /F/E */A/E will traverse all locations ending with /A/E *A/E* will traverse all locations that contains A/E - D:/A/*/B/C/* will traverse all locations that starts with D:/A/ and contains /B/C/ in the followed path.
?	Matches exactly one character	D:/A??? will traverse all locations starting with D:/A followed by any three characters.

Deployment Security

Security audit log event messages

The following table contains a complete list of security audit log event messages that DDC prints in the log file.

Message	Explanation
DDCScanClientInvalidCredentialsProbe	A probe with invalid credentials.
DDCScanClientUnexpectedErrorProbe	An unknown probe error.
DDCHDFSUpdateHDFSSettingsAuthenticationError	A failed authentication against HDFS updating HDFS
DDCHDFSBackgroundProcessAuthenticationError	A failed authentication against HDFS in background
DDCUnauthorizedCloneRequest	An unauthorized CLONE request.
DDCUnauthorizedGetRequest	An unauthorized GET request.
DDCUnauthorizedListRequest	An unauthorized LIST request.
DDCUnauthorizedListPaginatedRequestWithContext	An unauthorized LIST PAGINATED request with
DDCUnauthorizedCreateRequest	An unauthorized CREATE request.
DDCUnauthorizedUpdateRequest	An unauthorized UPDATE request.
DDCUnauthorizedListProvisionedRequest	An unauthorized LIST PROVISIONED request.
DDCUnauthorizedGetProvisionedRequest	An unauthorized GET PROVISIONED request.
DDCUnauthorizedGetActiveNodeRequest	An unauthorized GET ACTIVE NODE request.

Message	Explanation
DDCUnauthorizedTestConnectivityRequest	An unauthorized TEST CONNECTIVITY request.
DDCUnauthorizedGetLicenseRequest	An unauthorized GET LICENSE request.
DDCUnauthorizedDecryptRawDataFileRequest	An unauthorized DECRYPT RAW DATA FILE request.
DDCUnauthorizedGetDatastoreReportRequest	An unauthorized GET DATASTORE REPORT request.
DDCUnauthorizedFindScanRequest	An unauthorized FIND SCAN request.
DDCUnauthorizedScanActionRequest	An unauthorized SCAN ACTION request.
DDCResourceRetrievalGenericCloneError	A GENERIC CLONE request.
DDCResourceRetrievalGenericGetError	A GENERIC GET request.
DDCResourceRetrievalGenericListError	A GENERIC GET request.
DDCResourceRetrievalGenericListPaginatedRequest Error	A GENERIC LIST PAGINATED request.
DDCResourceRetrievalGenericCreateError	A GENERIC CREATE request.
DDCResourceRetrievalGenericUpdateError	A GENERIC UPDATE request.
DDCResourceRetrievalGenericListProvisionError	A GENERIC LIST PROVISION request.
DDCDatastoreDecryptDataEncryptionKeyNotFoundErr or	A GET KEY request.
DDCDatastoreEncryptDataError	An ENCRYPT DATA request.
DDCScanWatcherInterruptedTimeout	An INTERRUPTED TIMEOUT request.
DDCScanClientRetrieveScanTimeout	A RETRIEVE SCAN TIMEOUT request.
DDCScanActionRequest	A SCAN ACTION request.
DDCDatastoreUpdateRequest	A DATASTORE UPDATE request.
DDCDatastoreCreateRequest	A DATASTORE CREATE request.
DDCScanDeleteRequest	A SCAN DELETE request.
DDCSummaryReportGetRequest	A GET SUMMARY REPORT request.
DDCDatastoreDetailReportGetRequest	A GET DATASTORE DETAILS REPORT request.
DDCDataObjectsDetailReportGetRequest	A GET DATASTORE DETAILS REPORT request.
DDCInfotypesSummaryReportGetRequest	A GET INFOTYPES SUMMARY REPORT request.
DDCDataObjectsSummaryReportGetRequest	A GET DATAOBJECTS SUMMARY REPORT request.
DDCScanDetailsReportGetRequest	A GET SCAN DETAILS REPORT request.

Mitigating security risks

DDC provides you a handy method of mitigating security risks by means of the Agent and Data Store labels. In this section you can find a few procedures that you can use to improve the security of your DDC deployment.

In short, agent labels represent the capabilities and Data Store labels indicate the capabilities required by any agent to scan it. Therefore, in order to scan a particular Data Store, an agent must define all the labels defined for

that Data Store, but it may contain additional labels. You can leverage this functionality to mitigate the following security risks:

1. Ensure that only some hand-picked agents, strongly hardened and monitored, can access your sensitive Data Stores.
2. Ensure that attackers cannot access the Data Store credentials by registering a new agent that they control.
3. Respect network segmentation policies.

Restrict access to sensitive data stores

The default DDC behavior is to share the Data Store credentials with every agent to identify those with connectivity. In order to minimize the attack surface, restrict the agents receiving the credentials to access sensitive Data Stores to those that are properly hardened and monitored by your IT / security department.

1. Reserve a label to identify sensitive Data Stores. For example: SENSITIVE
2. Assign this label to any Data Store containing sensitive information.
3. Assign the SENSITIVE label to the selected agents.

Effect: DDC will only share the Data Store credentials with manually whitelisted agents.

Whitelist vetted agents

DDC identifies any agent installed as legitimate, so attackers that already control a host in the network can leverage this behavior to receive the Data Store credentials even if this host does not have network connectivity to the Data Stores. In order to minimize the attack surface, you may configure DDC to consider only vetted agents.

1. Reserve a DDC-wide label to identify vetted agents. For example: VETTED_AGENT
2. Assign the label to all Data Stores to ensure DDC only considers them to complete the scans.
3. Assign the label to all valid agents displayed in the agent list.

Effect: If an attacker registers a new agent, the attacker-controlled agent will not be considered by DDC to complete any scan nor will receive any Data Store credential.

Respect network segmentation

Companies usually segment the network and define policies restricting data movement between network security zones. As DDC considers all agent with connectivity to the Data Store to complete a scan, data may cross the boundaries and violate the company policies. In order to prevent this, ensure DDC uses agents on the same network security zone the data resides.

1. Ensure that you have labels matching the security zones defined by your corporate security policy. For example: SECURITY_LEVEL_1, SECURITY_LEVEL_2, SECURITY_LEVEL_3
 2. Assign each Data Store to the label indicating the security zone that they reside on.
 3. Deploy (at least) one agent in each security zone, and assign it the label representing the security zone they reside on.
-

Effect: When DDC selects the agent to complete any scan, it will only consider agents that reside in the same network security zone, so your data will never cross the security zone boundaries.

Solution Brief

CipherTrust Data Discovery and Classification Technical Brief

Bringing agility and
confidence to your data
management

cpl.thalesgroup.com

THALES
Building a future we can all trust

Contents

3	Executive summary
4	The role of CipherTrust Data Discovery and Classification (DDC)
5	How it all fits together
8	Categorizing sensitive data
9	Using agents for discovery
10	Analyzing scan results
12	Getting answers to key questions
13	Reasons for our choice of architecture
14	Integrating with other solutions
15	Key takeaways
16	Abbreviations and glossary

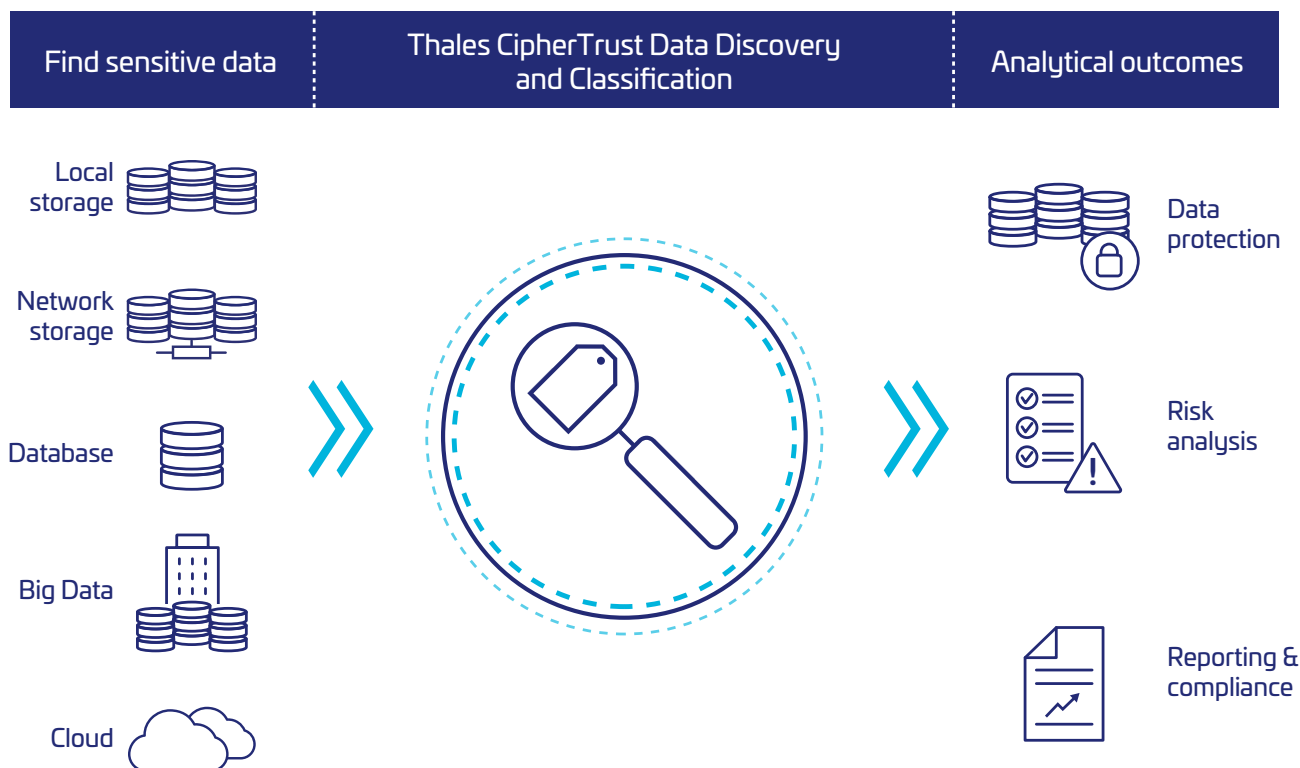
Executive summary

The rapid, often exponential, growth of data year-on-year in organizations like yours makes effective data management an extremely challenging proposition. An increasing switch to remote working raises the bar even higher with many data storage areas or volumes coming into play which are outside your IT team's direct control.

As part of the broader group within your organization responsible for data privacy and security compliance, ensuring no vulnerable areas are overlooked when implementing your data protection strategy is of paramount importance. After all, a data breach will inevitably cause severe business disruption in addition to large fines incurred for non-compliance with the seemingly endless stream of new or enhanced data privacy laws and regulations. Failing to prepare properly is definitely not a viable option for you.

A significant weakness often seen in typical data management implementations is the lack of visibility into the precise types of data being held across various local servers, network drives and increasingly cloud storage locations. Many organizations have been breached already, some have encountered near misses and others are migrating large workloads to the cloud without fully understanding the fundamental nature of the data itself and the exposure risks involved – situations for you to avoid.

Historically staff knowledge was sufficient and simple 'off-the-shelf' encryption methods from database vendors would satisfy needs – no longer with the vast data footprint that is widely dispersed and growing by the second. You inevitably need assistance to take control and keep your organization safe.



Thales CipherTrust Data Discovery and Classification is...

A tool that is significantly more effective and efficient than manual methods for data discovery as it scans the whole targets and not just sampling the data. It helps classify all your data, while supporting every mainstream operating system, data storage type, structured and unstructured data you are likely to possess. Existing processes and tools are supplemented, rather than replaced, providing high levels of automation in your search for sensitive data, wherever it resides. Ultimately, it can help you become more agile and support better data management decisions.



It finds sensitive data that you may not realize even exists in your organization, helping you eliminate threats to your business continuity and reduce your data footprint



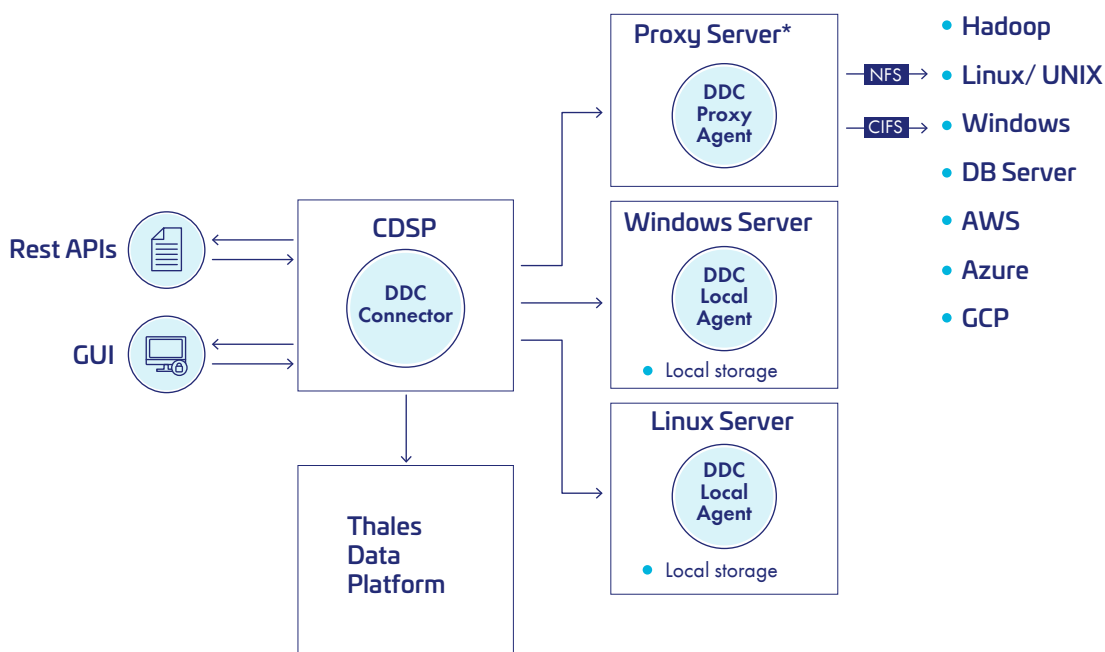
It enables you quickly to assess your risks of non-compliance and determine the appropriate protection actions to take



It provides confidence that you are protecting the right assets, including proprietary data or intellectual property, across all your data storage locations including the cloud

The role of CipherTrust Data Discovery and Classification (DDC)

DDC offers a more streamlined, accurate and unattended (or automated) approach in comparison to sensitive data searches using manual methods. DDC stands out from competitive products as it covers additional data storage types, extending the range of structured and unstructured data that can be analyzed while offering higher levels of accuracy by finding sensitive data other tools often miss.



Sensitive items found by DDC are not just restricted to data privacy laws and regulations. Fine tuning using custom infotypes (which you define) facilitates visibility on where data sensitive to your organization is present. By enabling critical assets (such as internal financial data, trade secrets, intellectual property and confidential business plans for example) to be located, their current levels of protection and associated risks of unintended exposure can be better understood. Considerable flexibility comes from running multiple scans simultaneously, each looking for different types of information (if desired) to create a faster and more efficient way to analyze your complete data footprint across a diverse range of storage locations. It is the ability to create multiple classification profiles (customized where needed) that underpins such performance and efficiency benefits that never could be achieved in a comparable timeframe using manual methods or less capable tools.

Discovering the data is only part of the task in hand – classification is equally important, providing deeper understanding of the types of data and their footprint in your organization. DDC does not move or modify your data, merely summarizing what it uncovers during scans in a non-intrusive manner.

You can see **exactly how your data is split** between the personal, financial, medical and national ID categories – you may be surprised that some of this even exists. At least now you will have insight into precisely where it resides to manage the business risk appropriately.

The insights available after scans have been performed provide **important guidance** on what you may need to do next, especially related to protection. You can choose how many scans you wish to incorporate in any given report. Each report improves data visibility, providing detailed insight into:

The names of the infotypes found together with the number of sensitive data matches

The risk associated with each of the data objects analyzed during the scan in question

DDC provides options to run scans at any time, over and over again if desired. The tool is intended to be used regularly to keep your insight up to date rather than just being a one-off activity. Since data is always in a state of flux, DDC is ready to convey the latest true representation of what you have and what you might need to do to keep your business safe.

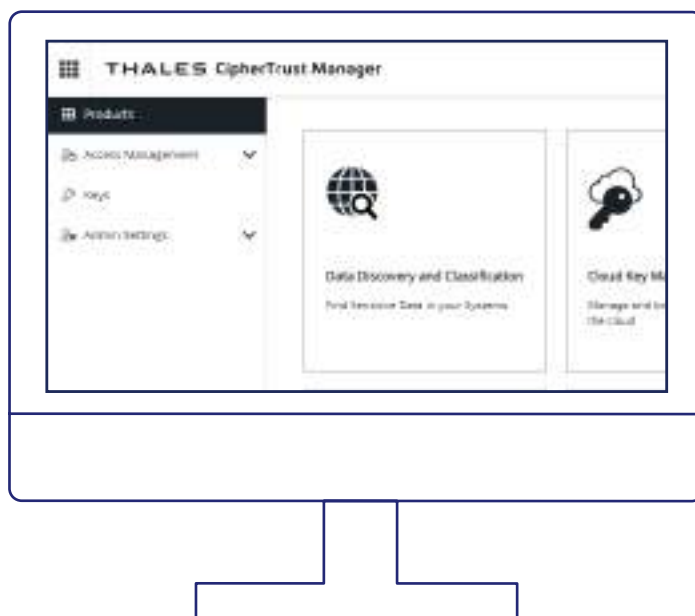
Secrets discovery is a critical component of cybersecurity and data protection. It involves identifying sensitive information, such as API keys, passwords, and social security numbers, that can be inadvertently exposed within systems or applications.

Secrets are difficult to gain visibility to as they reside in so many different places. Developers often scatter them in code repositories, configuration files and personal devices making both the discovery and management of them challenging. Older systems, false positives, and human error make discovering secrets a challenge, especially with cyberattackers constantly coming up with new ways to exploit these vulnerabilities. When secrets such as tokens, API keys, passwords, or usernames are discovered by threat actors, they can be used to break into IT systems. CipherTrust Data Discovery and Classification, using AI, proactively scans code for specific patterns, making developers aware of them before they become security threats. Secrets Discovery is a feature of Thales CipherTrust Data Discovery and Classification product and is the most comprehensive and reliable secrets discovery tool in the marketplace today. It proactively helps stop malicious actors before they gain unauthorized access to your data. Although there are other vendors for secrets discovery, they cannot tell the full story of Discover, Protect, Control like Thales can. The Thales solution is comprehensive, and doesn't require piecing together solutions, like many competitive solutions do.

How it all fits together

DDC is configured using a graphical user interface (launched from within the CipherTrust Manager console) or via the REST API. It offers a simple and intuitive workflow, enabling you to define data locations for interrogation, specify the types of data you wish to discover and decide when and how often you wish to repeat the scanning process. Ultimately the rich views provide a deep dive into the sensitive data matches, the associated protection status and risk assessment of the data locations in question. The main features are summarized in the sections below, providing a high level insight into the power, flexibility and control available to you.

The workflow for DDC is easy to follow via the user interface. **Three main ingredients** help define your environment – **data stores, classification profiles** and **infotypes**.



Typically, you might adopt the following sequence when using DDC for the first time:

Select and configure **data stores** of interest – we recommend concentrating on the most frequently accessed data stores first and a selected path rather than the whole store.

Choose one or more **classification profiles** – create custom classification profiles if necessary. During this process, select one or more **infotypes** of interest - use the custom infotype capability if the prebuilt list is insufficient.

Configure and start the scan – consider just selecting one data store for your first scan to test your configuration.



Data stores		
Local storage	SharePoint On Prem Exchange Server Local Windows and Linux	
Network Storage	Windows Share (CIFs/SMB) Unix File System (NFS)	
Database	IBM DB2 Microsoft SQL MongoDS MySQL	Oracle DB PostgreSQL SAP HANA SQL
Big Data	Hadoop clusters	Teradata
Cloud	AWS S3 Buckets Azure Blog Google Workspace (Gmail and Gdrive)	Azure Table Office 365 (Exchange, SharePoint, & OneDrive) SalesForce

The structured and unstructured data coverage provided by DDC is frequently being updated with new capabilities - check out the latest specifications [here](#).

Types of file supported

Databases	Access Dbase	SQLite MSSQL MDF & LDF	Microsoft Office	v5 6 95 97 2000	XP 2003 onwards Office Files: Word, Excel, PowerPoint, Access, Outlook, Other (.pub & .xps)
Images	BMP FAX GIF JPG	PDF (embedded) PNG TIF	Open Source	Star Office / Open Office / Libre Office	
Compressed	Bzip2 Gzip (all types)	TAR ZIP (all types)	Open Standards	PDF RTF HTML	XML
Microsoft Backup Archive	Microsoft Binary / BKF				CSV TXT

Regulations Supported

APA	Australia Privacy Amendment	HIPAA	Health Insurance Portability and Accountability Act
APPI	Act on Protection of Personal Information	KVKK	Turkish Personal Data Protection Law
CCPA	California Consumer Privacy Act	LGPD	General Data Protection Law (Brazil)
GDPR	Financial	NYDFS	New York State Department of Financial Services
GDPR	General Data Protection Regulation	PCI DSS	Payment Card Industry Data Security Standard
GDPR	Healthcare	SHIELD	Privacy Shield Framework
GDPR	National ID	UK-GDPR	General Data Protection Regulation (UK)
GDPR	Personal Details		

Core classification profiles are prebuilt and aligned with major data laws and regulations, both regional and global. You can make changes to these or add totally new profiles if required. Various shortcuts are provided in terms of editing and copying so that you can quickly create and test your own profiles. There is full flexibility in specifying various parameters including sensitivity levels and infotypes.

Sensitivity Levels Supported		
None	Public	Internal
Private	Restricted	

Categorizing sensitive data

DDC is prebuilt with various infotypes, over 250 and counting, covering the vast majority of regional and global data privacy laws and regulations. When a prebuilt classification profile template is selected, the appropriate subset of infotypes is added automatically. You can modify or extend this list by creating your own custom infotypes, allowing rules to be defined which describe precisely how the scanning engine should look for the data strings in question. Full access to the groups and categories is available as part of the definition so that data matches can be displayed in the appropriate sections of the scan results.

Infotype categories (prebuilt)		
Financial	Credit/ debit cards	Bank account info
Personal data	Email addresses Login credentials Card number Ethnicity License number Roll number Passport number Date of birth MAC address	Mailing address Telephone number Gender Religion IP address Phone number Name Profanity
Medical	Patient health data	
National ID	Personal identification	
Secrets	Secrets: Keys, credentials, others	

Using agents for discovery

Detailed data analysis is carried out by discovery agents which perform the scanning and report the results back to the DDC Connector for analysis and processing. Two types of agents exist – local and proxy. You would normally install local agents on data store locations (for which you have the appropriate access rights) to ensure the data never leaves the server for security or performance reasons. In contrast, proxy agents are used for network or remote data stores where a separate proxy server hosts the agent.

Type of agent	Pros	Cons
Local	Faster scan Data remains local No need for credential	Longer deployment time More complex management Only supports local data stores
Proxy	Faster to deploy and scale Ability to scan multiple data stores Supports multiple datastore types No resources consumed on target host	Data sent over network to agent prior to scanning Increased network load and data footprint Ideally should be located on same virtual LAN (VLAN)

Common benefits apply to both types of agent:



An **unlimited number** of agents is available for deployment at no additional cost since licensing is based on the volume of new data scanned



Multiple simultaneous sub-scans can be launched to reduce the time required for the overall scan



TLS encryption is available for all communications between data stores and agents to protect sensitive data from eavesdropping



Source data is untouched as agents only access a temporary data copy in internal memory during the scan



No sensitive data is stored during the process as only a summary of the scan results is sent to the DDC Connector



Scans can **still proceed** and record results even when connection to the DDC Connector is lost

With your agents installed, you can then start to think about what and how you wish to scan. We provide a great deal of flexibility in how scans are configured, managed and operated. Important parameters configurable for each scan independently include:

- Name (up to 64 characters)
- Description (up to 250 characters)
- One or more data stores (where to scan)
- One or more classification profiles (what to look for)
- Manual or scheduled operation (when to activate)

On completion of any given scan, it is added to a list of all scans that have been performed. Next, you would use the reporting tool to view detailed results from the scan. Depending on what is found you may make changes to the scan configuration and rerun the scan which is beneficial when working on different data stores or new types of compliance regulations for the first time.

Analyzing scan results

The results from historical and new scans are available for use with the built-in reporting tool. Three main views for scan reports are offered:

- **Scans**
- **Data stores**
- **Data objects**

Before detailed information is available, a report is generated which aggregates the results from one or more successful scans. Often for logistical, security or performance reasons you may wish to use multiple scans (each looking for different types of regulated data) to analyze your entire data footprint. With our flexible approach you can still do this and fine-tune each scan as required using multiple iterations, before generating a final report which can represent the current status of all your data. It is also easy to understand your organizational posture from a single regulation point of view.

Each tab within the report module offers an insight into slightly different aspects of your data – all however have a common goal in helping you identify quickly any sensitive data at risk.

Scans view

Provides a high level, aggregated scan view of the infotypes found (broken down by percentage in each category present) together with three other charts representing the presence of sensitive data objects in terms of category distribution, content types and file types in question.



Data stores view

Provides a list of the data stores with details including the risk, sensitivity level, infotypes found and the number of sensitive data objects present for each data store – this provides a rapid means of assessing at a high level whether or not sensitive data is at risk and which data stores are involved.



Data objects view

Provides a detailed list of all data objects scanned containing sensitive data, sorted in descending order of risk of exposure. For each sensitive data object in question, you can see where it is located, the data store in which it resides and the individual names and data matches for all of the associated infotypes found. Remediation status information is also present which provides a deeper insight into your sensitive data, confirming whether or not an encryption policy is active. No sensitive data is stored as part of the reporting process.



Exporting data

In addition to viewing objects using the built-in reporting tool, you can also export data in a NDJSON format for analysis by an external reporting tool such as one from Elastic used in Thales demonstrations



Reasons for our choice of architecture

Single integrated platform

We decided to make DDC an integral part of the Thales CipherTrust Data Security Platform because, in our experience, data discovery is the foundation of any effective data security strategy. This integrated platform approach enables CipherTrust Manager to act as the central management console for various connectors including data discovery, transparent encryption and tokenization. If you are using CTE already, you will be familiar with the console and will benefit from consistency in how common management tasks such as user groups, data policies, access control and logging are implemented.

Accurate pattern matching

Thales performed an extensive investigation to determine the best data discovery technology in the market. After much research and bench testing, the team experienced a significant difference using GLASS Technology™ over regex. GLASS is a proprietary market-leading technology from Ground Labs (a Thales Technology Partner) for defining and matching simple and complex patterns of data at scale as opposed to a developer code language not built specifically for purpose. It was designed from the ground up with the express purpose of finding patterns of data in modern data sets across both structured and unstructured data storage scenarios.

High performance discovery

An important goal was to provide a highly efficient and accurate discovery engine. Most data discovery solutions available today use the regular expression (regex) language (originating from the 1950's) to search for patterns inside text strings. This approach requires extensive developer knowledge to implement and the syntax is not based on English grammar. The biggest enemy of regex is performance – as pattern concurrency and complexity increases, the slower the performance achieved. Regex has limits to the patterns it can match accurately, making it far from ideal in the modern world where there are hundreds of different privacy regulations and a broad range of data management issues to address.

Easy customization capabilities

Customization support was another important criteria to supplement the comprehensive prebuilt templates. To address proprietary needs, GLASS data patterns can be written by non-developers using English-style grammar to express how a piece of data or pattern exists. Detailed analysis shows it has overcome all important regex limitations for pattern matching techniques, especially its ability to operate across multiple platform and cloud environments, while being very frugal in terms of CPU overhead. This made GLASS an ideal core component for embedding inside DDC, working efficiently and transparently as part of the CipherTrust platform in identifying and securing critical data.

Flexible licensing model

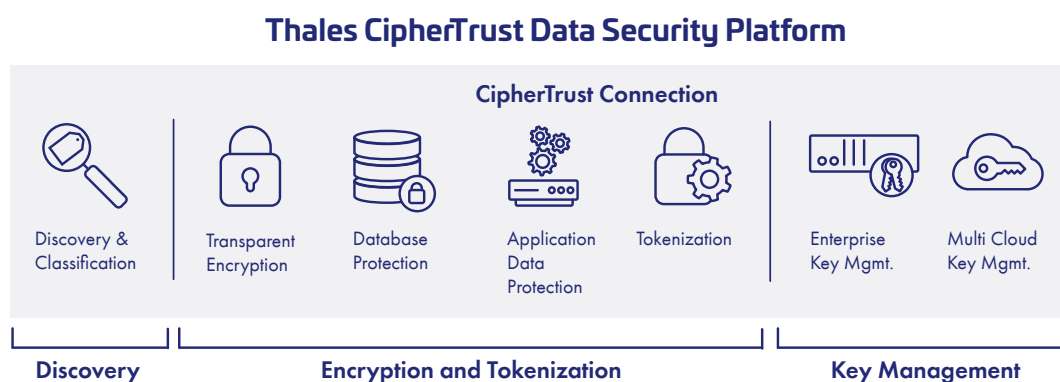
A consumption model based on total data scanned is used to license DDC, providing close alignment with many cloud subscription models and offering an unlimited number of discovery agents without incurring any additional fees. This makes it much easier for all your teams to have access to the data discovery capabilities, with the only real decision to make is how much overall data discovery capacity is required for your organization. Re-scanning existing data does not subtract from your remaining allowance, only new data scanned is counted.

Security first

Naturally as a market leading security company we design and implement secure solutions. DDC is no exception with strong emphasis on ensuring no sensitive data is stored or vulnerable to exposure:

- Ensuring that all data flowing between your data stores and the agents is protected using strong encryption
- Providing granular access controls so that you can tailor the solution to your exact needs

Integrating with other solutions



CipherTrust DataSecurity Platform

DDC is optimized for use as part of the CipherTrust Data Security Platform, an integrated suite of data-centric security products and solutions that unify data discovery, protection and control in one platform. Consequently you can cover all data protection needs using a single platform from one vendor, in this case Thales. This provides you with a full suite of data protection capabilities including key management, user access control, file-level encryption, application-layer encryption, database encryption, format-preserving encryption, tokenization and data masking.

CipherTrust Manager

All components utilize CipherTrust Manager as their management console, delivering central management of encryption keys, granular access control and security policy configuration. One key advantage is that our solution is designed to protect your data wherever it resides, on-premises or in the cloud and whatever storage mechanism is in use, such as files, databases, big data or containers. Ultimately you will end up requiring fewer resources dedicated to data security operations, meet your compliance obligations with greater confidence and reduce the business risk to your organization.

It is also possible to deploy DDC alongside other third party data protection tools, such as those supporting data encryption or tokenization for example. It is non-intrusive, complementing tools you have already, rather than replacing them. However, CipherTrust Manager is still necessary since it provides the sole method to launch the data discovery capability and to manage the licenses required for its operation together with keys required for encrypting sensitive data.

Some of the main reasons Thales customers have been using DDC in conjunction with third party products include:

We often get asked... **'What is the difference between CipherTrust Data Discovery and Classification and other Data Loss Prevention (DLP) solutions?'**

The DLP solutions focus on preventing sensitive data from leaving the organization's perimeter. DDC focuses on data privacy - identifying sensitive data, and getting a clear understanding of data and its risk. This enables appropriate steps to be taken to protect data and comply with data privacy and data security regulations.

- Trying to fill gaps in coverage since their existing discovery tool cannot scan all their structured and unstructured data
- Seeking higher accuracy with data matches to eliminate the limitations they are experiencing with regex technology
- Supporting a wider range of operating systems and platforms to interrogate more of their data footprint

Key takeaways

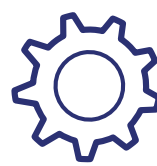
CipherTrust Data Security Platform



Discover

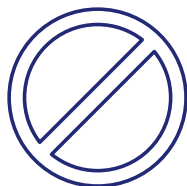


Protect



Control

Your data management and protection strategy can ultimately be improved significantly using CipherTrust Data Discovery and Classification. In conjunction with CipherTrust Manager and CipherTrust Transparent Encryption it provides considerable visibility and insight into your sensitive data while offering quick and effective protection options. It enables you to:



Block unnecessary data access to reduce risk of data breaches



Keep up to date with the evolving regulatory landscape to remain in compliance



Address digital transformation concerns relating to cloud security to support business and technology goals

****Try our free 90 day evaluation to see for yourself how it can help your organization now***

Abbreviations

ACL	Access Control List
CDSP	CipherTrust Data Security Platform
CIFS	Common Internet File System
CM	CipherTrust Manager
CTE	CipherTrust Transparent Encryption
DB	Database
DDC	CipherTrust Data Discovery and Classification
GUI	Graphical User Interface
NAS	Network-attached Storage
REST API	Representational State Transfer Application Programming Interface
TDP	Thales Data Platform
NAS	Network-attached Storage

Glossary

Classification profile	A classification profile uses a list of infotypes to define what kind of sensitive information to search for during a scan
Connector	A connector is a generic term which relates to the different licensable products or components (which includes DDC and CTE) that are managed using the CipherTrust Manager console
CTE agent	A CTE agent is a software component that is used to encrypt data associated with GuardPoints defined using CTE policies
Data match	A data match occurs during a scan when a qualified instance of an infotype actively included in the scope of the search is found
Data object	A file or database table located inside a data store is known as a data object
Data store	A data store is the entity where the data is actually stored, with DDC supporting various types - local, network, database, Big Data and cloud
DDC agent	A DDC agent is a software component that is used to scan a data store for specific types of data defined using infotypes in the associated
Discovery	classification profile
False negatives	Data discovery is the process of mapping an organization's data assets including their locations, types and sensitivity levels
False positives	False negatives are where the discovery process fails to identify one or more valid instances of sensitive data
GuardPoint	A GuardPoint specifies the list of folders or paths to be protected –access to files and encryption of files under the GuardPoint is controlled by security policies
Infotype	An infotype is used to categorize specific data (such as passport numbers or email addresses) to look for during a discovery scan, forming an integral component in the definition of a classification profile
Policy	A policy is a collection of rules that govern data access and encryption
Remediation	Remediation refers to the process of protecting (or securing) data that has been identified as vulnerable as the result of a scan – typical remediation methods include encryption, tokenization, data masking and access control
Resource set	A resource set refers to the files or directories to which the policy will apply together with the associated governing key rules
Risk	A risk is the presence of a sensitive data object in a data store and is directly related to the data matches found in the data object or data store
Scan	A scan is part of the discovery process that is used to search for sensitive data within data stores using criteria defined in classification profiles
Sensitive data object	A data object that contains any data match is known as a sensitive data object

Sensitivity level	The sensitivity level is a mandatory parameter (when defining data stores and classification profiles) relating to the degree of potential vulnerability of any given data object if exposed
Structured data	Structured data is highly organized and easily understood by machine language – examples include names, dates, addresses, credit card numbers, stock information, geolocation and more
Tag	A tag helps group data together and can be specified when creating data stores and classification profiles
Thales Data Platform	The Thales Data Platform (TDP) is a Big Data platform based on Hadoop technology which is used by DDC for various tasks including storage of scan results
Unstructured data	Unstructured data is information that either does not have a pre-defined data model (or schema) or is not organized in a pre-defined manner, making it unsuitable for storage in a relational database



Contact us

For all office locations and contact information,
please visit cpl.thalesgroup.com/contact-us

cpl.thalesgroup.com



Descrição
da solução

CipherTrust Data Discovery and Classification - dados técnicos

Traga agilidade e confiança
para seu gerenciamento
de dados

cpl.thalesgroup.com

THALES
Building a future we can all trust

Conteúdo

3	Sumário executivo
4	O papel da CipherTrust Data Discovery and Classification (DDC)
5	Como tudo se encaixa
8	Categorização de dados confidenciais
9	Uso de agente para descoberta
10	Análise de resultados da varredura
12	Respostas para as principais perguntas
13	Motivos da nossa escolha de tipo de arquitetura
14	Integração com outras soluções
15	Principais conclusões
16	Abreviações e glossário

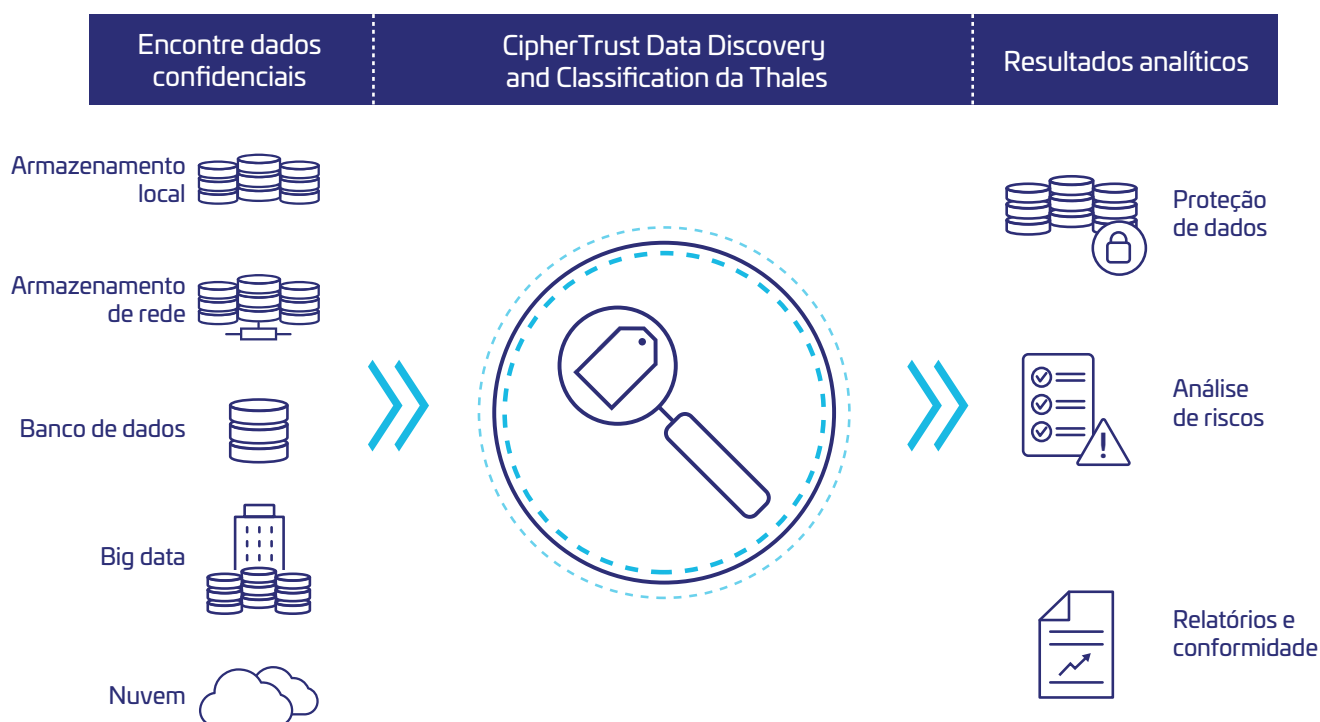
Sumário executivo

O crescimento rápido, muitas vezes exponencial, de dados ano a ano em empresas como a sua torna o gerenciamento eficaz de dados uma proposta extremamente desafiadora. A crescente mudança para o trabalho remoto eleva ainda mais o nível de exigência, com muitas áreas ou volumes de armazenamento de dados que estão fora do controle direto da sua equipe de TI.

Como parte do grupo mais amplo de sua empresa responsável pela conformidade com a privacidade e a segurança dos dados, é de suma importância garantir que nenhuma área vulnerável seja negligenciada ao implementar sua estratégia de proteção de dados. Afinal de contas, uma violação de dados inevitavelmente causará graves interrupções nos negócios, além de grandes multas incorridas pela não conformidade com o fluxo aparentemente interminável de leis e normas de privacidade de dados novas ou aprimoradas. Deixar de se preparar adequadamente definitivamente não é uma opção viável para você.

Um ponto fraco significativo frequentemente observado em implementações típicas de gerenciamento de dados é a falta de visibilidade dos tipos precisos de dados mantidos em vários servidores locais, unidades de rede e, cada vez mais, locais de armazenamento em nuvem. Muitas empresas já sofreram violações, algumas passaram por situações de quase violação e outras estão migrando grandes cargas de trabalho para a nuvem sem entender completamente a natureza fundamental dos dados em si e os riscos de exposição envolvidos, situações que você deve evitar.

Antigamente, o conhecimento da equipe era suficiente e os métodos simples de criptografia “prontos para uso” de fornecedores de banco de dados satisfaziam as necessidades, mas isso não é mais suficiente com a vasta superfície de dados que está amplamente dispersa e cresce a cada segundo. É inevitável que você precise de assistência para assumir o controle e manter sua empresa segura.



A solução CipherTrust Data Discovery and Classification da Thales é...

Uma ferramenta que é significativamente mais eficaz e eficiente do que os métodos manuais para a descoberta de dados, pois examina todos os alvos e não apenas uma amostragem dos dados. Esta ferramenta ajuda a classificar todos os seus dados, ao mesmo tempo em que oferece suporte a todos os principais sistemas operacionais, tipos de armazenamento de dados e dados estruturados e não estruturados que você provavelmente possui. Os processos e as ferramentas existentes são complementados, e não substituídos, proporcionando altos níveis de automação em sua busca por dados confidenciais, onde quer que eles estejam. Por fim, ela pode ajudar você a se tornar mais ágil e ajuda a tomar melhores decisões de gerenciamento de dados.



Encontra dados confidenciais que você talvez não saiba que existem em sua empresa, ajudando a eliminar ameaças à continuidade dos seus negócios e reduzir a superfície de dados



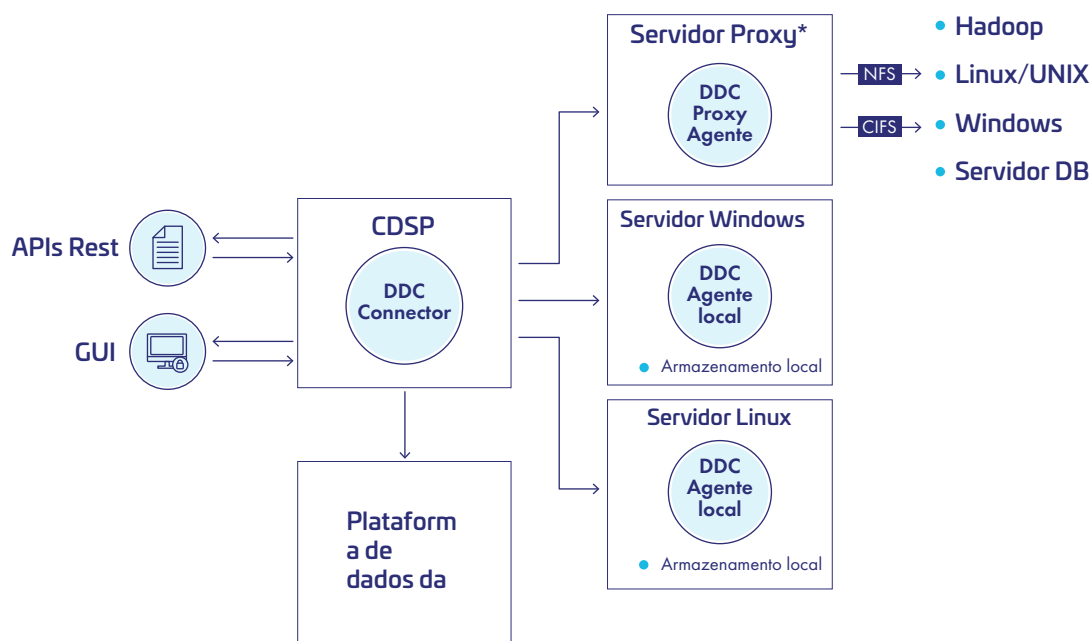
Permite avaliar rapidamente os riscos da falta de conformidade e determinar medidas de proteção adequadas



Garante a proteção dos dados certos, como dados proprietários ou propriedade intelectual, em todos os locais de armazenamento de dados, incluindo na nuvem

O papel da CipherTrust Data Discovery and Classification (DDC)

A DDC oferece uma abordagem mais simplificada, precisa e autônoma (ou automatizada) em comparação com as pesquisas de dados confidenciais que usam métodos manuais. A DDC se destaca dos produtos da concorrência por abranger tipos adicionais de armazenamento de dados, ampliando a gama de dados estruturados e não estruturados que podem ser analisados e, ao mesmo tempo, oferecendo níveis mais altos de precisão ao encontrar dados confidenciais que outras ferramentas geralmente deixam passar.



Os elementos confidenciais encontrados pela DDC não se restringem apenas às leis e normas de privacidade de dados. A fina sintonia que usa infotipos personalizados (definidos por você) facilita a visibilidade de onde os dados confidenciais da sua organização estão presentes. Ao permitir que itens críticos (como dados financeiros internos, segredos comerciais, propriedade intelectual e planos de negócios confidenciais) sejam localizados, seus níveis atuais de proteção e os riscos associados de exposição não intencional podem ser mais bem compreendidos. A flexibilidade considerável vem da execução de várias varreduras simultaneamente, cada uma buscando diferentes tipos de informações (se desejado) para criar uma maneira mais rápida e eficiente de analisar o volume completo de dados em uma ampla variedade de locais de armazenamento. É a capacidade de criar vários perfis de classificação (personalizados quando necessário) que sustenta esses benefícios de desempenho e eficiência que nunca poderiam ser obtidos em um período de tempo comparável usando métodos manuais ou ferramentas menos capazes.

Descobrir dados é apenas uma parte da tarefa em questão, a classificação é igualmente importante, fornecendo uma compreensão aprofundada dos tipos de dados e sua pegada em sua empresa. A DDC não move nem modifica seus dados, apenas resume o que descobre durante as varreduras de maneira não intrusiva.

Você pode ver **exatamente como seus dados estão divididos** entre as categorias dados pessoais, financeiros, de saúde e de identificação nacional, e talvez se surpreenda com a existência de alguns deles. Pelo menos agora você terá uma visão exata de onde eles se encontram para gerenciar adequadamente o risco comercial.

Os insights disponíveis após a realização das varreduras fornecem **orientações importantes** sobre o que você precisa fazer, especialmente em relação à proteção. Você pode escolher quantas varreduras deseja incorporar em um determinado relatório. Cada relatório melhora a visibilidade dos dados, fornecendo informações detalhadas sobre:

Nomes dos infotipos encontrados com o número de correspondências de dados confidenciais

O risco associado a cada um dos objetos de dados analisados durante a varredura em questão

O status de proteção para poder ver o que está protegido e o que não está

A DDC oferece opções para executar varreduras a qualquer momento e repetidas vezes, se você quiser. A ferramenta deve ser usada regularmente para manter sua percepção atualizada, em vez de ser apenas uma atividade eventual. Como os dados estão sempre em um estado de fluxo, a DDC está pronta para transmitir a representação real mais recente do que você tem e do que pode ser necessário fazer para manter sua empresa segura.

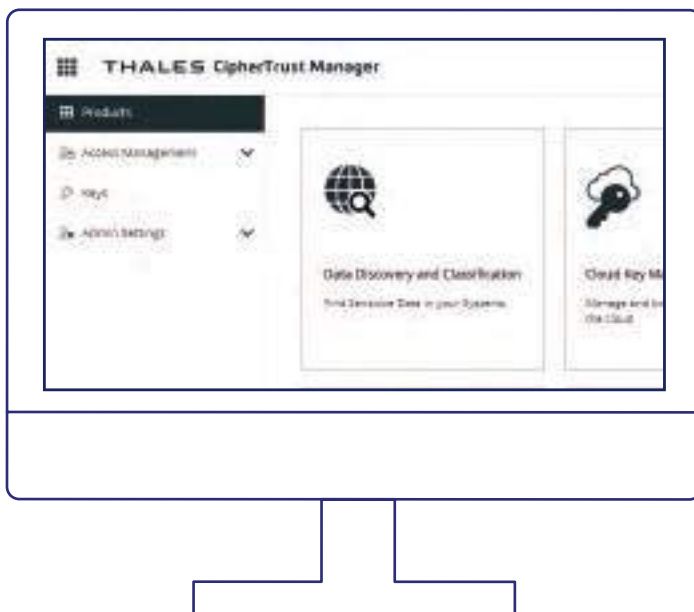
A descoberta de segredos é um componente essencial da cibersegurança e da proteção de dados. Isso envolve a identificação de informações confidenciais, como chaves de API, senhas e números da previdência social, que podem ser expostas inadvertidamente em sistemas ou aplicativos.

É difícil ter visibilidade de segredos, pois eles estão em muitos lugares diferentes. Os desenvolvedores geralmente os espalham em repositórios de código, arquivos de configuração e dispositivos pessoais, o que torna a descoberta e o gerenciamento deles um desafio. Sistemas mais antigos, falsos positivos e o erro humano tornam a descoberta de segredos um desafio, especialmente porque os cibercriminosos estão sempre criando novas maneiras de explorar essas vulnerabilidades. Quando segredos como tokens, chaves de API, senhas ou nomes de usuário são descobertos por criminosos, eles podem ser usados para invadir os sistemas de TI. A CipherTrust Data Discovery and Classification utiliza IA para examinar proativamente o código em busca de padrões específicos, tornando os desenvolvedores cientes deles antes que se tornem ameaças à segurança. Secrets Discovery é um recurso da CipherTrust Data Discovery and Classification da Thales e é a ferramenta de descoberta de segredos mais abrangente e confiável do mercado atual. Ele ajuda a interromper proativamente agentes criminosos antes que eles obtenham acesso não autorizado aos seus dados. Embora existam outros provedores para a descoberta de segredos, eles não sabem contar a história completa da descoberta, proteção e controle como a Thales pode. A solução da Thales é abrangente e não requer soluções de envio de peças, como acontece com muitas soluções da concorrência.

Como tudo se encaixa

A DDC é configurada usando uma interface gráfica de usuário (iniciada a partir do console CipherTrust Manager) ou por meio da API REST. Esta solução oferece um fluxo de trabalho simples e intuitivo, permitindo que você defina locais de dados para questionamentos, especifique os tipos de dados que deseja descobrir e decida quando e com que frequência deseja repetir o processo de varredura. As visualizações avançadas fornecem uma visão aprofundada das correspondências de dados confidenciais, do status de proteção associado e da avaliação de risco dos locais dos dados em questão. Os principais recursos estão resumidos nas seções abaixo, fornecendo uma visão bem definida do poder, da flexibilidade e do controle disponíveis para você.

O fluxo de trabalho da DDC é fácil de seguir por meio da interface do usuário. **Três ingredientes principais** ajudam a definir seu ambiente: **armazenamentos de dados, perfis de classificação e infotipos**.

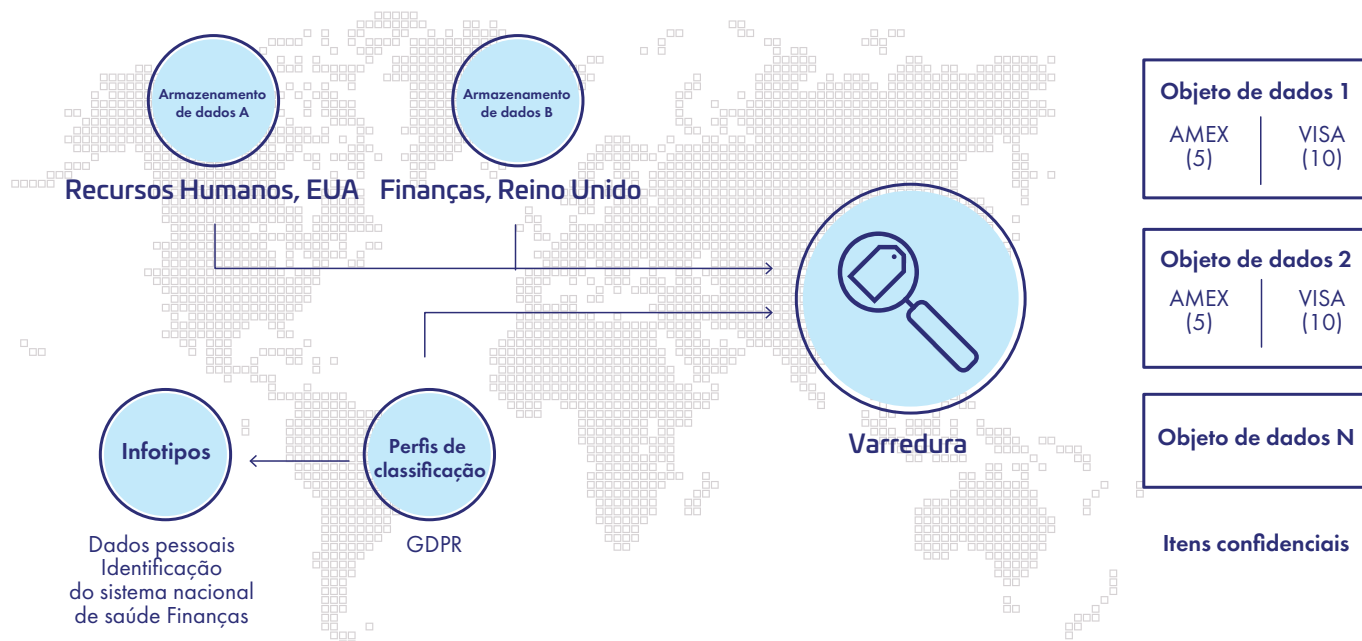


Normalmente, você pode adotar a seguinte sequência ao usar a DDC pela primeira vez:

Selecione e configure **armazéns de dados** de interesse – recomendamos primeiro se concentrar nos armazenamentos de dados mais frequentemente acessados e em um caminho selecionado em vez de todo o armazenamento.

Selecione um ou mais **perfis de classificação** – cria perfis de classificação personalizados, se necessário. Durante esse processo, seleciona um ou mais **infotipos** de interesse - utiliza a capacidade de infotipo personalizado se a lista criada for insuficiente.

Configure e inicie a **varredura** – considere a possibilidade de selecionar apenas um armazenamento de dados para sua primeira análise, a fim de testar sua configuração.



Armazenamentos de dados

Armazenamento local	Armazenamento local SharePoint sob demanda Servidor Exchange Windows e Linux locais	Armazenamento em rede - Windows Share (CIS/SMB) - Sistema de arquivos Unix (NFS)
Armazenamento em rede	Windows share (CIS/SMB)	
Banco de dados	BM DB2 Microsoft SQL MongoDS MySQL	Oracle DB PostgreSQL SAP HANA SQL
Big data	Hadoop clusters	Teradata
Nuvem	AWS S3 Buckets Azure Blobs e Table Google Workspace (Gmail e Gdrive)	Azure Table Office 365 (Exchange, SharePoint, & OneDrive) SalesForce

Tipos de arquivos suportados

Bancos de dados	Access Dbase	SQLite MSSQL MDF e LDF	Microsoft Office	v5 6 95 97 2000	XP 2003 em diante Arquivos Office: Word, Excel, PowerPoint. Access, Outlook, Other(.pub e .xps)
Imagens	BMP FAX GIF JPG	PDF (integrado) PNG TIF	Código aberto	Star Office /Open Office/Libre Office	
Compactados	Bzip2 Gzip (todos os tipos)	TAR ZIP (todos os tipos)	Padrões abertos	PDF RTF HTML	XML CSV TXT
Microsoft Backup Archive	Microsoft Binary/ KF				

Tipos de arquivos suportados

APA	Australia Privacy Amendment	HIPAA	Health Insurance Portability and Accountability Act
APPI	Act on Protection of Personal Information	KVKK	Lei turca de proteção de dados pessoais
CCPA	California Consumer Privacy Act	LGPD	Lei Geral de Proteção de Dados (Brasil)
GDPR	Finanças	NYDFS	New York State Department of Financial Services
GDPR	General Data Protection Regulation	PCI DSS	Payment Card Industry DataSecurity Standard
GDPR	Saúde	SHIELD	Privacy Shield Framework
GDPR	Identidade do sistema nacional de saúde	UK-GDPR	General Data Protection Regulation (UK)
GDPR	Dados pessoais		

A cobertura de dados estruturados e não estruturados fornecida pela DDC é atualizada frequentemente com novos recursos. Confira as especificações mais recentes aqui.

Os principais perfis de classificação estão integrados e alinhados com as principais leis e regulamentações de dados, tanto regionais quanto globais. É possível fazer alterações nesses perfis ou adicionar perfis totalmente novos, se necessário. Vários atalhos são fornecidos em termos de edição e cópia para que você possa criar e testar rapidamente seus próprios perfis. Há total flexibilidade na especificação de vários parâmetros, incluindo níveis de sensibilidade e infotipos.

Tipos de arquivos suportados

Nenhum	Público	Interno
Privado	Restrito	

Categorização de dados confidenciais

A DDC já vem com vários infotipos, mais de 250, abrangendo a grande maioria das leis e normas regionais e globais de privacidade de dados. Quando um modelo de perfil de classificação já existente é selecionado, o subconjunto apropriado de infotipos é adicionado automaticamente. Você pode modificar ou ampliar essa lista criando seus próprios infotipos personalizados, permitindo que sejam definidas regras que descrevam precisamente como o mecanismo de varredura deve procurar as cadeias de dados em questão. O acesso total aos grupos e categorias está disponível como parte da definição, de modo que as correspondências de dados possam ser exibidas nas seções apropriadas dos resultados da varredura.

Categorias de infotipos (já no produto)

Finanças	Cartões de crédito/débito	Informações de conta bancária
Dados pessoais	E-mails Credenciais de login Número do cartão Etnia Número da placa Número de registro Número do passaporte Data de nascimento Endereço MAC	Lista de endereços Número do telefone Sexo Religião Endereço IP Número do celular Nome Profanidade
Saúde	Informações de saúde do paciente	
Identidade do sistema nacional de saúde	Identificação pessoal	
Segredos	Descoberta de segredos	

Uso de agente para descoberta

A análise detalhada dos dados é realizada por agentes de descoberta que executam a varredura e informam os resultados ao DDC Connector para análise e processamento. Existem dois tipos de agentes: local e proxy. Normalmente, você instalaria agentes locais em locais de armazenamento de dados (para os quais você tem os direitos de acesso apropriados) para garantir que os dados nunca saiam do servidor por motivos de segurança ou desempenho. Por outro lado, os agentes proxy são usados para armazenamentos de dados em rede ou remotos, nos quais um servidor proxy separado hospeda o agente.

Tipo de agente	Prós	Contras
Local	Varredura mais rápida Os dados permanecem locais Não é preciso credencial	Tempo de implementação mais longo Gerenciamento mais complexo Suporta apenas armazenamentos de dados locais
Proxy	Implementação e dimensionamento mais rápidos Capacidade de fazer varredura de múltiplos armazenamentos de dados Suporte a vários tipos de armazenamentos de dados Nenhum recurso consumido no host de destino	Dados enviados pela rede para o agente antes da varredura Aumento da carga da rede e da superfície de dados O ideal é que estejam localizados na mesma LAN virtual (VLAN)

Benefícios comuns de ambos os tipos de agentes:



Um **número ilimitado** de agentes disponíveis para implementação sem custo extra, pois o licenciamento é baseado no volume de dados novos analisados



É possível executar várias subvarreduras para reduzir o tempo necessário para a varredura geral



A criptografia TLS está disponível para todas as comunicações entre armazenamentos de dados e agentes para proteger dados confidenciais contra escutas



Os dados de origem não são afetados, pois os agentes acessam apenas uma cópia temporária dos dados na memória interna durante a varredura



Os dados confidenciais não são armazenados durante o processo, pois é enviado apenas um resumo dos resultados da varredura para o DDC Connector



As varreduras **continuam** e registram resultados mesmo quando a conexão com o DDC Connector for cortada

Com seus agentes instalados, você pode começar a pensar sobre o que e como deseja fazer a varredura. Ofecemos uma grande flexibilidade na forma como as varreduras são configuradas, gerenciadas e operadas. Os parâmetros importantes configuráveis para cada varredura de forma independente incluem:

- Nome (até 64 caracteres)
- Descrição (até 250 caracteres)
- Um ou mais armazenamentos de dados (onde fazer a varredura)
- Um ou mais perfis de classificação (o que procurar)
- Operação manual ou programada (quando ativar)

Após a conclusão de uma determinada varredura, ela é adicionada a uma lista de todas as varreduras que foram realizadas. Após, utiliza-se a ferramenta de relatório para visualizar os resultados detalhados da varredura. Dependendo do que for encontrado, você poderá fazer alterações na configuração da varredura e executá-la novamente, o que é vantajoso ao trabalhar em diferentes armazenamentos de dados ou em novos tipos de normas de conformidade pela primeira vez.

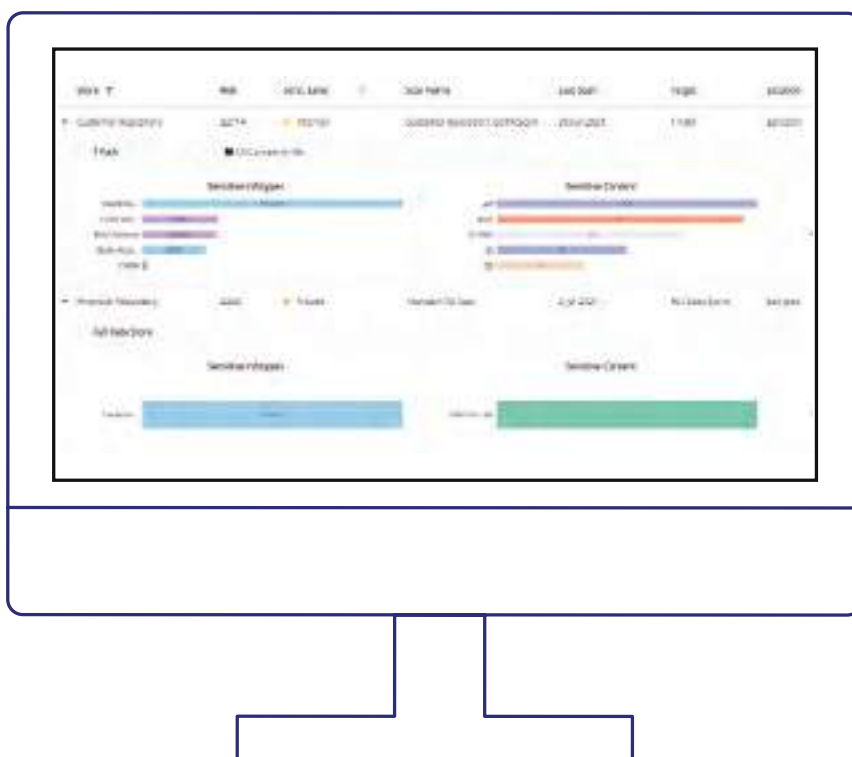
Análise de resultados da varredura

Os resultados de varreduras anteriores e novas estão disponíveis para uso com a ferramenta de relatório integrada. São oferecidas três principais exibições para relatórios de varredura:

- **Varreduras**
- **Armazenamentos de dados**
- **Objeto de dados**

Antes que as informações detalhadas estejam disponíveis, é gerado um relatório que agrega os resultados de uma ou mais varreduras de sucesso. Muitas vezes, por motivos logísticos, de segurança ou de desempenho, você pode querer usar várias varreduras (cada uma procurando diferentes tipos de dados regulamentados) para analisar todo o seu espaço de dados. Com nossa abordagem flexível, você ainda pode fazer isso e ajustar cada varredura conforme necessário, usando várias iterações, antes de gerar um relatório final que possa representar o status atual de todos os seus dados. Também é fácil entender sua postura organizacional do ponto de vista de um único regulamento.

Cada guia do módulo de relatório oferece uma visão de aspectos ligeiramente diferentes dos seus dados - todas, porém, têm o objetivo comum de ajudar a identificar rapidamente quaisquer dados confidenciais em risco.



Visualização de varreduras

Fornece uma visão de varredura agregada de alto nível dos infotipos encontrados (divididos por porcentagem em cada categoria presente), juntamente com três outros gráficos que representam a presença de objetos de dados confidenciais em termos de distribuição de categorias, tipos de conteúdo e tipos de arquivo em questão.



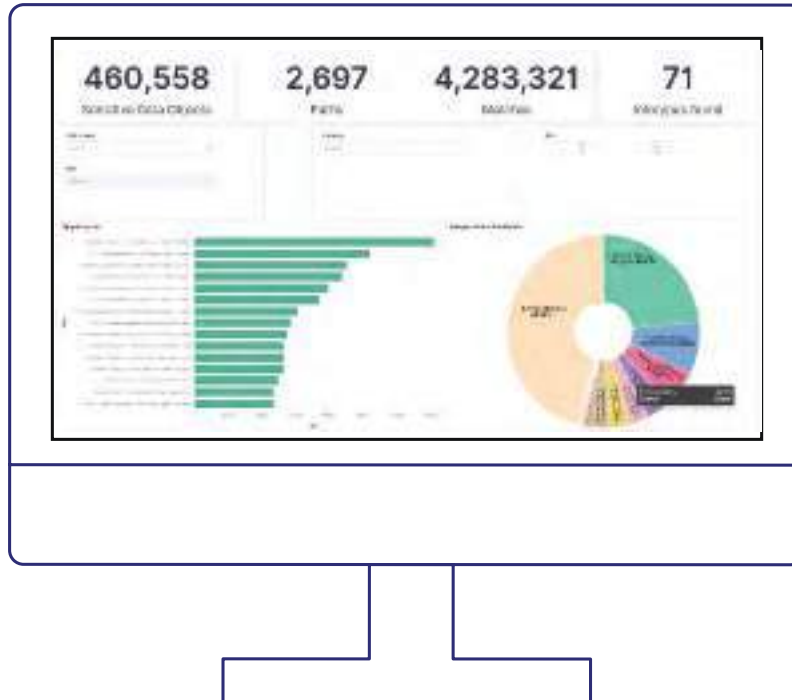
Visualização dos armazenamentos de dados

Fornece uma lista dos armazenamentos de dados com detalhes que incluem risco, nível de confidencialidade, infotipos encontrados e número de objetos de dados confidenciais presentes em cada armazenamento de dados. Isso proporciona um meio rápido de avaliar em alto nível se os dados confidenciais estão ou não em risco e quais armazenamentos de dados estão envolvidos.



Visualização de objeto de dados

Fornece uma lista detalhada de todos os objetos de dados verificados que contêm dados confidenciais classificados em ordem decrescente de risco de exposição. Para cada objeto de dados confidenciais em questão, é possível ver onde ele está localizado, o armazenamento de dados no qual ele reside e os nomes individuais e as correspondências de dados para todos os infotipos associados encontrados. As informações sobre o status da correção também estão presentes, o que proporciona uma visão mais profunda dos seus dados confidenciais, confirmando se uma política de criptografia está ativa ou não. Nenhum dado confidencial é armazenado como parte do processo de relatório.



Exportação de dados

Além de visualizar objetos usando a ferramenta de relatório integrada, você também pode exportar dados em um formato NDJSON para análise por uma ferramenta de relatório externa, como a da Elastic usada nas demonstrações da Thales.

Motivos da nossa escolha de tipo de arquitetura

Plataforma única e integrada

Decidimos fazer da DDC uma parte integrante da Plataforma CipherTrust Data Security da Thales CipherTrust porque, em nossa experiência, a descoberta de dados é a base de qualquer estratégia eficaz de segurança de dados. Essa abordagem de plataforma integrada permite que o CipherTrust Manager atue como o console de gerenciamento central para vários conectores, incluindo descoberta de dados, criptografia transparente e tokenização. Se já estiver usando o CTE, você estará familiarizado com o console e se beneficiará da consistência na implementação de tarefas comuns de gerenciamento, como grupos de usuários, políticas de dados, controle de acesso e login.

Correspondência precisa de padrões

A Thales realizou uma extensa pesquisa para determinar a melhor tecnologia de descoberta de dados do mercado. Após muitas pesquisas e testes de bancada, a equipe percebeu uma diferença significativa ao usar a GLASS Technology™ em vez da Regex. A GLASS é uma tecnologia proprietária líder de Ground Labs (um parceiro tecnológico da Thales) para definir e combinar padrões simples e complexos de dados em escala, em oposição a uma linguagem de código de desenvolvedor não criada especificamente para esse fim. Ela foi projetada desde o início com o objetivo expresso de encontrar padrões de dados em conjuntos de dados modernos em cenários de armazenamentos de dados estruturados e não estruturados.

Descoberta de alto desempenho

Um objetivo importante era fornecer um mecanismo de descoberta altamente eficiente e preciso. A maioria das soluções de descoberta de dados disponíveis atualmente usa a linguagem de expressão regular (Regex) (originada na década de 1950) para pesquisar padrões dentro de sequências de texto. Essa abordagem requer amplo conhecimento do desenvolvedor para ser implementada e a sintaxe não se baseia na gramática inglesa. O maior inimigo do Regex é o desempenho, à medida que a simultaneidade e a complexidade dos padrões aumentam, mais lento é o desempenho obtido. O padrão Regex tem limites para os padrões que pode corresponder com precisão, o que o torna longe de ser ideal no mundo moderno, onde há centenas de regulamentações de privacidade diferentes e uma ampla gama de problemas de gerenciamento de dados a serem abordados.

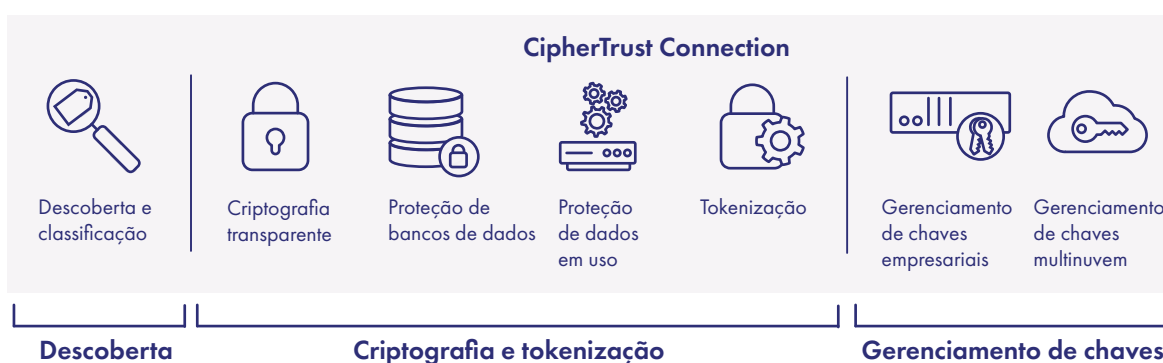
Recursos fáceis de personalização

O suporte à personalização foi outro critério importante para complementar os modelos abrangentes incluídos. Para atender às necessidades proprietárias, os padrões de dados GLASS podem ser escritos por não desenvolvedores usando uma gramática no estilo inglês para expressar como um dado ou padrão existe. A análise detalhada mostra que ele superou todas as limitações importantes do Regex para técnicas de correspondência de padrões, especialmente sua capacidade de operar em várias plataformas e ambientes de nuvem, além de ser muito econômico em termos de sobrecarga da CPU. Isso fez do GLASS um componente central ideal para ser incorporado à DDC, funcionando de forma eficiente e transparente como parte da plataforma CipherTrust na identificação e proteção de dados críticos.

Modelo de licenciamento flexível

Um modelo de consumo baseado no total de dados analisados é usado para licenciar a DDC, proporcionando um alinhamento próximo com muitos modelos de assinatura de nuvem e oferecendo um número ilimitado de agentes de descoberta sem incorrer em taxas adicionais. Isso facilita muito o acesso de todas as suas equipes aos recursos de descoberta de dados, sendo que a única decisão real a ser tomada é a capacidade geral de descoberta de dados necessária para a sua empresa. Uma nova varredura de dados antigos não é subtraída da sua franquia de uso de dados restante, apenas os novos dados analisados são contabilizados.

Plataforma CipherTrust Data Security da Thales



Segurança em primeiro lugar

Como uma empresa de segurança líder de mercado, é óbvio que criamos e implementamos soluções seguras. A DDC não é exceção, com grande ênfase em garantir que nenhum dado confidencial seja armazenado ou fique vulnerável à exposição.

- Garante que todos os dados que fluem entre seus armazenamentos de dados e agentes sejam protegidos usando criptografia robusta
- Fornece controles de acesso granulares para que você possa adaptar a solução às suas necessidades exatas

Integração com outras soluções

Plataforma CipherTrust DataSecurity

A DDC foi otimizada para ser usada como parte da plataforma CipherTrust Data Security, um conjunto integrado de produtos e soluções de segurança centrados em dados que unificam a descoberta, proteção e controle de dados em uma única plataforma. Consequentemente, você pode atender a todas as necessidades de proteção de dados usando uma única plataforma de um único provedor, neste caso a Thales. Isso lhe oferece um conjunto completo de recursos de proteção de dados, incluindo gerenciamento de chaves, controle de acesso do usuário, criptografia no nível do arquivo, criptografia na camada de aplicativos, criptografia de banco de dados, criptografia com preservação de formato, tokenização e mascaramento de dados.

CipherTrust Manager

Todos os componentes utilizam o CipherTrust Manager como console de gerenciamento, oferecendo gerenciamento central de chaves de criptografia, controle de acesso granular e configuração de políticas de segurança. Uma das principais vantagens é que nossa solução foi projetada para proteger seus dados onde quer que eles residam, no local ou na nuvem, e qualquer que seja o mecanismo de armazenamento em uso, como arquivos, bancos de dados, big data ou containers. Por fim, você acabará exigindo menos recursos dedicados às operações de segurança de dados, cumprirá suas obrigações de conformidade com mais confiança e reduzirá o risco comercial para a sua empresa.

Também é possível implementar a DDC com outras ferramentas de proteção de dados de terceiros, como as que oferecem suporte à criptografia ou tokenização de dados. Nossa solução não é intrusiva e complementa as ferramentas que você já possui, em vez de substituí-las. No entanto, o CipherTrust Manager ainda é necessário, pois fornece o único método para iniciar o recurso de descoberta de dados e

gerenciar as licenças necessárias para sua operação, juntamente com as chaves necessárias para criptografar dados confidenciais.

Alguns dos principais motivos pelos quais os clientes da Thales têm usado a DDC junto com produtos de terceiros incluem:

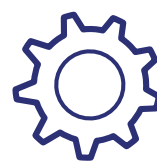
Muitas vezes nos perguntam... **"Qual é a diferença entre a CipherTrust Data Discovery and Classification e outras soluções de prevenção contra perda de dados (DLP)?"**



Descoberta



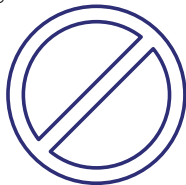
Proteção



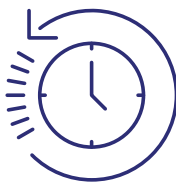
Controle

As soluções de DLP se concentram em impedir que dados confidenciais saiam do perímetro da empresa. A DDC se concentra na privacidade de dados, identificando dados confidenciais e obtendo uma compreensão clara dos dados e seus riscos. Isso permite as medidas adequadas para proteger seus dados e o cumprimento das normas de privacidade e segurança de dados.

- Nossa solução corrige erros na cobertura, já que a ferramenta de descoberta existente não pode fazer a varredura de todos os dados



Bloqueie o acesso desnecessário a dados para reduzir o risco de violações de dados



Mantenha-se atualizado com a evolução do panorama regulamentar para permanecer em



Aborda as preocupações com a transformação digital sobre segurança na nuvem para ajudar com objetivos empresariais e tecnológicos

estruturados e não estruturados

- Busca maior precisão nas correspondências de dados para eliminar as limitações que estão enfrentando com o padrão Regex
- Suporte a uma gama mais ampla de sistemas operacionais e plataformas para questionar mais a pegada de dados

Principais conclusões

Plataforma CipherTrust Data Security

Sua estratégia de gerenciamento e proteção de dados pode ser melhorada significativamente com a solução CipherTrust Data Discovery and Classification. Junto com o CipherTrust Manager e o CipherTrust Transparent Encryption, ela proporciona visibilidade e percepção consideráveis de seus dados confidenciais, oferecendo opções de proteção rápidas e eficazes. A solução permite:

***Experimente nossa avaliação gratuita de 90 dias para ver por si mesmo como nossa solução pode ajudar sua empresa agora.**

Abreviações

ACL	Lista de controles de acesso
CDSP	Plataforma CipherTrust Data Security
CIFS	Common Internet File System
CM	CipherTrust Manager
CTE	CipherTrust Transparent Encryption
DB	Banco de dados
DDC	CipherTrust Data Discovery and Classification
GUI	Interface gráfica do usuário
NAS	Armazenamento conectado à rede
REST API	Aplicativo de transferência de estado representacional Interface de programação
TDP	Plataforma de dados da Thales
NAS	Armazenamento conectado à rede

Glossário

Perfil de classificação	Um perfil de classificação usa uma lista de infotipos para definir o tipo de informação confidencial a ser pesquisada durante uma varredura
Conector	Um conector é um termo genérico relacionado aos diferentes produtos ou componentes licenciáveis (que incluem DDC e CTE) que são gerenciados usando o console CipherTrust Manager
Agente CTE	Um agente CTE é um componente de software usado para criptografar dados associados a GuardPoints definidos usando políticas CTE
Correspondência de dados	Uma correspondência de dados ocorre durante uma varredura quando é encontrada uma instância qualificada de um infotipo incluído no escopo da pesquisa
Objeto de dados	Um arquivo ou tabela de banco de dados localizado em um armazenamento de dados é conhecido como objeto de dados
Armazenamento de dados	Um armazenamento de dados é a entidade onde os dados são realmente armazenados, com a DDC oferecendo suporte a vários tipos: local, rede, banco de dados, big data e nuvem
Agente DDC	Um agente DDC é um componente de software usado para verificar um armazenamento de dados em busca de tipos específicos de dados definidos por meio de infotipos associados
Descoberta	perfil de classificação
Falsos negativos	A descoberta de dados é o processo de mapeamento dos ativos de dados de uma organização, incluindo seus locais, tipos e níveis de confidencialidade
Falsos positivos	Os falsos negativos ocorrem quando o processo de descoberta não consegue identificar uma ou mais instâncias válidas de dados confidenciais
GuardPoint	Um GuardPoint especifica a lista de pastas ou caminhos a serem protegidos, o acesso a arquivos e a criptografia de arquivos no GuardPoint são controlados por políticas de segurança
Infotipo	Um infotipo é usado para categorizar dados específicos (como números de passaporte ou e-mail) a serem procurados durante uma varredura de descoberta, formando um componente integral na definição de um perfil de classificação
Política	Uma política é um conjunto de regras que regem o acesso e a criptografia de dados
Remediação	Remediação se refere ao processo de proteção (ou segurança) dos dados que foram identificados como vulneráveis como resultado de uma varredura, os métodos típicos de correção incluem criptografia, tokenização, mascaramento de dados e controle de acesso.
Conjunto de recursos	Um conjunto de recursos se refere aos arquivos ou diretórios aos quais a política se aplicará, juntamente com as regras principais associadas
Risco	Um risco é a presença de um objeto de dados confidenciais em um armazenamento de dados e está diretamente relacionado às correspondências de dados encontradas no objeto de dados ou no armazenamento de dados
Varredura	Uma varredura faz parte do processo de descoberta usado para pesquisar dados confidenciais em armazenamentos de dados usando critérios definidos em perfis de classificação

Objeto de dados confidenciais	Um objeto de dados que contém qualquer correspondência de dados é conhecido como um objeto de dados confidenciais
Nível de confidencialidade	Nível de confidencialidade é um parâmetro obrigatório (ao definir armazenamentos de dados e perfis de classificação) relacionado ao grau de vulnerabilidade potencial de um determinado objeto de dados, se exposto
Dados estruturados	Os dados estruturados são altamente organizados e facilmente compreendidos pela linguagem de máquina, exemplos incluem nomes, datas, endereços, números de cartão de crédito, informações de estoque, geolocalização e muito mais
Tag	Uma tag ajuda a agrupar dados e pode ser especificada ao criar armazenamentos de dados e perfis de classificação
Plataforma de dados da Thales	A plataforma de dados da Thales (TDP) é uma plataforma de big data baseada na tecnologia Hadoop que é usada pela DDC para várias tarefas, incluindo o armazenamento de resultados de varreduras
Dados não estruturados	Dados não estruturados são informações que não têm um modelo de dados (ou esquema) predefinido ou não são organizados de maneira predefinida, o que os torna inadequados para armazenamento em um banco de dados relacionado



Entre em contato conosco

Para saber as localizações de todos os escritórios e obter informações de contato, acesse cpl.thalesgroup.com/contact-us

cpl.thalesgroup.com



Thales CipherTrust Data Discovery and Classification



The crucial first step in privacy and data protection regulatory compliance is to understand what constitutes sensitive data, where it is stored, and how it is used. If you don't know what sensitive data you have, where it is, and why you have it, you cannot apply effective policies and controls to protect it.

Thales CipherTrust Data Discovery and Classification enables you to efficiently locate structured and unstructured regulated data across the cloud, big data, and traditional data stores in your enterprise. You get a complete understanding of your sensitive data and its risks, so you can make better decisions about closing your gaps, prioritizing remediation, securing your cloud transformation and third party data sharing.

Unlike alternative disjointed solutions that can leave data exposed or compromised, Thales CipherTrust Data Discovery and Classification provides a streamlined workflow all the way from policy configuration, discovery, and classification, to risk analysis and reporting. This eliminates security blind spots and complexities. As a result, you can easily uncover and mitigate your data privacy risks, enforce data sovereignty, and proactively respond to a growing number of data privacy and security regulations, including GDPR, CCPA, LGPD, PCI DSS and HIPAA.

Thales CipherTrust Data Discovery and Classification is available for sale to the U.S. Federal Government exclusively through Thales Trusted Cyber Technologies.

Effective for enterprise-wide data privacy

Thales CipherTrust Data Discovery and Classification is an enterprise-wide data privacy solution that is simple to deploy and scale. It provides a streamlined workflow to help you quickly discover your regulated data across traditional and cloud repositories.

Streamline data discovery and automate data protection

You can rely on Thales CipherTrust Intelligent Remediation to discovery, classify and use policy-based data encryption to automate data remediation and reduce business risk. It enables organizations to streamline processes, gain complete visibility to identify compliance gaps, automatically mitigate risks, and comply with data privacy and security regulations.

Quick start with built-in templates and flexibility for custom policies

Thales CipherTrust Data Discovery and Classification provides a comprehensive set of built-in classification templates for commonly requested data privacy and security regulations, such as GDPR, CCPA, etc., but is flexible enough to handle custom policies based on specific patterns, algorithms, and more.



Demonstrate compliance

Thales CipherTrust Data Discovery and Classification provides detailed reports that can help your auditors demonstrate compliance with various regulatory and business laws. Efficient scans enable you to build a strong foundation for overall data privacy and security.

Use Cases

- **Finding and classifying data:** Scans both structured and unstructured data using categorization techniques to find similar data more efficiently.
- **Enhancing security posture:** Delivers a detailed report of what sensitive data you have, where it resides and associated risk to help you make better decisions about what data needs protection.
- **Migrating data to the cloud:** Presents a detailed risk analysis prior to moving data to the cloud in order to qualify decisions based on solid facts.
- **Managing large data volume increases:** Maintains visibility of your data during rapid rises in storage and usage with a robust and scalable solution.
- **Ongoing compliance:** Keeps you up to date automatically with all major global and regional compliance requirements, while reducing the risk of failed audits or fines by proactively identifying compliance gaps.

Technical Specifications

Data Stores

- Local storage and local memory on the host
- Network storage
 - Windows Share (CIS/SMB)
 - Unix File System (NFS)
- Cloud
 - Office 365 Exchange Online
 - Office 365 SharePoint Online
 - Azure Blobs and Table
 - AWS S3
- Databases
 - IBM DB2
 - Oracle
 - SQL
 - PostgreSQL
 - SAP Hana
 - Teradata
 - MySQL
 - Mongo
- Big Data
 - Hadoop Clusters

Type of files supported

- Databases: Access, DBase, SQLite, MSSQL MDF & LDF
- Images: BMP, FAX, GIF, JPG, PDF (embedded), PNG, TIF
- Compressed: bzip2, Gzip (all types), TAR, Zip (all types)
- Microsoft Backup Archive: Microsoft Binary / BKF
- Microsoft Office: v5, 6, 95, 97, 2000, XP, 2003 onwards
- Open Source: Star Office / Open Office
- Open Standards: PDF, HTML, CSV, TXT

Type of data identified

- Health (US Health Insurance Claim number, etc.)
- Financial (American Express, Diners Club, Mastercard, VISA card numbers, bank account number, etc.)
- Personal (name, last name, address, DOB, email, etc.)
- National ID (social security number, etc.)
- Custom Information Types

Pre-built templates

The solution includes a wide range of ready-to-use templates that can help you meet common regulatory and business policy needs. For example:

- CCPA
- GDPR
- HIPAA
- PCI DSS
- LGPD
- NYDFS

Minimum RAM required

- 16GB.

Network Connection

- At least 1GB

CipherTrust Data Security Platform

CipherTrust Data Discovery and Classification is part of the CipherTrust Data Security Platform. The CipherTrust Platform unifies data discovery, classification, data protection, and provides unprecedented granular access controls, all with centralized key management. This simplifies data security operations, accelerates time to compliance, secures cloud migrations, and reduces risk across your business. You can rely on Thales CipherTrust Data Security Platform to help you discover, protect and control your organization's sensitive data, wherever it resides.

About Thales Trusted Cyber Technologies

Thales Trusted Cyber Technologies, a business area of Thales Defense & Security, Inc., protects the most vital data from the core to the cloud to the field. We serve as a trusted, U.S. based source for cyber security solutions for the U.S. Federal Government. Our solutions enable agencies to deploy a holistic data protection ecosystem where data and cryptographic keys are secured and managed, and access and distribution are controlled.

For more information, visit www.thalestct.com

Build a strong foundation for data privacy and security with CipherTrust Data Discovery and Classification



Data privacy in the spotlight

Today, the volume of enterprise data is exploding across industries. We talk about petabytes, and even exabytes, of data strewn across data centers, file shares, databases, cloud storage and backups. However, few organizations have the necessary visibility into their sensitive data to ensure its protection and to comply with regulatory requirements.

Most organizations simply do not know what sensitive data they have, where it is, how it flows, who can access it, and its risk of exposure. In organizations that are complex or have siloes of data, the lack of clear visibility into sensitive data can make it very difficult to apply effective policies and controls to protect this most critical asset.

Third-party risk is also a big concern. Vendors, partners, service providers, and others you share data with all add to your organization's risk not only for a data breach or a failed audit, but also for collateral damage to your customers' trust, your brand value, and your partner ecosystem.

Recently, high-profile data breaches have put data privacy in the spotlight. From lost business and reputational damage to devastating financial and legal consequences, the impact of a data breach can haunt companies for years.

However, that is just a part of the story. Stringent data privacy laws continue to be enacted around the world. This ever-evolving regulatory landscape leaves companies to interpret and comply with a growing patchwork of state, global, and industry-specific data privacy and security laws, creating significant risks for compliance. Clearly, companies need a simpler way to comply with regulatory requirements.

Data privacy versus data security

Data privacy and data security share a common goal to protect sensitive data, but they are not the same.

Data privacy focuses on the use and governance of personal data. It defines the policies to ensure that consumers' personal information is collected, shared, and processed in appropriate ways. It is about who has authorized access, and who can define or control it.

Data security focuses on protecting sensitive data from unauthorized access, malicious attacks, or misuse. A variety of tools, encryption techniques, and identity and access controls can be used to enforce policies and secure data.

For example, your organization may have deployed multiple layers of defense—encryption, key management, and access controls—to secure the sensitive data of your customers. But, if that data was collected, used, or shared with third parties without your customer's consent, you may be violating data privacy laws, even if the data is secured.

Thales CipherTrust Data Discovery and Classification

Thales CipherTrust Data Discovery and Classification enables your organization to get complete visibility of sensitive data with efficient data discovery, classification, and risk analysis across cloud, big data, and traditional environments. Thales Data Discovery and Classification is available for sale to the U.S. Federal Government exclusively through Thales Trusted Cyber Technologies.

The rich visualizations and insights provides a clear view of what sensitive data you have, where it resides, and its risk of exposure.

You can easily uncover and mitigate privacy risks, enforce data sovereignty, and comply with a growing number of data privacy and security regulations, including GDPR, CCPA, LGPD, PCI DSS and HIPAA.

Helps you uncover and mitigate data privacy risks

Thales CipherTrust Data Discovery and Classification streamlines the workflow from policy configuration, discovery, and classification, to risk analysis and reporting, enabling you to proactively mitigate risks and respond to evolving regulatory challenges.

1. Policies: Define your data privacy policies, data stores, classification profiles, and scans.
2. Discovery: Locate both structured and unstructured sensitive data across the entire enterprise in multi-cloud, big data, relational databases, or file storage systems.
3. Classification: Classify sensitive data, such as national IDs, financial data, and personal data, based on built-in templates or market-proven classification techniques.
4. Risk analysis: Understand your data and its risks with rich visualizations and risk scores.



5. Remediation: Remediate the risks with market leading data encryption and tokenization products and solutions available on the Thales CipherTrust Data Security Platform. CipherTrust Intelligent Remediation improves operational efficiency by applying the most appropriate protection immediately after sensitive data has been discovered.
6. Detailed reports Leverage charts and reports for risk analysis, status, and alerts throughout the data lifecycle.

Why Thales CipherTrust Data Discovery and Classification

Effective solution for enterprise-wide data privacy

- Efficiently locates structured and unstructured sensitive data across cloud, big data, and traditional data stores.
- Provides a comprehensive set of built-in templates for rapid discovery of your regulated or sensitive data.
- Available in agentless and/or agent-based deployment modes for reduced complexity and easy scalability.
- Offers a streamlined workflow that eliminates security blind spots and allows you to enforce unified policies.
- Supports a wide variety of datatypes, including custom Information Types.

Delivering complete visibility and control

- A centralized console with rich visualizations offers a clear view of sensitive data, allowing your organization to make informed decisions about cloud migration, data sharing, and risk remediation.
- Risk scores enable your security teams to uncover and close privacy gaps and prioritize remediation efforts.

Demonstrate compliance

- Detailed reports help your auditors demonstrate compliance to various regulations.
- Efficient scans enable you to build a strong foundation for your enterprise-wide data privacy and security.

CipherTrust Data Security Platform

CipherTrust Data Discovery and Classification is part of the CipherTrust Data Security Platform. The CipherTrust Platform unifies data discovery, classification, and data protection and provides unprecedented granular access controls, all with centralized key management. This simplifies data security operations, accelerates time to compliance, secures cloud migrations, and reduces risk across your business. You can rely on Thales CipherTrust Data Security Platform to help you discover, protect, and control your organization's sensitive data, wherever it resides.

About Thales Trusted Cyber Technologies

Thales Trusted Cyber Technologies, a business area of Thales Defense & Security, Inc., protects the most vital data from the core to the cloud to the field. We serve as a trusted, U.S. based source for cyber security solutions for the U.S. Federal Government. Our solutions enable agencies to deploy a holistic data protection ecosystem where data and cryptographic keys are secured and managed, and access and distribution are controlled. For more information, visit thalestct.com

Evaluation Guide : CipherTrust Data Discovery and Classification

Bringing agility and confidence to your data management



Thank you for agreeing to take part in an evaluation of the Thales [CipherTrust Data Discovery and Classification](#) (DDC) solution.

We are confident that it will help improve visibility on all the data in your organization, wherever it resides, enabling you to take the appropriate actions to protect sensitive or vulnerable assets with greater confidence.



About CipherTrust Data Discovery and Classification

Out of the box, CipherTrust Data Discovery and Classification (DDC) enables you to uncover the sensitive data you have (relating to all major privacy laws and regulations) and all the places it is stored, on-premises or in the cloud. Flexible customization capabilities enable you to discover and classify other sensitive assets, secrets and intellectual property, specific to you. Such comprehensive insight helps you make critical business decisions including precisely what data can be safely migrated to the cloud or shared with third parties for example.

The visual reports created as part of the discovery and classification analysis of the data locations you select are key to helping your team make better decisions about what specific data needs to be better protected – a strong factor in minimizing disruption in the event of a future data breach. Guidance is provided relating to data that can be eliminated, which if otherwise left to grow out of control could significantly increase both risk and liability for your organization.

Unlike the manual approach to searching for data, the use of our specialist solution eliminates the assumptions about what sensitive data may or may not exist – it provides a high degree of confidence that you are building a solid data management strategy with no critical data left unprotected.

Insights you can expect from your evaluation

The solution focuses on delivering rapid and insightful analysis of your data. You can configure scans to look for specific types of data in locations of your choice and then utilize the reporting module capabilities to establish what sensitive data you have, where it resides, its risk of exposure and any compliance gaps that may exist. You are then in prime position to take the necessary corrective actions.

The screen shots below provide a quick look at the main aspects you will be using.



Type of sensitive data stored (financial, national IDs, personal, medical)



Sensitive data location

Data Objects						
Q Export Data Objects						
Object Name	Risk	Type	Path	Owner	Info/Tags	
% Employee-Data.doc	10000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Health-Data.doc	10000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Employee-Data.pdf	10000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Employee-Data.xls	1100	File	2\Company\HR\Employee	Employee-Registry C&I	0	
% Data-Data.doc	1100	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% HR-Data.doc	1100	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	
% Data-Data.doc	000	File	2\Company\HR\Data documents	Employee-Registry C&I	0	

Risk assessment

Object Name

Risk

Type

Path

Owner

Info/Tags

2

Employee-Data.doc

0

File

2\Company\HR\Data documents

Employee-Registry C&I

0

4 Info/Tags

21 Metadata

View

14

Information

12

Remediation Information

Emergency - Inside GuardPoint

Emergency-Response-Info.doc

Policy?

Emergency-Response.doc

28 April 2021

Emergency-Response-Response.doc

PCI DSS - Payment Card Industry Data Security

PCI-DSS-Response-Response.doc

Security Rules

View Rules

Order

Response Set

Error Set

Response Set

Action

Flow

Warning

1

2020

2020

2020

all logs

Emergency-Response

Yes

2

2020

2020

2020

all logs

Emergency-Response

Yes

3

2020

2020

2020

all logs

Emergency-Response

Yes

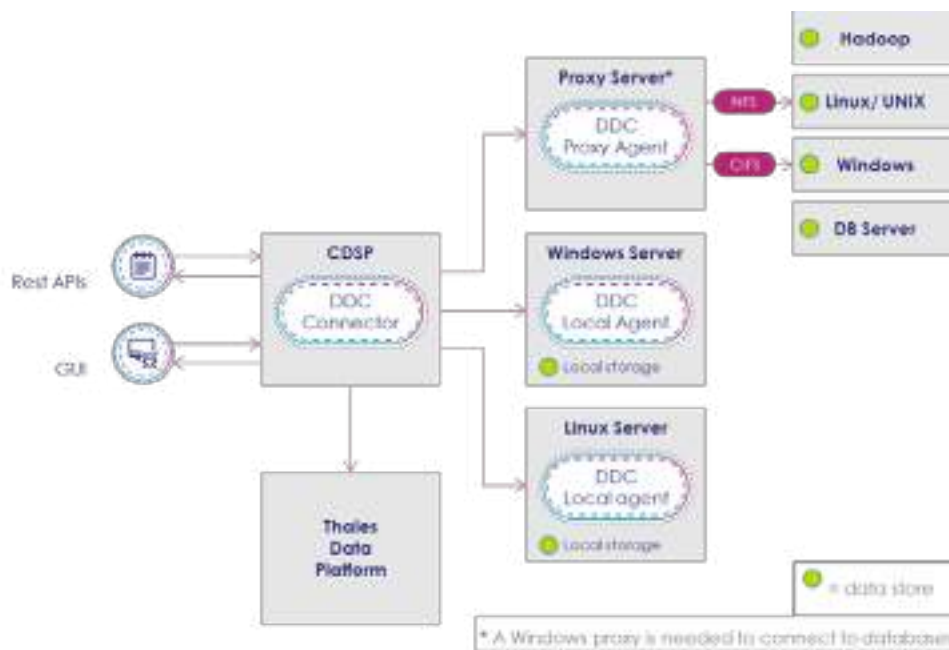
A quick overview

As part of the CipherTrust Data Security Platform (CDSP), CipherTrust Data Discovery and Classification (DDC) allows you to search data repositories for sensitive data elements such as credit card numbers, tax IDs or personal information including names and email addresses. When the search is complete it can build reports that assess what data elements were present, their location within the data repositories and assign a risk assessment based upon the number and type of data elements found.

There are 3 main components of the solution.

- [CipherTrust Manager](#) (CM) – Management node where you use a browser interface to access this node to perform setup, define searches and view reports
- Thales Data Platform (TDP) – Thales-built Hadoop node which acts as the data repository of the solution for configuration and scan data
- Discovery Agent – The DDC agent which performs the data scan

The diagram below illustrates a typical DDC deployment. The agent component can be deployed directly on the server to be scanned or remotely on a proxy server. An agent deployed as a proxy can scan any supported databases or shared file systems. In contrast, a local agent must be deployed directly on servers to scan files that are only available on local file systems.



Typical DDC deployment

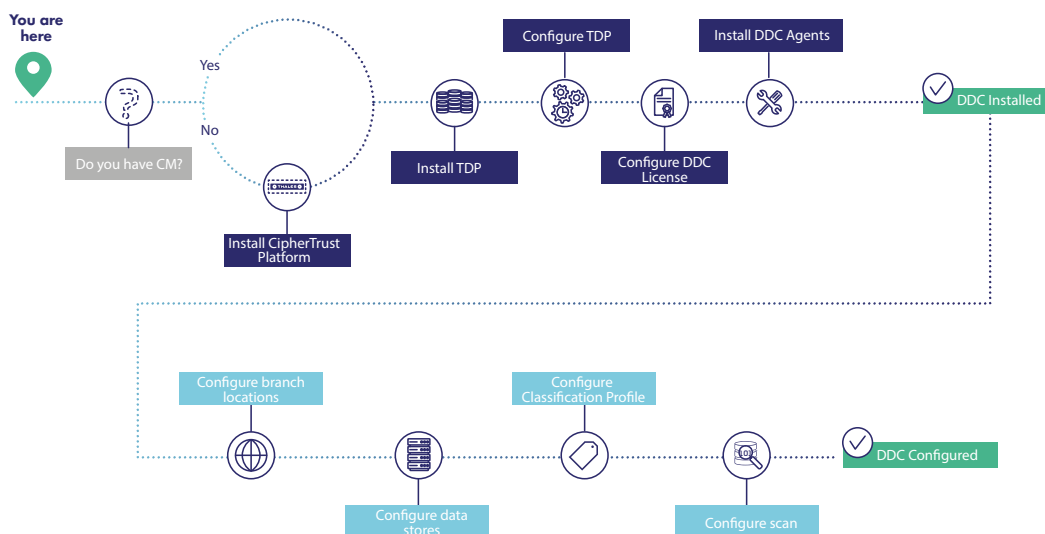
Steps to consider before deployment

There are some preparations you need to make before installing and using DDC. The table below provides guidance on ensuring you have the best possible environment in place to run your DDC evaluation.

Step	Action	Details
1.	Ensure you have suitable servers and network access	At certain stages of the installation you will need to install discovery agents – make sure you have details of the server names, their IP addresses to use and that the appropriate access rights are available to the team involved. The credentials required for all the locations to scan should have “read only” access rights.
2.	Decide which team members will participate in the evaluation	You may wish to check data belonging to different teams – make sure you have identified the critical people involved and secured their availability in advance.
3.	Clarify the locations and types of data stores to be part of your scans	Our recommendation would be to have clarity on the data store category, type and version in order to be able to properly select the agent for scanning. Specific information and supplementary guidance is provided in the online documentation. It is very important to select the correct agent and define the proper data store in order to be able to provide accurate scan results.
4.	Define the type(s) of data you wish to find	Ask the data owners well in advance, what types of data they wish to find during the evaluation – everything linked to data privacy laws are covered using pre-defined templates, but other data types proprietary to your organization will require extra preparation time (and subsequent use of the customization capabilities of the solution).
5.	Identify any data that is already encrypted (possibly using another vendor solution)	Will DDC be able to gain access to that data during the scan to perform some meaningful analysis? Please check with your Thales Account Manager or authorized reseller, if CipherTrust Manager is able to manage the keys for any third party product in question being used for the encryption.
6.	Decide when and how often you will activate the scans	Although the scanning process does not have any significant impact on the performance of your systems, you should start thinking about when you will run the scans and how many different types of scan you intend to perform during the evaluation period. Our recommendation is to start with simple scans over a small data footprint to gain experience of how long it takes in your environment - then you can add more complex search parameters and run more frequent scans. The targets, classification profiles, filters and frequency of the scan can be edited after a scan has been defined.
7.	Consider what action(s) you will take when you uncover sensitive data that is not protected	A main benefit of the evaluation is in qualifying whether or not you have sensitive data that is not currently protected. This may come as a surprise or you may have suspected such data would be found somewhere throughout your network of servers. If you are unfortunate enough to find regulated data that is not secure then you need to be prepared to take rapid corrective action to reduce your risk. Making use of other CipherTrust Data Security Platform connectors such as CipherTrust Transparent Encryption or CipherTrust Tokenization may prove to be an astute move.

Getting started

All of the documentation you need is available [online](#). To help you navigate through the important pieces, below you will find an outline of the minimal set of installation and configuration steps you need to complete before you can start defining the locations you wish to scan for sensitive data.



Pre-requisites:

- 1) Make sure you fulfill all the [requirements](#) for having DDC in place before you start with the installation steps

Installation steps:

- 1) If you are new to the CipherTrust platform or wish to install a separate version of CipherTrust Manager to use with your DDC evaluation, then you need to [install Virtual CipherTrust Manager](#)
- 2) If you are an existing CipherTrust Data Security Platform customer, then you will already have CipherTrust Manager installed and in use - **please note that currently DDC is designed to work with Virtual CipherTrust Manager rather than the physical appliance**
- 3) Once CipherTrust Manager is installed you can then install DDC in a specific sequence as follows:
 - a. [Install TDP](#)
 - b. [Configure TDP](#)
 - c. [Configure the DDC license in CipherTrust Manager](#)
- 4) [Install DDC agents](#)

When installation of CM, TDP and DDC is complete, you can then launch DDC from within the CM console and begin configuring the various items needed to create and run a scan.

Configurations steps:

- 1) [Configure branch locations](#) - a branch location specifies a site where the file servers, databases, and data centers that contain data to be scanned are physically located
- 2) [Configure data stores](#) - a data store is the entity where the data is actually stored, with DDC supporting various types - local, network, database, Big Data and cloud
- 3) [Configure classification profiles](#) using pre-defined infotypes where possible - an infotype is used to categorize specific data (such as passport numbers or email addresses) to look for during a discovery scan, forming an integral component in the definition of a classification profile
- 4) If necessary create one or more [custom infotypes](#) to meet your specific needs
- 5) [Configure scan](#) - a scan is part of the discovery process that is used to search for sensitive data within data stores using criteria defined in classification profiles

Suggested use case to explore – local storage protection

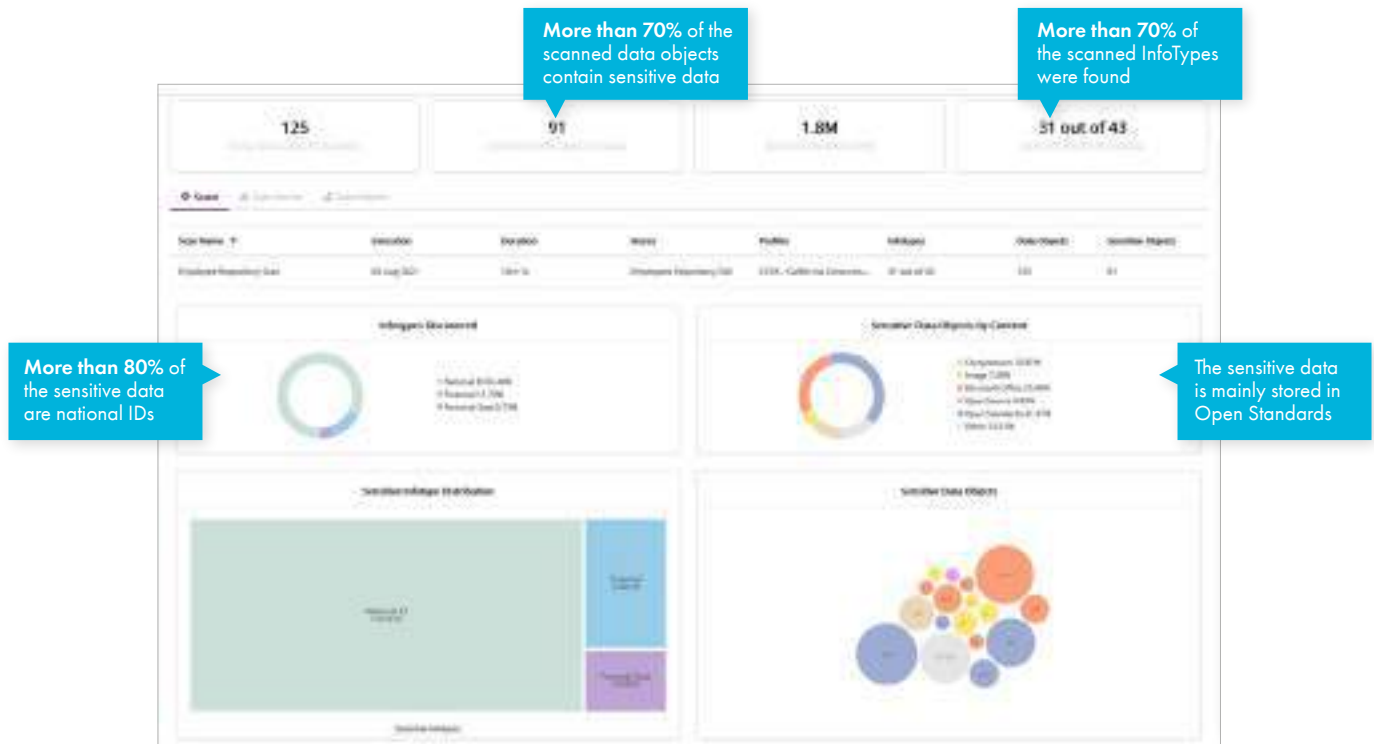
Following the steps below will help you plan your local storage protection. The links take you to the relevant section in our online documentation portal, Thales Docs.

- 1) [Set up your local data store in DDC](#)
- 2) [Configure a scan](#) defining what type of data you want to look for using one of the pre-defined classification profiles (GDPR, PCI DSS, CCPA, etc.)
- 3) [Run the scan](#)
- 4) [Generate the report](#) for the scan you have just run - the report will show locations and associated risk levels associated with data objects in the various data stores
- 5) [Analyze the results](#) locating the paths that contains sensitive data without protection and define your actions over those locations

Analyzing the results

The screen shots below provide an overview of how to interpret the results from the scans. We are using some examples from the Thales test environment to illustrate the power of the scanning and reporting capabilities of DDC.

In this first example, we are scanning a local storage called "Employee Repository Old" to search for CCPA data.



In this second example, we are grouping the information that we got after scanning a customer repository searching for GDPR data and a financial repository (Oracle DB) searching for PCI data.



In the following example, we are displaying a Data Object that we encrypted automatically using DDC after we discovered the file contained sensitive data (credit card numbers).

I can demonstrate the file is encrypted as it contains credit card numbers

Visibility into the different types of data stored in files without compromising the data security

If I have a breach, I can see if the file was encrypted before the incident

In the following example, we are showing a Data Object that contains sensitive data (email addresses, MC address, financial information, among others) and it cannot be protected as no CTE agent is available in that location.

Visibility into the riskiest file and its contents to assist with the security plan accordingly

The following report has been generated based on a NDJSON file created by DDC and using ELK for analysis and display.



Your feedback

Thank you for completing the evaluation of our CipherTrust Data Discovery and Classification solution. We trust you found it useful and look forward to hearing your feedback. We should be grateful if you would answer as many of the following questions as you can.

Question	Your response
How did DDC fit into your overall data security strategy?	
What is the main use case of interest for deployment of DDC?	
What were the highlights or benefits that you saw?	
What information was most useful in the dashboards (e.g. statistics, insights, incident report or something else)?	
What were the drawbacks of the solution for you?	
What, if anything, seemed confusing to you?	
What missing features would you like to see?	
How would that help you?	
Any other feedback you wish to share?	

You can use the following [link](#) to take the survey or scan the QR code below to launch the survey on a mobile device.



Thanks in advance.

The CipherTrust team at Thales.

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.



Contact us

For all office locations and contact information,
please visit cpl.thalesgroup.com/contact-us

> cpl.thalesgroup.com <



Thales CipherTrust Data Discovery and Classification



The crucial first step in privacy and data protection regulatory compliance is to understand what constitutes sensitive data, where it is stored, and how it is used. If you don't know what sensitive data you have, where it is, and why you have it, you cannot apply effective policies and controls to protect it.

Thales CipherTrust Data Discovery and Classification enables you to efficiently locate structured and unstructured regulated data across the cloud, big data, and traditional data stores in your enterprise. You get a complete understanding of your sensitive data and its risks, so you can make better decisions about closing your gaps, prioritizing remediation, securing your cloud transformation and third party data sharing.

Unlike alternative disjointed solutions that can leave data exposed or compromised, Thales CipherTrust Data Discovery and Classification provides a streamlined workflow all the way from policy configuration, discovery, and classification, to risk analysis and reporting. This eliminates security blind spots and complexities. As a result, you can easily uncover and mitigate your data privacy risks, enforce data sovereignty, and proactively respond to a growing number of data privacy and security regulations, including GDPR, CCPA, LGPD, PCI DSS and HIPAA.

Effective for enterprise-wide data privacy

Thales CipherTrust Data Discovery and Classification is an enterprise-wide data privacy solution that is simple to deploy and scale. It provides a streamlined workflow to help you quickly discover your regulated data across traditional and cloud repositories.

Streamline data discovery and automate data protection

Thales has created an automatic workflow that links CipherTrust Data Discovery and Classification with CipherTrust Transparent Encryption - we call this CipherTrust Intelligent Protection. This new solution discovers, classifies and uses policy-based data encryption to automate data remediation and reduce business risk. It enables organizations to streamline processes, gain complete visibility to identify compliance gaps, automatically mitigate risks, and comply with data privacy and security regulations.

Quick start with built-in templates and flexibility for custom policies

Thales CipherTrust Data Discovery and Classification provides a comprehensive set of built-in classification templates for commonly requested data privacy and security regulations, such as CCPA, HIPAA, PCI DSS, LGPD, NYDFS and UK GDPR, but is flexible enough to handle custom policies based on specific patterns, algorithms, and more.



Demonstrate compliance

Thales CipherTrust Data Discovery and Classification provides detailed reports that can help you demonstrate to auditors compliance with various regulatory and business laws. Efficient scans enable you to build a strong foundation for overall data privacy and security.

Use Cases

- **Finding and classifying data:** Scans both structured and unstructured data using categorization techniques to find similar data more efficiently.
- **Enhancing security posture:** Delivers a detailed report of what sensitive data you have, where it resides and associated risk to help you make better decisions about what data needs protection.
- **Migrating data to the cloud:** Presents a detailed risk analysis prior to moving data to the cloud in order to qualify decisions based on solid facts.
- **Managing large data volume increases:** Maintains visibility of your data during rapid rises in storage and usage with a robust and scalable solution.
- **Ongoing compliance:** Keeps you up to date automatically with all major global and regional compliance requirements, while reducing the risk of failed audits or fines by proactively identifying compliance gaps.

Technical Specifications

Data Stores

- Local storage and local memory on the host
- Network storage
 - Windows Share (CIFS/SMB)
 - Unix File System (NFS)
- Cloud
 - Office 365 Exchange Online
 - Office 365 SharePoint Online
 - Azure Blobs and Table
 - AWS S3
 - Google Mail
 - Google Drive
 - Salesforce
- Databases
 - IBM DB2
 - Oracle
 - SQL
 - PostgreSQL
 - SAP Hana
 - Teradata
 - MySQL
 - Mongo
- Big Data
 - Hadoop Clusters
 - SharePoint OnPremises

Type of files supported

- Databases: Access, DBase, SQLite, MSSQL MDF & LDF
- Images: BMP, FAX, GIF, JPG, PDF (embedded), PNG, TIF
- Compressed: bzip2, Gzip (all types), TAR, Zip (all types)
- Microsoft Backup Archive: Microsoft Binary / BKF
- Microsoft Office: v5, 6, 95, 97, 2000, XP, 2003 onwards
- Open Source: Star Office / Open Office
- Open Standards: PDF, HTML, CSV, TXT

Type of data identified

- Health (Australian Medicare Card, European EHIC, US Health Insurance Claim number, etc.)
- Financial (American Express, Diners Club, Mastercard, VISA card numbers, bank account number, etc.)
- Personal (name, last name, address, DOB, email, etc.)
- National ID (social security number, Spanish DNI, etc.)
- Custom Information Types

Pre-built templates

The solution includes a wide range of ready-to-use templates that can help you meet common regulatory and business policy needs. For example:

- CCPA
- GDPR
- HIPAA
- PCI DSS
- LGPD
- NYDFS
- UK GDPR

Minimum RAM required

- 16GB

Network Connection

- At least 1GB

CipherTrust Data Security Platform

CipherTrust Data Discovery and Classification is part of the CipherTrust Data Security Platform. The CipherTrust Platform unifies data discovery, classification, data protection, and provides unprecedented granular access controls, all with centralized key management. This simplifies data security operations, accelerates time to compliance, secures cloud migrations, and reduces risk across your business. You can rely on Thales CipherTrust Data Security Platform to help you discover, protect and control your organization's sensitive data, wherever it resides.

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.



Thales Data Platform

Version: v3.1.6 (latest)

4 February 2026

Contents

- Get Started. 3
 - Hardening Guidelines 3
 - 1. Install Virtual Thales Data Platform 6
 - Amazon Web Services Deployment 6
 - Microsoft Azure Deployment 15
 - Private Cloud Deployment 26
 - 2. Changing the Initial Password 28
 - 3. Network Configuration 28
 - 4. Configuring Services After Deployment 31
 - 5. Post Install 40

Thales Data Platform (On-prem) Deployment

Thales Data Platform (TDP) (on-prem) is a Big Data platform based on Hadoop technology. It can be used for POCs and production in conjunction with DDC.

Prior to deploying or upgrading your TDP cluster, please check the [Hardening Guidelines](#) and the [Compatibility Matrix between CM and TDP](#).

To deploy TDP 3.1.6, please follow the instructions belows depending on your current state.

Quick Setup Wizard

If you wish to try out TDP prior to installing it in production, you can use a quick setup script which simplifies the installation and reduces the number of manual configuration steps.

You can download the TDP IQ Wizard ZIP file corresponding to the CipherTrust Manager version from the [Thales Customer Support Portal](#).

Deployment Environments

How the Thales Data Platform is distributed and deployed is based on the environment. The supported environments are listed below along with the supported distribution method:

Public Cloud Images

We support Virtual Thales Data Platform in the following environments.

- Amazon Web Services: SafeNet Cloud Provisioning System

Refer to: [AWS Deployment](#)

- Microsoft Azure: Available as a BYOL image in the Microsoft Azure Marketplace

Refer to: [Azure Deployment](#)

Private Cloud Images

- VMware vSphere: OVA image file

Refer to: [Private Cloud Deployment](#)

Hardening Guidelines

The Thales Data Platform should be deployed into as secure an environment as possible. Every effort has been made to include Hadoop components without known security vulnerabilities, however, additional precautions should be taken especially when the Thales Data Platform is deployed into an untrusted environment.

Securing Hadoop

Here are some general guidelines on securing your TDP deployment. These recommendations are general and apply to any Hadoop cluster.

- The Hadoop cluster should only be deployed inside a corporate or trusted network. You must use a VPC network if deploying to a cloud.
- None of the Hadoop nodes should be exposed to the Internet.
- To access services from the Internet, it is recommended to use VPN access. If VPN is not available, it is recommended to use jump servers or bastion hosts. For example, to access SSH (port 22), do not expose it directly to the Internet. Instead, use a jump server or a bastion host.
- To access web services, such as DDC or Ambari, use a reverse proxy or some kind of a load balancer. These devices normally sit in a DMZ (demilitarized zone) and forwards only a specific TCP port to the Hadoop cluster.
- For Hadoop clusters configure the replication channels to use transport level encryption.
- Make sure that you configure a firewall to forbid external access to any IP/port other than those of Knox (by default it is listening on port 8443) unless the source machine is another TDP node. The SSH and Ambari ports should only be accessible to trusted source IPs.

Additionally for the DDC-TDP configuration, below you can find some guidelines that will help you set secure credentials and secure TLC configuration. For details and instructions, refer to your Hadoop documentation.

▪ Secure credentials

- Hadoop credentials must be rotated regularly.
- Credentials must be different for every installation.
- Password must meet the security requirements described below:
 - At least 10 characters in length.
 - Contains both uppercase (A-Z) and lowercase (a-z) alphabetic characters.
 - Has at least one numeric (0-9) character.
 - Has at least one special character (! @ # \$ % ^ & * () _ - + = , . / < > ? ; ' : \ " [] \ \ { } | ~ `).
 - Does not contain spaces or tabs.

▪ Secure TLS configuration

- Disable SSL and TLS 1.0 and 1.1.
- Use secure renegotiation or disable renegotiation.
- Disable TLS compression.

-
- Use Authenticated Encryption cipher suites.
 - Use cipher suites with strong key exchange.
 - Do not use cipher suites with known vulnerabilities.
 - Cipher suites order must be defined by the user.
 - Use Perfect Forward Secrecy (PFS).
 - Download the tool from <https://testssl.sh/> to check the configuration.

References

Thales strongly recommends appropriately configuring and securing your Hadoop environment according to industry best practices and your organization's security policies.

The following references are not comprehensive, but are intended to provide a starting point for background, tools, and best practices that may be applied to your Hadoop environment:

- [HDP Security Overview](#)
- [Introduction to Hadoop Security](#)
- [Securing Hadoop: Security Recommendations for Hadoop \(Paper\)](#)
- [Hadoop Security \(O'Reilly Book\)](#)

System Administrative Key

The SSH Private Key, used to access the System Administrative account `root`, is extremely sensitive and should be kept in a secure environment.

Configuring TDP nodes

While deploying a Thales Data Platform (TDP) instance, it is recommended to use a five-node cluster: **Two master nodes** and **three worker nodes**.

- **Two master nodes:** Using two master nodes eliminates single point of failure. If you use only one master node and it fails for some reason, TDP remains down until the administrator recovers the master node.
 - **Three worker nodes:** This is the default setting/replication factor (meaning each file is copied to three different nodes). Three worker nodes are recommended because:
 - HDFS clusters are not backed up frequently and using three nodes ensures data resilience.
 - All data stored in the service can withstand the failure of up to 'replication factor - 1' nodes. For example, in a system with three nodes, it can tolerate the failure of two nodes (*no data loss occurs when two nodes fail*).
 - If you use only two worker nodes, one node failure will necessitate an immediate action, as losing the
-

second node will lead to data loss. Three worker nodes provide sufficient time for the administrators to provision a new node and replicate the data, a process that can take several days. Using two worker nodes increases data loss risk during this time frame.

Note

You may choose to reduce the number of master/worker nodes if you can address the above mentioned risks or have a reliable backup system capable of quickly recovering any TDP node and its data to meet the required SLA.

Tip

If you are using TDP for proof of concept (PoC) or on a trial basis, you have the option to use the suggested five-node cluster or a one-node cluster to reduce costs and speed up the evaluation process.

1. Install Virtual Thales Data Platform

Virtual Thales Data Platform instances are available in select private and public cloud environments.

Virtual Deployment Environments

Public Cloud Images

We support Virtual Thales Data Platform in the following environments.

- Amazon Web Services: SafeNet Cloud Provisioning System

Refer to: [AWS Deployment](#)

- Microsoft Azure: Available as a BYOL image in the Microsoft Azure Marketplace

Refer to: [Azure Deployment](#)

Private Cloud Images

- VMware vSphere: OVA image file

Refer to: [Private Cloud Deployment](#)

Amazon Web Services Deployment

You can deploy a Thales Data Platform image within Amazon Web Services (AWS).

Minimum Requirements

To deploy a Thales Data Platform instance, the following minimum requirements apply:

The minimum requirements for each node are:

- System volume: minimum of 200 GB
- Memory: 32 GB
- vCPUs: 8

We recommend a five-node cluster:

- Two nodes as namenodes, Knox gateway,
- Three nodes as datanodes and Node Manager.

See [configuring TDP node](#) for more details.

The configuration recommended for each node when a five-node cluster is build is the following:

Instance	vCPU	Mem (GiB)	Storage	CPU Credits/ hour	Network Performance
t2.2xlarge	8	32	EBS-Only	81	Moderate

Note

These minimum system requirements are for a system with light to moderate load. For applications that heavily load the system, additional memory and CPU allocation are required. The system volume holds all data as well as backups.

Provisioning the AMI

AWS region-specific and confidential Virtual Thales Data Platform images (AMI) are obtained by provisioning. Provisioning places a Thales Data Platform AMI in your target AWS account in the target region(s).

Use the Thales Customer Support Portal to request provisioning for a Thales Data Platform AMI.

Note

Only authorized Customer Support Portal users can make provisioning requests.

1. Login to the [Thales Customer Support Portal](#).
2. Navigate to the [My Provisioning Requests](#) page, through the **Tools > Provisioning Requests** menu.

3. Click **Create Provisioning Request**.
4. Select **Thales Data Platform** from the **Product Name** dropdown.
5. Select the desired **Product Version** from the dropdown.

Note

We recommend deploying the most recent version for the latest security fixes and most current features.

6. Select the desired **AWS Region** from the dropdown.
7. Provide your **AWS Account Number**.
8. (Optional) If desired, type in the names of **Additional Regions** in the text field, to deploy AMIs in more than one region.
9. (Optional) If desired, adjust the **Priority**. The default priority is **3-Medium**.
10. Click **Submit**.

A Thales client services engineer will receive and work on the request.

When your provisioning request is completed, you will an email notification with the subject line **Cloud Provisioning Request Complete**. Alternatively, you can periodically check the provisioning request status directly on the **My Provisioning Requests** page.

11. Retrieve the AMI ID from the completed provisioning request.
 - a. Navigate to the [My Provisioning Requests](#) page, through the **Tools > Provisioning Requests** menu.
 - b. Click on the name of the completed provisioning request. The **Status** column indicates requests that are completed.
12. Proceed to [deploy](#) the Virtual Thales Data Platform image.

Deploying in AWS

This section provides the steps for deploying a Virtual Thales Data Platform instance in AWS.

Prerequisites

- A Virtual Thales Data Platform AMI provisioned using the **Thales Customer Support Portal**, as described above in [Provisioning the AMI](#).

- If using a Windows client use PuTTY or similar utility to SSH to your Thales Data Platform instance as root.

If needed, use PuTTYgen or similar utility to format the SSH Key Pair.

- If using a Linux client use SSH to login as root.

To launch a Thales Data Platform instance in AWS

1. Sign in to your AWS account at the AWS portal at: <https://aws.amazon.com>
2. In the **Compute** group, select **EC2** service.
3. On the AWS top bar and to the right, make sure you select the AWS Region you provisioned for your AMI.



4. In the left pane, under the **Images**, select **AMIs** and search for the AMI (e.g. ami-04d5de93aaa02ceef) that was provisioned for you by the **SafeNet Cloud Provisioning System**.



5. Select your image and then click on **Launch instance from image**.
6. In the **Choose an Instance Type** screen select the desired **Instance Type**:
 - a. Select "t2.2xlarge", if you're installing a node that will be part of a 5 node cluster.
 - b. Select **Next: Configure Instance Details**.
7. In the **Configure Instance Details** screen,
 - a. Key in the number of instances required for the **Number of instances**. For example, 5 for a 5-node cluster.
 - b. Select **Next: Add Storage**.
8. In the **Add Storage** screen, select the desired **Size (GiB)**.
Select **Next: Add Tags**.
9. In the **Add Tags** screen, select **Add Tag**
 - a. Enter Tag: Name

-
- b. Value: Something meaningful to the operator; for example "Thales Data Platform for lab use"
 - c. Select **Next: Configure Security Group**.

10. In the **Configure Security Group** screen:

- a. Create a new security group or re-use a previously created group.
- b. Refer to [Security Group Rules](#). Consider adding all of the recommended rules.

Note

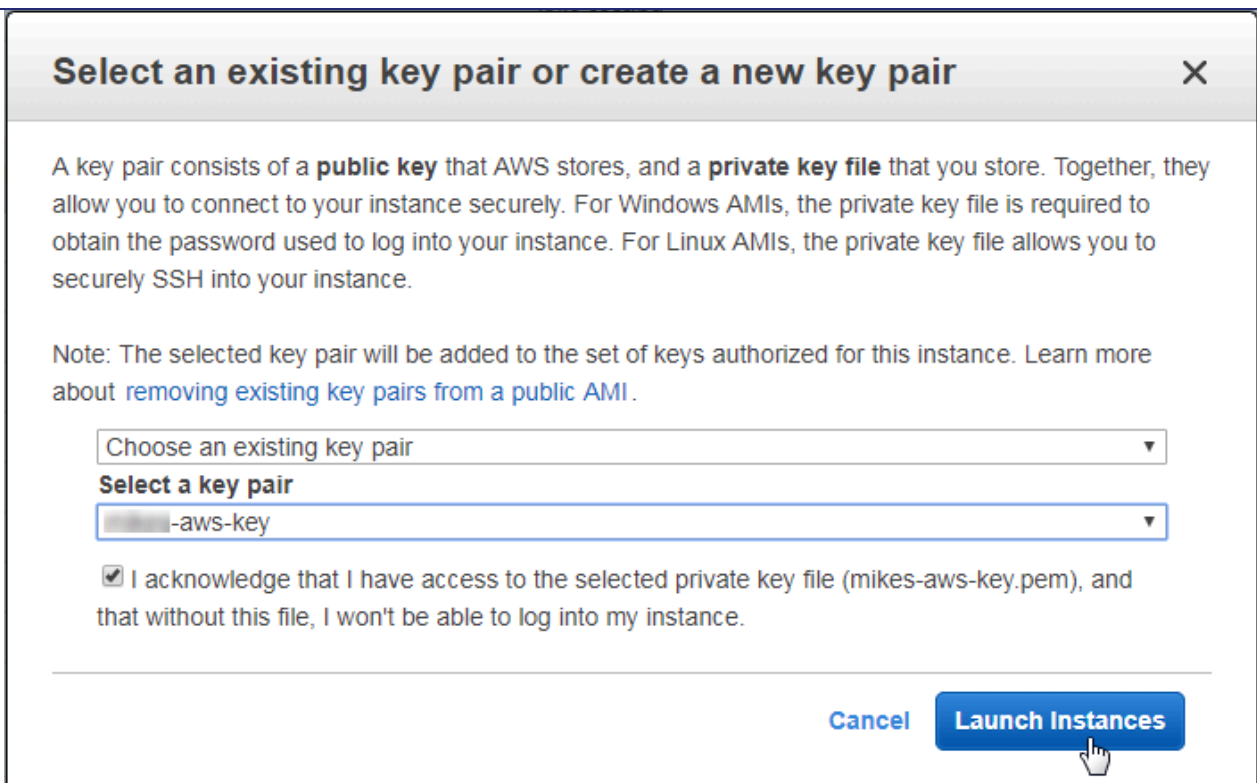
To launch the Thales Data Platform, you need at a minimum Port 22 inbound/outbound for TDP SSH connection and Port 443 for an HTTPS connection to the GUI.

- c. Select the **Review and Launch** screen.

11. In the **Review Instance Launch** screen:

- a. Review your launch configuration settings.
- b. Make any changes by selecting the **Edit ...** at the right of each section.
- c. After your review, select **Launch**.

You are presented with this dialog box to select a key pair.



Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair ▼

Select a key pair

-aws-key ▼

☒ I acknowledge that I have access to the selected private key file (mikes-aws-key.pem), and that without this file, I won't be able to log into my instance.

Cancel Launch Instances

- d. Select which **Key Pair** option to use. The key pair should be in Linux (PEM) format, not OpenSSH format. Valid options are:

- Choose an existing key pair from the drop-down list. If your key pair is not on this list, you can import it here:

Services > EC2 > Network and Security > Key Pairs

- Create a new key pair. (This option allows you to download your own key pair.)

- e. After selecting the **Key Pair**, check off the acknowledge statement at the bottom of the dialog box, confirming that you have access to the selected private key file.

Warning

It is important that you have access to the key pair you select, otherwise you will not have permissions to perform administrator operations like performing upgrades, advanced logging or an appliance reset.

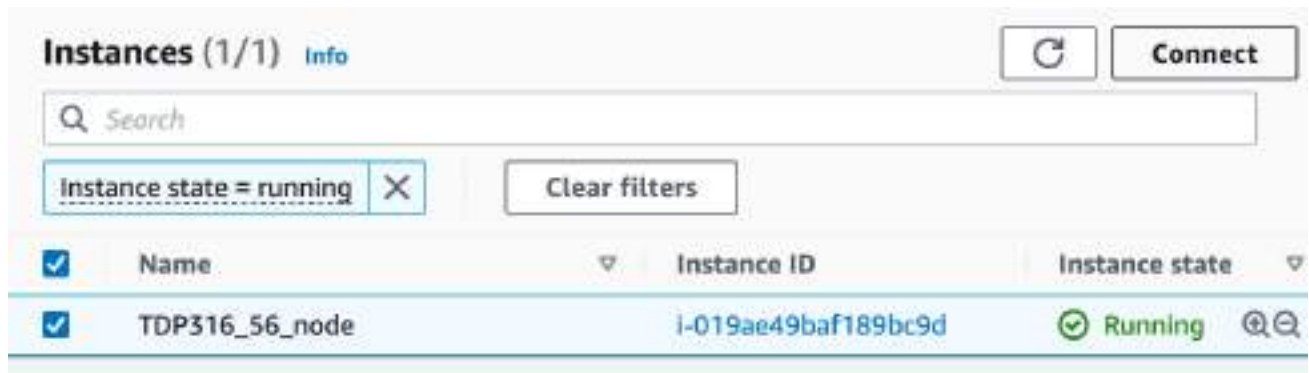
12. Select **Launch Instances**.

The **Launch Status** screen appears. You can review the launch process on this screen.

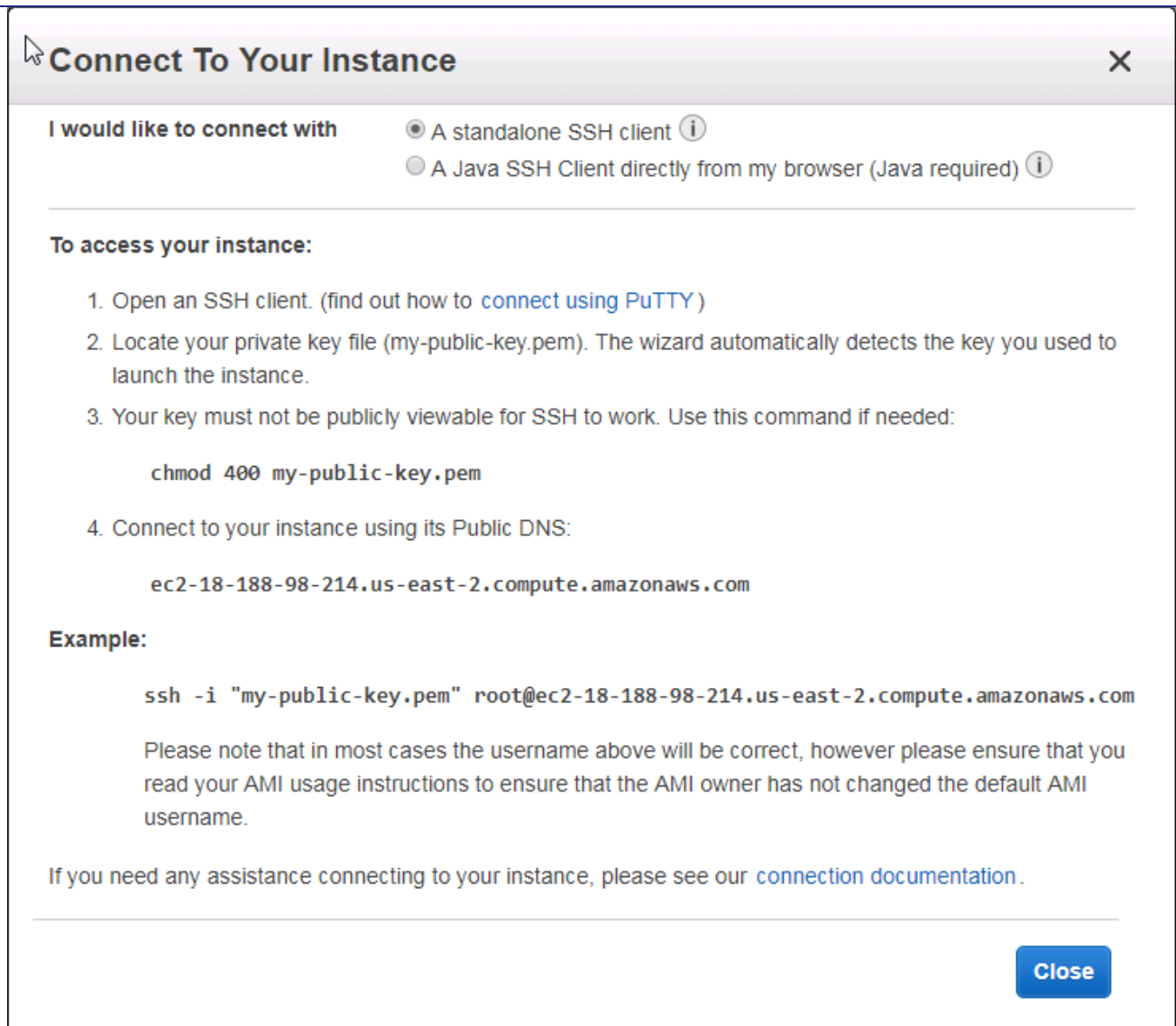
Note

It can take a several minutes for your instance to launch.

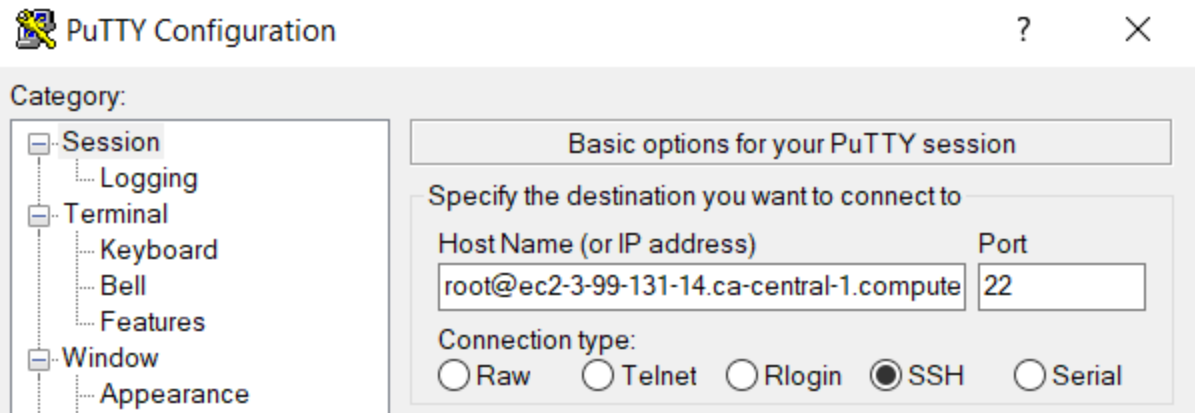
13. Select **View Instances** to view the state of your instance as it is being launched. You may need to search for the name of your instance in the Instances table if it is among many others.
14. Select your instance in the table and then select **Connect**.



The **Connect To Your Instance** dialog box appears:

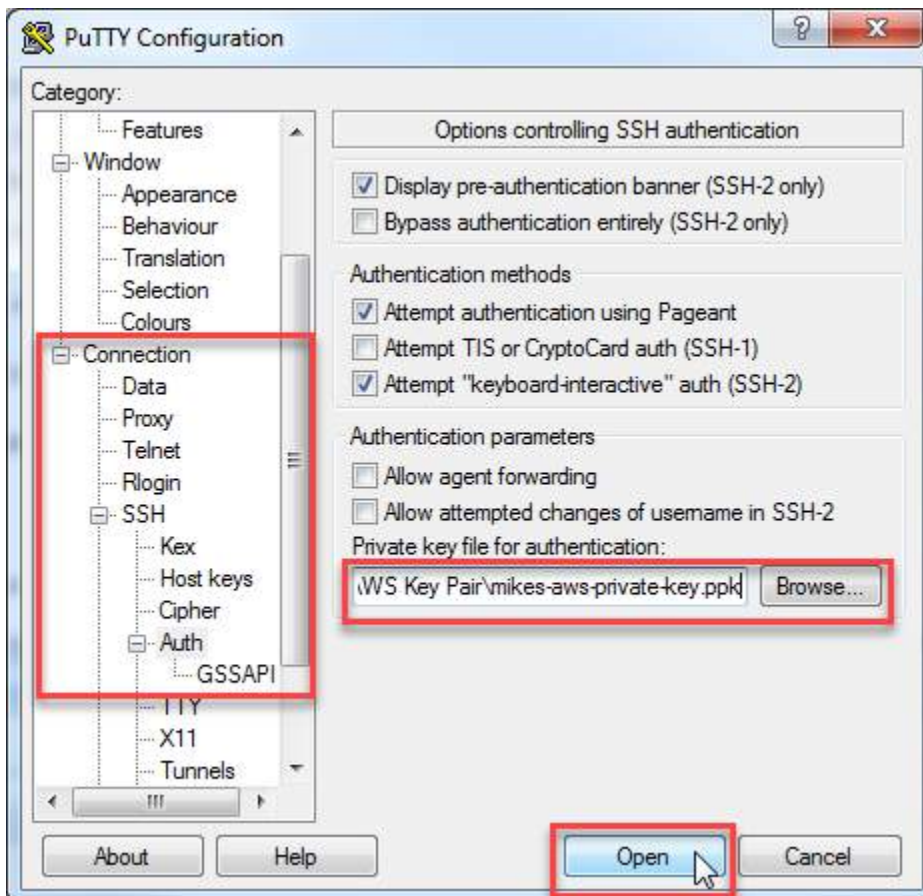


15. Copy the instance **Public DNS** string, for example: `ec2-54-89-148-184.compute-1.amazonaws.com`
16. If using PuTTY, begin a new session.
 - a. Paste the Public DNS in the **Host Name (or IP address)** field and preface this string with "root@".
 - b. Set the **Connection Type** to **SSH**.



17. In the **Category** pane:

- a. Select **Connection > SSH > Auth**.
- b. Select **Browse** and navigate to the folder with your Private Key.
- c. Select your Public Key file name. The file name and path is automatically places in the **Private key file for authentication** field.



- d. Select **Open**.

Note

If you are unable to connect to your instance using PuTTY, go to: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html> for further assistance.

Please follow instructions in [Changing the Initial Password](#) and all its following steps: [Network Configuration](#), [Configuring Services After Deployment](#), and [Post Install](#) to continue configuring your Thales Data Platform cluster.

Microsoft Azure Deployment

You can deploy a Thales Data Platform within the Microsoft Azure Cloud Computing Platform, including in the Azure Government environment (Arizona, Iowa, Virginia, and Texas regions).

Minimum Requirements

To deploy a Thales Data Platform instance, the following minimum requirements apply:

On TDP version **MISSING VARIABLE: version** the minimum requirements for each node are:

- System volume: minimum of 200 GB
- Memory: 28 GB
- vCPUs: 8

We recommend a 5-node cluster:

- 2 nodes as namenodes, Knox gateway,
- 3 nodes as datanodes and Node Manager.

See [configuring TDP node](#) for more details.

The config recommended for each node when a 5-node cluster:

- Standard_DS4 - 8 vcpus, 28 GiB memory

Note

These minimum system requirements are for a system with light to moderate load. For applications that heavily load the system, additional memory and CPU allocation are required. The system volume holds all data as well as backups.

Deploying in Azure

This section provides the steps for deploying a Virtual Thales Data Platform instance in Microsoft Azure. Refer to the Azure documentation for general information on launching a VM in Microsoft Azure.

Prerequisites

- If using a Windows client, use PuTTY or similar utility to SSH to your Thales Data Platform instance, using the username that you create in the following Step 11, for example, tdpuser.

If needed, use PuTTYgen or similar utility to format the SSH Key Pair.

- If using a Linux client, use SSH to your Thales Data Platform instance, using the username that you create in the following Step 11, for example, tdpuser.

To launch a Virtual Thales Data Platform instance

1. Sign in to the Azure or Azure Government portal.
2. Search for **Thales Data Platform** on the Marketplace page.
3. Select the **Thales Data Platform** image from the Virtual Machines group.

Note

The following steps apply to the Azure recommended 'Resources Manager' deployment model.

4. Click **Create**. The first screen of the **Create virtual machine** page is displayed.

Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Pay-As-You-Go
Resource group *	ddc-test
	Create new

Instance details

Virtual machine name *	tdp316n1
Region *	(US) West US
Availability options	No infrastructure redundancy required
Security type	Standard
Image *	Thales Data Platform 3.1.6 - Gen1
	See all images Configure VM generation
Azure Spot instance	☐
Size *	Standard_DS4 - 8 vcpus, 28 GiB memory (\$449.58/month)
	See all sizes

- Change the **Subscription** type if desired.
- Select an existing **Resource group** or enter the name for a new one.
- Specify the **Region** of an Azure Datacenter. For example, **West US**.

Note

For Azure government, this must be one of USGov Arizona, USGov Iowa, USGov Virginia, or USGov Texas.

- Enter a **Virtual machine name**, which is the hostname for the virtual machine your are creating, for example, "tdp316n1".

9. Select the **Size** for the VM that supports the [Minimum Requirements](#). **Standard_DS4** is recommended at this time.
10. Select the **SSH Public Key** for the **Authentication type**.

Note

SSH Public Key authentication must be used. **Password** authentication is not allowed when connecting as the initial user. We support OpenSSH format for the public key, and OpenSSH, PKCS1, or PKCS8 format for the private key.

Administrator account

Authentication type ☒ SSH public key ☐ Password

Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username *

SSH public key source

Key pair name *

[Review + create](#) [Previous](#) [Next : Disks](#)

11. For the **Username**, enter any name but not 'root', for example, tdpuser.
12. For the **SSH public key source**, select one of "Generate new key pair", "Use existing public key", "Use existing public key stored in Azure". We support OpenSSH format for the public key, and OpenSSH, PKCS1, or PKCS8 format for the private key.

Warning

It is important that you have access to the key pair you select, otherwise you will not have permissions to perform administrator operations like performing upgrades, advanced logging or an appliance reset.

13. In the **SSH Public Key** field, provide the necessary information for the SSH public key source selected in the last step.

14. Select **Next: Disks** >. The **Disks** screen is displayed.

Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#) ⓘ

Disk options

OS disk type * ⓘ

Standard SSD (locally-redundant storage) ▼

The selected VM size supports premium disks. We recommend Premium SSD for high IOPS workloads. Virtual machines with Premium SSD disks qualify for the 99.9% connectivity SLA.

Delete with VM ⓘ



Encryption at host ⓘ



❗ Encryption at host is not registered for the selected subscription. [Learn more about enabling this feature](#) ⓘ

Encryption type *

(Default) Encryption at-rest with a platform-managed key ▼

Enable Ultra Disk compatibility ⓘ



Ultra disk is not supported for the selected VM size Standard_DS4 in West US.

Data disks for tdp316n1

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

LUN	Name	Size (GiB)	Disk type	Host caching	Delete with VM ⓘ

[Create and attach a new disk](#)

[Attach an existing disk](#)

Advanced

Use managed disks ⓘ



Gallery package requires managed disks.

Ephemeral OS disk ⓘ



None



OS cache placement



Temp disk placement



The selected image is too large for the temp disk of the selected instance.

15. Change the **OS disk type** to "Standard HDD", unless you desire a faster disk.
16. Select an **Encryption type** depending on the [Azure-level disk encryption](#) you prefer.
17. Select **Next: Networking** >. The **Networking** screen is displayed.

Create a virtual machine

Basics Disks **Networking** Management Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.

[Learn more](#)

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network *	tdp316-56-masterVNET Create new
Subnet *	tdp316-56-masterSubnet (10.0.0.0/24) Manage subnet configuration
Public IP	(new) tdp316n1-ip Create new
NIC network security group	☐ None ☐ Basic ☒ Advanced This VM image has preconfigured NSG rules
Configure network security group *	(new) tdp316n1-rsg Create new
Delete public IP and NIC when VM is deleted	☐
Accelerated networking	☐ The selected image does not support accelerated networking.

Load balancing

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. [Learn more](#)

Load balancing options	☒ None ☐ Azure load balancer Supports all TCP/UDP network traffic, port-forwarding, and outbound flows. ☐ Application gateway Web traffic load balancer for HTTP/HTTPS with URL-based routing, SSL termination, session persistence, and web application firewall.

[Review + create](#)

< Previous

Next : Management >

The **Virtual Network**, **Subnet**, **Public IP** and **Configure network security group** fields are populated with default values for this VM, if you have preset networking values for other Azure VMs. Create new values if needed.

For a list of security groups/ports, refer to [Security Group Rules](#).

18. Select **Next: Management** >. The Management screen is displayed.
19. Select **Next: Advanced** >. The Advanced screen is displayed.
20. Select **Next: Tags** >. The **Tags** screen is displayed.

Create a virtual machine ...

Basics Disks Networking Management Advanced **Tags** Review + create

Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. [Learn more about tags.](#)

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

Name ⓘ	Value ⓘ	Resource
		12 selected ▼

Review + create

< Previous

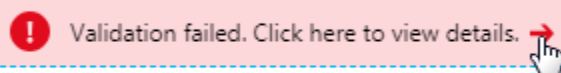
Next: Review + create >

21. Enter any desired tags.
22. Select **Next: Review + create** >. This is the final screen. Enter an email address, and click **Create** to launch the VM.

Azure will run an evaluation of your virtual machine creation configuration.

- a. If the validation was not successful, a **Validation failed** message is displayed:

Create a virtual machine



Basics Disks Networking Management Guest config **Tags** Review + create

Click on the arrow for details and precede to correct the cause of the validation error.

- b. If the validation is successful, a **Validation passed** screen is displayed, listing all VM details:

Create a virtual machine ...

 Validation passed

Basics Disks Networking Management Advanced Tags Review + create

PRODUCT DETAILS

Thales Data Platform by Thales Terms of use Privacy policy	Not covered by credits ⓘ 0.0000 USD/hr
Standard DS4 v2 by Microsoft Terms of use Privacy policy	Subscription credits apply ⓘ 0.5850 USD/hr Pricing for other VM sizes

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Basics

Subscription	Pay-As-You-Go
Resource group	TDP
Virtual machine name	tdp316n1
Region	East US
Availability options	No infrastructure redundancy required
Security type	Standard
Image	Thales Data Platform 3.1.6 - Gen1

23. Select **Create** to begin deployment of this VM. This screen is displayed indicating that deployment is in process.

CreateVm-gemalto-safenet.thales-data-platform-tha-20211202144244...

Deployment

Search (Ctrl+/)

Overview | Inputs | Outputs | Templates

Delete | Cancel | Redeploy | Refresh

We'd love your feedback! →

Deployment is in progress

Deployment name: CreateVm-gemalto-safenet.thales-data-pl...
 Subscription: Pay-As-You-Go
 Resource group: TDP

Start time: 12/2/2021, 3:13:04 PM
 Correlation ID: eedd993d-f0cb-49f3-8...

Deployment details (Download)

Resource	Type	Status	Operation details
ldp316n1	Microsoft.Compute/vir...	Created	Operation details
ldp316n141	Microsoft.Network/net...	Created	Operation details
ldp316n1-ip	Microsoft.Network/pu...	OK	Operation details
ldp316n1-nsg	Microsoft.Network/net...	OK	Operation details

When deployment completes, this screen is displayed, providing access to all resources supporting the new VM.

CreateVm-gemalto-safenet.thales-data-platform-tha-20211202144244...

Deployment

Search (Ctrl+/)

Overview | Inputs | Outputs | Templates

Delete | Cancel | Redeploy | Refresh

We'd love your feedback! →

Your deployment is complete

Deployment name: CreateVm-gemalto-safenet.thales-data-pl...
 Subscription: Pay-As-You-Go
 Resource group: TDP

Start time: 12/2/2021, 3:13:04 PM
 Correlation ID: eedd993d-f0cb-49f3-8...

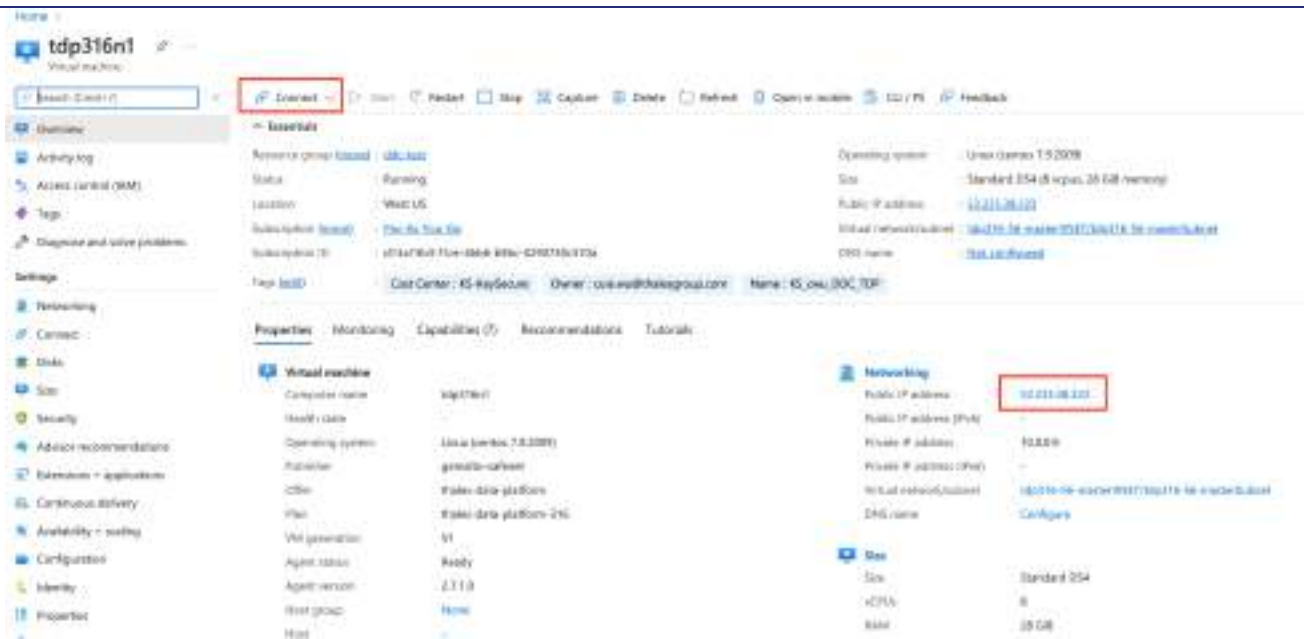
Deployment details (Download)

Next steps

- Setup auto-shutdown: Recommended
- Monitor VM health, performance and network dependencies: Recommended
- Run a script inside the virtual machine: Recommended

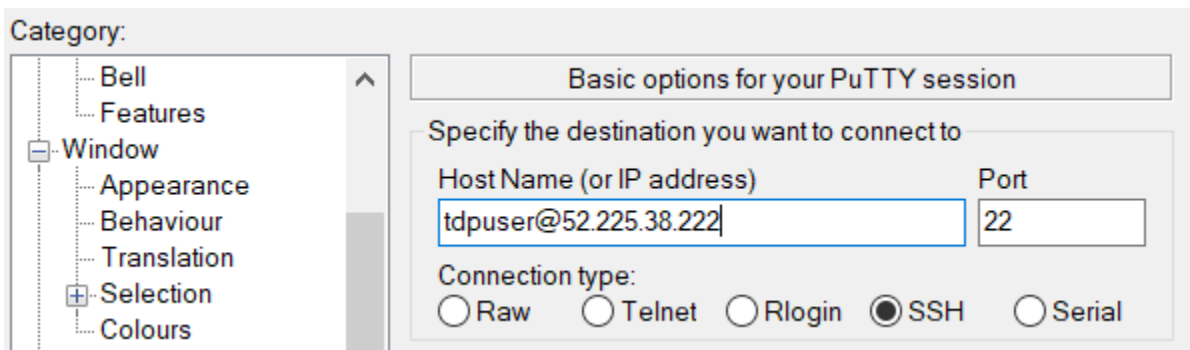
[Go to resource](#) [Create another VM](#)

- Click **Go to resource** then click **connect --> SSH**. Follow the same instructions as in [Connecting via SSH with client](#).

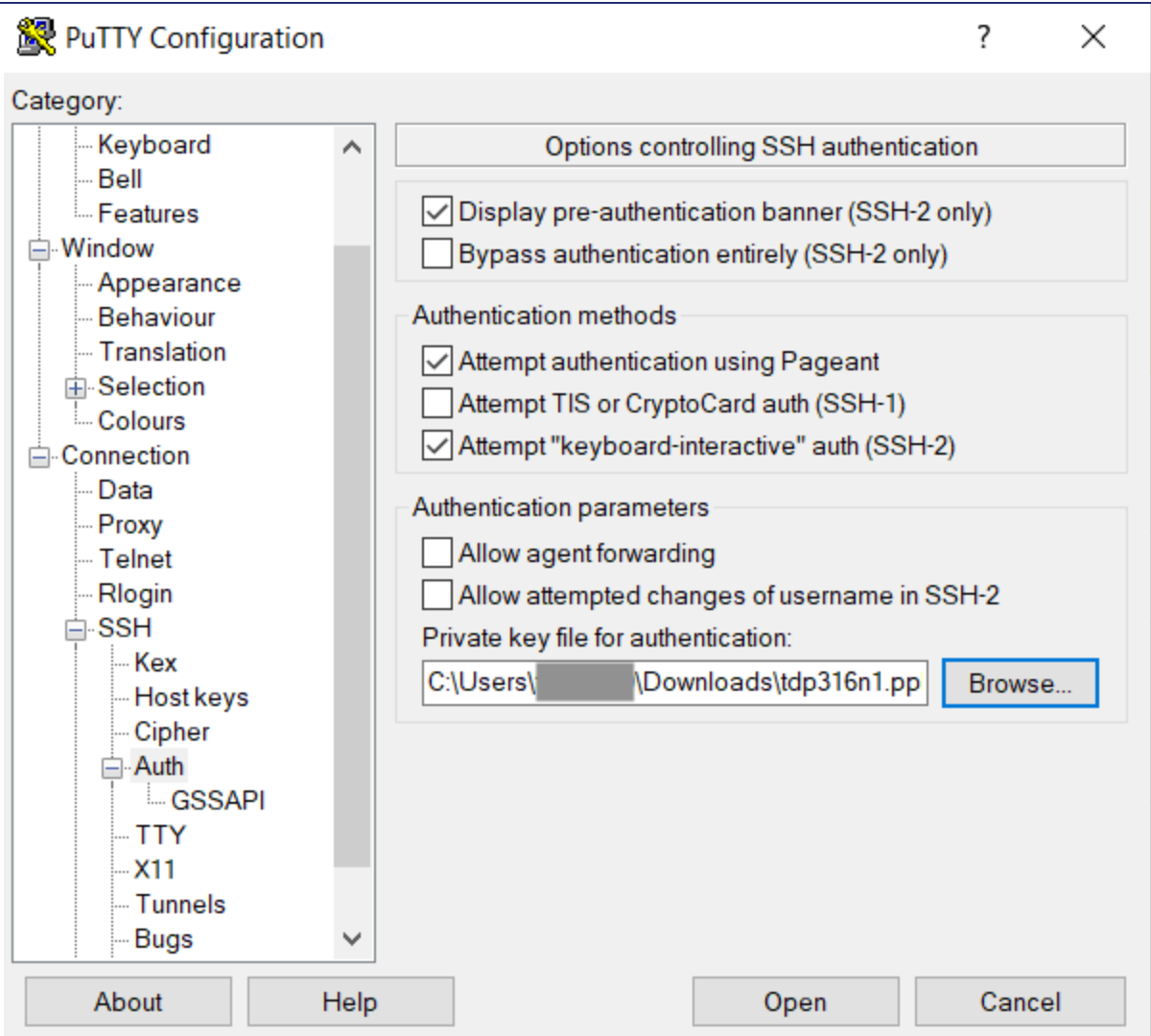


25. If using PuTTY, begin a new session.

- a. Paste the Public IP address and preface with '<username>@', in which <username> is the one created in step 11.
- b. Set the **Connection Type** to **SSH**.



- c. In the **Category** pane:
 - a. Select **Connection > SSH > Auth**.
 - b. Select **Browse** and navigate to the folder with your Private Key.
 - c. Select your Public Key file name. The file name and path is automatically places in the **Private key file for authentication** field.



d. Select **Open**.

Note

If you are unable to connect to your instance using PuTTY, go to [Connect with SSH by using PuTTY](#) for further assistance.

26. After the SSH session to your instance, run the following command, to switch to user 'root':

```
su -
```

Then follow the instructions in [Changing the Initial Password](#) and all its following steps: [Network Configuration](#), [Configuring Services After Deployment](#), and [Post Install](#) to continue configuring your Thales Data Platform cluster.

Connecting via SSH with client

1. Open the client of your choice, for example, PuTTY.
2. Ensure that you have read-only access to the private key.

```
chmod 400 tdpuser.pem
```

3. Provide a path to your SSH private key file.
Private key path:

```
~/.ssh/tdpuser
```

4. Run the example command below to connect to your VM.

```
ssh -i <private key path> tdpuser@52.225.38.222
```

Private Cloud Deployment

You can deploy a Thales Data Platform instance in a private cloud. Support is currently provided for VMware vSphere only.

Minimum Requirements

To deploy a Thales Data Platform instance, the following minimum requirements apply:

On TDP version **MISSING VARIABLE: version** the minimum requirements for each node are:

- System volume: minimum of 200 GB
- Memory: 32 GB
- vCPUs: 8

We recommend a 5-node cluster:

- 2 nodes as namenodes, Knox gateway,
- 3 nodes as datanodes and Node Manager.

See [configuring TDP nodes](#) for more details.

Note

These minimum system requirements are for a system with light to moderate load. For applications that heavily load the system, additional memory and CPU allocation are required. The system volume holds all data as well as backups.

Deploying on VMware vSphere

This section guides you through the steps needed to deploy a Virtual Thales Data Platform on VMware vSphere.

When installing the TDP OVA on ESXi 6.0 you may see an error regarding the vSphere Client not supporting the SHA256 hashing algorithm. If you see this error, please refer to the following KB article:

<https://kb.portsys.com/help/the-ovf-package-is-invalid-and-cannot-be-deployed-sha256-error>

On ESXi 6.0 you may see a warning that the configured guest (CentOS 4/5 or later) for this virtual machine does not match the guest that is currently running (CentOS 7). This is only a warning message. You may click the X button to dismiss it.

Prerequisite

- OVA file.

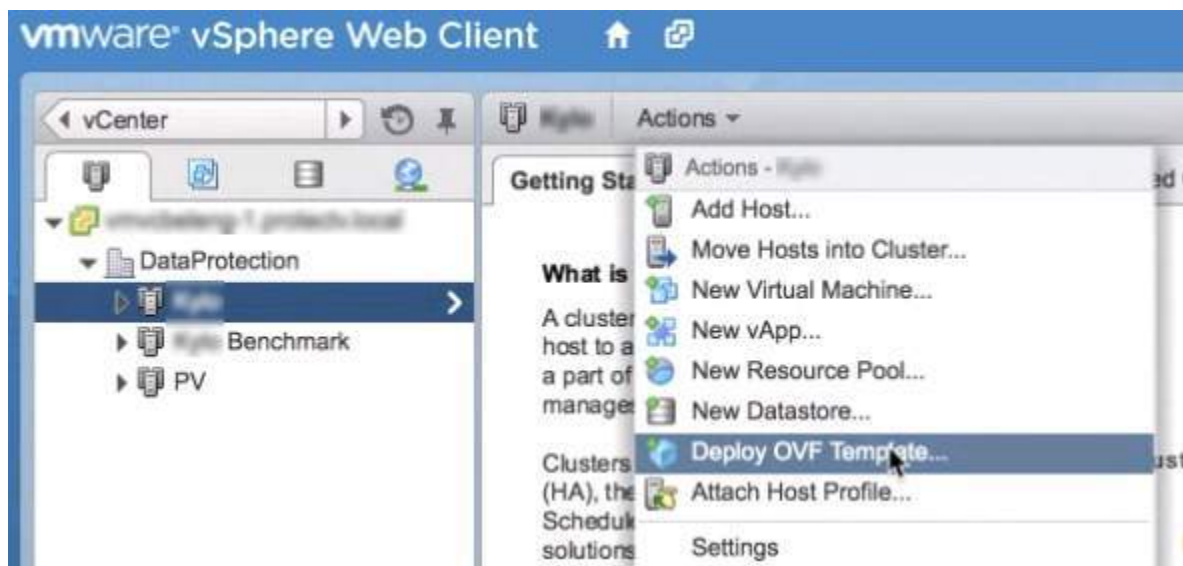
To launch the Virtual Thales Data Platform

Virtual Thales Data Platform supports VMware vSphere/ESXi v6.0 and higher versions.

1. In vSphere Client, select **Deploy OVF Template...**

Note

You will need sufficient privileges to access this selection.



2. In the **Deploy OVF Template** dialog box, enter the location of the OVA file in the **Source** field and launch the VM. After successful launch, the Virtual Thales Data Platform appears under the inventory folder in the left pane.

3. In the left pane, select the newly launched Virtual Thales Data Platform.
4. In the middle pane, select the **Summary** tab and click on the **Web console**.

The window opens in your browser.

Please follow instructions in [Changing the Initial Password](#) and all its following steps: [Network Configuration](#), [Configuring Services After Deployment](#), and [Post Install](#) to continue configuring your Thales Data Platform cluster.

2. Changing the Initial Password

The initial password must be changed before using the Thales Data Platform.

You should perform this step through ssh/putty because a private key will be generated and you will need to copy-and-paste it from the terminal to a file. You need the **public hostname** of the TDP instance (in VMware or AWS), to be able to ssh to it.

The initial `root` credentials for the image are:

Login: `root`

Password: `thales123`

The user will be prompted to change the root password after the initial login.

3. Network Configuration

Deploying on AWS

Change the Hostname

Set a hostname for your Ambari server and nodes of the Hadoop cluster. Change the hostname to the Private DNS. Run this command, for example, as root to set a new hostname:

```
hostnamectl set-hostname ip-172-31-3-54.ca-central-1.compute.internal
```

Repeat this on all hosts!

Deploying OVA on ESXi

Configure Network

The Hadoop VM is set to use DHCP by default. If a static IP is desired, you may choose **either** of these options:

- **Option 1** Use this utility:

```
/root/setup/network_setup.sh
```

- **Option 2** Manually update file `/etc/sysconfig/network-scripts/ifcfg-eth0`. Below, is an example of a static IP setting:

```
BOOTPROTO=static
IPADDR=10.3.16.227
NETMASK=255.255.0.0
GATEWAY=10.3.30.254
DNS1=10.3.110.224
DNS2=10.3.110.104
NM_CONTROLLED=no
```

Then restart the network service or reboot the server for the changes to take effect. Command to restart the network:

```
systemctl restart network
```

Set the Hostname

To set a hostname for your Ambari server, run the command:

```
hostnamectl set-hostname <hostname>
```

In the above command, substitute the `<hostname>` variable for the actual hostname of your Ambari server. For example: Set a hostname for your Ambari server and nodes of the Hadoop cluster. Change the hostname to the Private DNS. Run this command, for example, as root to set a new hostname:

```
hostnamectl set-hostname node1.hadoop-server.company.com
```

This hostname must be configured for both forward and reverse DNS. The Ambari and Hadoop applications must be able to translate the hostname to its IP address and back in order to work properly.

Update `/etc/hosts` file

Note

This step is optional if the hostname resolution is done by your DNS.

Update the `/etc/hosts` file by adding, for each node, one entry of IP address and hostname. You will need to repeat this procedure in every cluster node.

Warning

The `/etc/hosts` file on **ALL nodes** must have the hostname and IP of **ALL nodes** in the Hadoop cluster.

Example of how your `/etc/hosts` file should look like:

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.1.1.1    node1.hadoop-server.company.com
10.1.1.2    node2.hadoop-server.company.com
10.1.1.3    node3.hadoop-server.company.com
10.1.1.4    node4.hadoop-server.company.com
10.1.1.5    node5.hadoop-server.company.com
```

Where `10.1.1.1` is the IP address of the `node1` with hostname `node1.hadoop-server.company.com`. The following line applies to a `node2` in the same fashion, and so forth. The `/etc/hosts` file in this example is identical in `node1`, `node2`, `node3`, `node4` and `node5`.

Tip

If you are using DHCP, the IP address can be obtained using the `ifconfig` command.

Caution

Do **NOT** set `127.0.0.1` to the hostname of the node VM. Hadoop host registration will fail.

Deploying on Azure

Change the Hostname

Set a hostname for your Ambari server and the nodes of the Hadoop cluster. Change the hostname so that it uses the name of the VM name with the Azure's default DNS suffix (`internal.cloudapp.net`).

Run this command, for example, as root to set a new hostname:

```
hostnamectl set-hostname tdp316-56-master.internal.cloudapp.net
```

Repeat this on all hosts!

4. Configuring Services After Deployment

Configure Ambari

Setting up a Multiple-Node Cluster

Use the same OVA image to create all nodes for a multiple-node cluster, for example 5-nodes. On each node, perform the step [Changing the Initial Password and Network Configuration](#).

Choose one node as Ambari Server. On this node, run the script:

```
/root/setup/ambari_setup.sh
```

This will prompt you to set the admin password. **This will be the password of the Ambari admin.** You can use these credentials to access the Ambari UI.

When this script finishes, it shows you the private key that has been set up for the node. **Please save the key.** You will need it for setting up the Ambari server private key.

```
Aug 05 09:55:09 hadoop-ddc-top ambari-server[1985]: waiting for server start.....
Aug 05 09:55:09 hadoop-ddc-top ambari-server[1985]: Server started listening on 443
Aug 05 09:55:09 hadoop-ddc-top ambari-server[1985]: DB configs consistency check: no errors
Aug 05 09:55:09 hadoop-ddc-top ambari-server[1985]: Ambari Server 'start' completed success
Aug 05 09:55:09 hadoop-ddc-top systemd[1]: Started LSB: ambari-server daemon.
-----
https://127.0.0.1:443/api/v1/users/admin
/usr/lib/python2.7/site-packages/urllib3/connectionpool.py:769: InsecureRequestWarning: Unverified HTTPS request is being made. Adding certificate verification is strongly advised. See
https://docs.python-requests.org/en/latest/user/advanced/#ssl-tls-issues for more details.
InsecureRequestWarning)
200
-----
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAj3P0jRDRgXCPsA0z8xhB9s3Jr5ezDPmW2X00spD/bg8lKR/
+EKKXQtCo+Dhru2WnyE4aPH1k7wtjhTex1kbY1rLC900Ehs1whQx5IOvxJvIowx
SRFPpZDXYV4/BBAXqJIEyDhC/7lC6jE9d7ChICFEL0pG0bjcPv1b2kheL6zR0e
GoyShEy3AqL2FgyAwTITWqC225N+7aPKXAd7nd0McFF+NIkz7KoYXBLdjhUIK00
bNAQ+XVGr/g4zTIXGvKzLUAjEkYASlFjtXc/XmwjRkkt68QUklevCsIncBt909
nHrsPgdvAP1e8angfwS2RsbvaZuung/pk tu5gQIDAQABAoIBAfd5KH0o4D49vdX/
RZ4ck3r+MM1lw/T08D1aM0AlvLw02xwlv0em/ViYfK0ZnY75nzt03g9pFLUN/B
ajV0eJb+7vZ1l7f5VSQlw03D0xCES1qyp9YdstVvVlqcCbizId0CP0gHlIvv07FFn
nKsA9chrJBQYl1sRxcApsTy0hGP6ishIBYMDU/acwMScyZ1a4hwfrg6eyv+Gy
wAMUEduq+60THqXRlPBQyH0PKc0whfPD3TeA8duCunxoKNTCCV/Rk3BB1Tu2Nhwz
MHE1TinID8ncpXfokATnSR10cMHjZqCesGRO+QumRRD15k8nkH0CqB8Du5uFMP
npC3scEcGEA50eMkUvwxJajlngM0nApT4gB0dLvh8G+fZhdFLCXn681znEwKwT
f1ck7HyRDhRrXoqyYhFGY+BRdx+X7FYMHBSGnVrZUvz2wFdsDJF4QX2qVvKK
vF9n3MKPgITsNcBy8kFwGD3mrdRMesBu7FmsTBAxZ4yFT5fxwxcDxkCgYEABLSX
j3rda36TxxYqRIn80AHbyruXNun3zVvKN2Fr9cvIIafYMSNQ+NjHQuvsVap01H
v1u0kNct0MBTsgsVJ+dsM7qD1Ght0A30T49dbCPhmleVVQFI0bfzmcP0xBy+2VB
Z0ajhun0Djy8rFYCND0ZP27vzand05AhhRRy/66kCgYEAB0rPtvK6V5NzAS2NB/i3Y
n1+G/OnhrwtaH4TPaZMw9rBk1GpdtKwLWwyf1QfchqHRB1pGjkzjo1+ZLG6VE
Q+CUDFmPV7gk76nEY6enZ0go0H9F0oEq6LZ/1S1v1xn/wE0yYADYU83Ga6pKbVo
cy3IrcRUYz2bD9g1JkAY08ECgYAv53sVfHJ+jLgMB1Hk00Rets00jFgCrmwuf
uBFgfoJG+ePQvLL5qAARv3hwL7I++jAMB3AMdvp41BNjwVjpoHur2/v/k1Bwktw
uTqKFR0THpwwG113ng+qemVMS7cdFpYB8KIQxocREtGwelnofzTvifm4J2v045
1FtvaQK0gB0bqR7hJ0J71y15Qs46d7Zu5Ua2Xa2b4306saH+ex/1bKt0S0ASNcTh
GP2/UlH2IvXkNF+RdSINi3lpcqLj8Ajad81uAq/Ih9nLRbw00IMJoy3P7otnQle
3M67uJtXdm22y08s7Nkuog1LAhpKousnSum3eq18PBR9crFT270a
-----END RSA PRIVATE KEY-----

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQcPc/5NENGpCM9IA7PzGEH2Lcmv17MM+Z6Zf06ykP9uDYUpH/45Q
SGYrXNFCgajLMEjLCCQxYMrIDGM+PBBzbbkz7uwrC83ud0N2wUX48grPsqhhcsET20FQgrTRs8BD5duav+DjNMhca
HOST: hadoop-ddc-top
Ambari GUI: https://hadoop-ddc-top
Troot@hadoop-ddc-top ~#
```

Private
Key

Public
key



You can always view the private key later by issuing the `cat ~/.ssh/id_rsa` command, as illustrated in the example below:

```
[root@hadoop-ddc-tdp ~]# cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAj3P0jRDRqXDPSA0z8xhB9i3Jr5ezDPmW2X00spD/bg8lKR/
+EkKXQtC0+DhrUZmHyE4aPHILK7mtjhTeXlkbY1rLC99oEhslWhQxSIDvxJvIowx
SRRPpZDXAYV4/BBAXqJIOnyDhC/7lc6iE9d70hICFEL0pGQbjcPv1b2khmL6zRQo
Goy5hIy3Aql2FqyAwTITwQc225M+7sPkXAd7ndDwCFF+NIKz7KoYXLBLdjhUIK00
bNAQ+XVGr/g4zTIXGvKzLTUAiEkYA8lfjtXc/XwwjRKkwt68QUklevCsInCBt909
mHrsPgdvAPle8engYwSzRsbvaZuXmg/pktuSgQIDAQABAoIBAFd5KH0o4D49vdX/
RZA6k2r+NMilw/T08DiaWoAlvLm02xwlv0m0m/VIYfK0ZnY75nzt03g9pFLpUN/8
ajV0eJb+7YZ1L7f5VSQlW03DxCE5Iqyp9Yd6tVyViqcCbizIdDCPQgHlIvvG7FFn
n5KsA9chrJ8QYlisRxcAPs7y0hGP6ish1BYWdu/aCmWMScyZ1a4hwfrgG6eyv+Gy
w4WUEDuq+6DfMgxRLPBqnH0PKcowhFPD3IeA8duCunxoKNTcCV/Rkj8B1Tu2Nhwz
m+HEiTinI08nqpXfokATn5RJ0oMHjZqCoaGR0+QumRRDt5k8nkh0CqB8Du5uFaMP
npC5soECgYEA6VowkUVmxJajlngA6nApT4gB0wLvh8G+fZh6FLCXn60iznEWeXwT
Fiqk7HyRDhBrXoqvYhhFGY+BREdx+X7FYMCHB5GmKvRZUvz2WFdsDJF4QX2qVsKK
vF9n3WKPgIYsNcBy8kFwqD5mrdRWesBu7FwsTBaxZAYFTSfxwuxCdxkCgYEA0LSX
jJraDA36TXxYqRIm8m0AHbyYuXMun3zVvKMZFr9cviIafYMSNQ+NjHQUVsVap0lH
vluDdkNctOMBYSqaVJ+dsM7qDlGhtQA3oT49dbcPhmlVvVQF10bfzmzP0xBy+2VB
Z0ajhunGDjy8rFYCNDeZP27vzem05AhhRRy/66kCgYEA0rPtvK6V5NzASZWB/i3Y
n1+G/OnhrwtaHb4YPaZwmm9rBk1GpdtKwLEwyfiQYcqhqHRB1pGjkzjo1+ZLG6VE
Q+CUdFmPV7gYk76mEY6emZqgo0H9F0oEq6LZ/iSIvtxn/wE0yYAdYU03Ga6pKbVo
cy3IrcXRUY2bD9gJjKAYQ8ECgYAVlv53sVfHJ+jLgm81Hk6D0RetsD0jFgCrnWuf
eBFgfoJG+ePQevLL5qAARv3hwl7I++jAwB3AWdvp41mNjWVjpoHurz/v/k1BWKtW
uTqKfR0tMpmwG1i5ng+qomVBW57cdFpVB8KCIQxocrETGwmlnofzTvifm4JZvQ4S
1FtvAQKBgBdbqR7hJoJ71yiSqe4Gd7ZwSua2Xa2b4396saN+ex/lbKt0S0ASNsTh
GP2/UlH2IvxkNFeRdSIMij3Upcqlj8Ajad81uUq/Ih9nLRbwQ0iMUoy3P7otnQle
1Mb7ujtxUM22y08s7Nkuog3LAHpK0usn5uW3eqi8PBR9crFT270a
-----END RSA PRIVATE KEY-----
```

On all other nodes, run the script:

```
/root/setup/node_setup.sh
```

This will prompt you for the public key of the Ambari server. Go to the Ambari server to retrieve the public key. This can be any of the two files:

- /root/.ssh/authorized_keys
- /root/.ssh/id_rsa.pub

Hadoop Services Installation Through Ambari

The Ambari server is configured to use TLS (https) at port 443 and using a self-signed certificate.

1. Navigate to Ambari GUI by using the hostname configured in the [Set the Hostname](#). Log in as **admin** with the password that you set in the step [Configure Ambari](#).

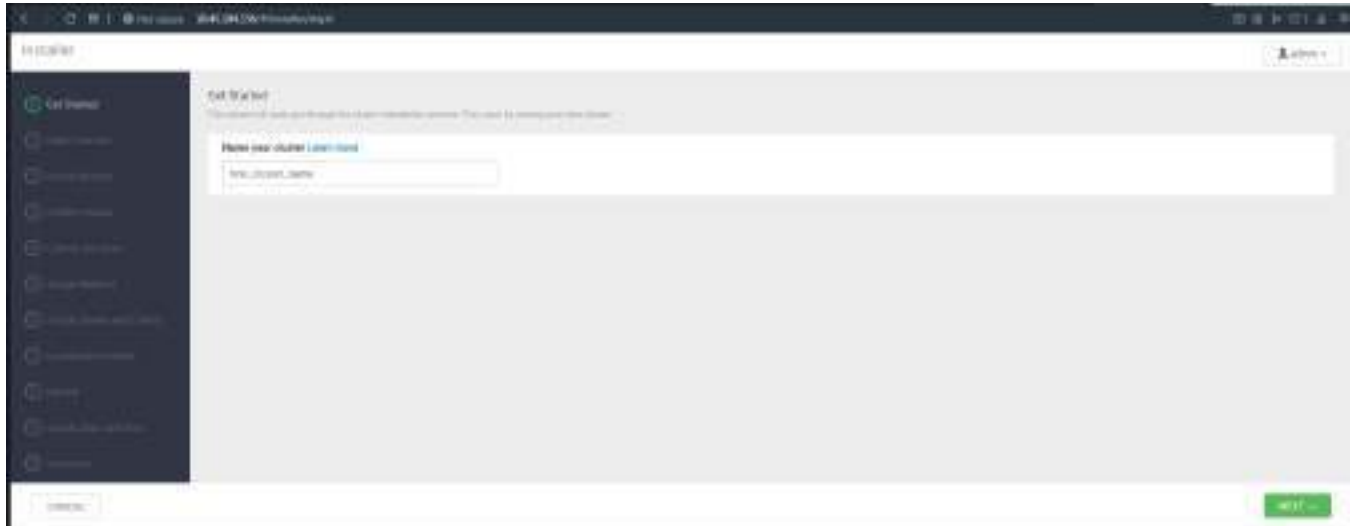
Tip

If you have troubles logging in, please refer to [Accessing Ambari](#) for further details.

- Click LAUNCH INSTALL WIZARD.

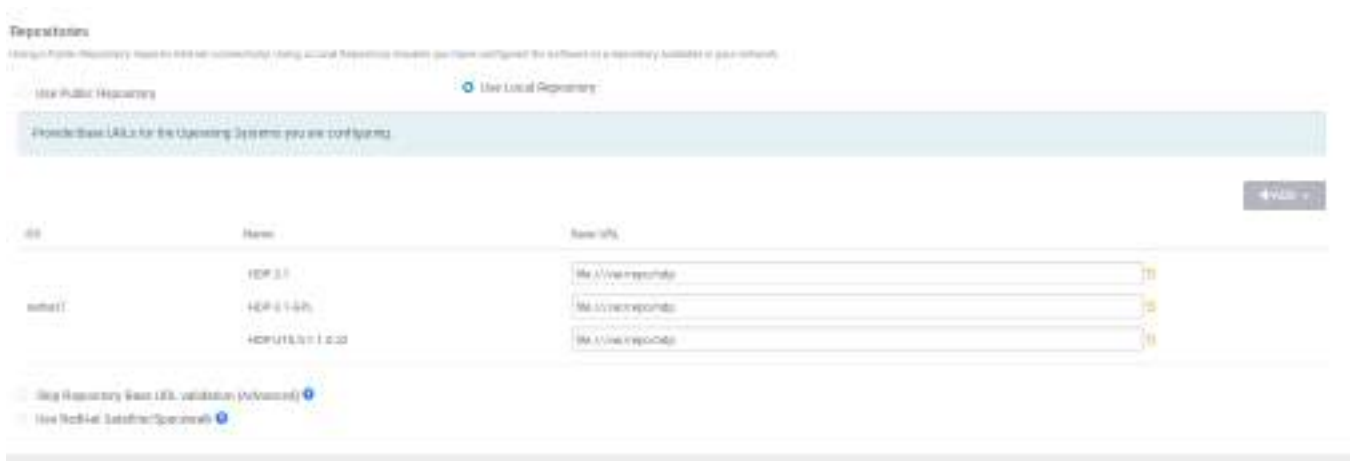
0. Name the Cluster

Choose a name that you consider more appropriate and descriptive for your cluster:



1. Select Version

At this point, you should see the UI in this state:



In the **Repositories** screen, make sure that you select the **Use Local Repository** option and fill in the **Base URL** fields per the screenshot above. The only needed OS in the Ambari configuration is Redhat 7, with the string `file:///var/repo/hdp` as value for the three paths. Those fields should be already filled in, if you can see any other OS, you can remove it.

- Click **NEXT**.

2. Install Options

In this screen the hosts are configured. Put the hostnames and private keys of all nodes that will be on your cluster. The private key is the one that you saved in the step **Configure Ambari**.

1. Click **REGISTER AND CONFIRM**.

3. Confirm Hosts

The installation will start for all nodes. This can take a few minutes.

Host	Progress	Status	Action
localhost		Registering	

Note

When this step finishes you may see the message "Some warnings were encountered while performing checks against the registered hosts above". This is a common behavior but usually there is no reason to worry so you can skip it.

1. Click **NEXT** → whenever the button gets enabled.

4. Choose Services

Here, we are going to select the services that will be available on our virtual machine. These are the required services for DDC:

- HDFS

- Yarn + MapReduce2
- HBase
- Spark
- Hive
- Tez
- Zookeeper
- Ambari Metrics
- Knox

Choose File System
Choose which file system you want to install on your cluster.

Service	Version	Description
☒ HDFS	3.1.1.0.1	Apache Hadoop Distributed File System

Choose Services
Choose which services you want to install on your cluster.

Service	Version	Description
☒ YARN + MapReduce2	3.1.1	Apache Hadoop Yarn/MapReduce (YARN)
☒ Tez	3.9.1	Tez is the next generation Hadoop Query Processing framework written on top of YARN
☒ Hive	3.1.2	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management services
☒ HBase	2.1.8	Non-relational distributed database and centralized service for configuration management & synchronization
☒ ZooKeeper	3.4.14	Centralized service which provides highly reliable distributed coordination
☐ Infra-GUI	6.1.8	Core shared service used by Ambari managed components
☒ Ambari Metrics	6.1.8	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
☐ Kafka	2.8.0	A high-throughput distributed messaging system
☒ Knox	1.8.0	Provides a single point of authentication and access for Apache Hadoop services in a cluster
☐ Log Search	6.6.0	Log aggregation, analysis, and visualization for Ambari managed services. This service is Technical Preview
☒ Spark2	2.3.3	Apache Spark 2.3 is a fast and general engine for large-scale data processing

In the list shown in the GUI, we only need to select those. In the next step, we'll add PQS and Livy to the cluster.

Tip

To save resources please unselect the following not-mandatory services:

- Kafka

- Log Search
- Infra Solr

When prompted if you accept the *Limited Functionality Warning* confirm by clicking **PROCEED ANYWAY**.

1. Click **NEXT** →.

5. Assign Masters

All the services will be automatically distributed between all the nodes. You may keep all the default settings.

The screenshot displays the 'Assign Masters' configuration interface. On the left, a list of services is shown, each with a dropdown menu indicating the master node assignment. The services and their assigned masters are:

- MasterNode: hadoop-master (16.7 GB, 8 cores)
- SlaveNode: hadoop-slave (15.7 GB, 8 cores)
- Timeline Service V1.3: hadoop-master (16.7 GB, 8 cores)
- ResourceManager: hadoop-master (16.7 GB, 8 cores)
- Timeline Service V2.3 Reader: hadoop-master (16.7 GB, 8 cores)
- YARN Registry DNS: hadoop-master (16.7 GB, 8 cores)
- History Server: hadoop-master (16.7 GB, 8 cores)
- HBase Master: hadoop-master (16.7 GB, 8 cores)
- ZooKeeper Server: hadoop-master (15.7 GB, 8 cores)
- Metrics Collector: hadoop-master (16.7 GB, 8 cores)
- CephFS: hadoop-master (16.7 GB, 8 cores)
- Hive Gateway: hadoop-master (16.7 GB, 8 cores)
- Activity Explorer: hadoop-master (15.7 GB, 8 cores)
- HST Server: hadoop-master (16.7 GB, 8 cores)
- Activity Analyzer: hadoop-master (15.7 GB, 8 cores)

On the right, a preview of the 'hadoop-master' node configuration is shown, displaying a grid of service icons that will be installed on the master node.

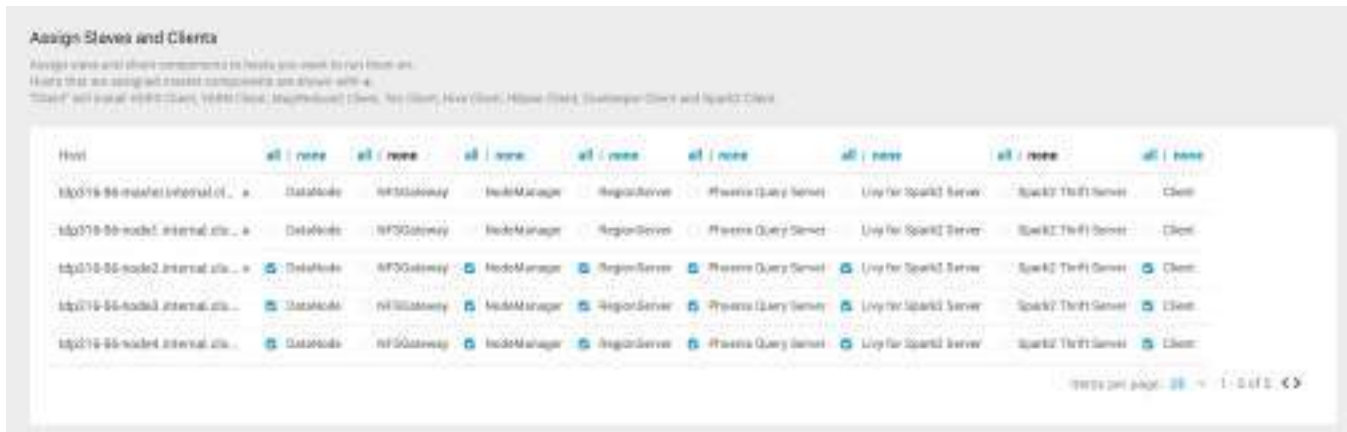
Click **NEXT** →.

6. Assign Slaves and Clients

Select "Livy for Spark2 Server" and "Phoenix Query Server" in all nodes that you want these services running. You will need to remember which hosts you selected when you reach the "Update Livy configuration for Knox" step.

Caution

This step is really important. Make sure that you select "Livy for Spark2 Server" in at least one of your hosts because it is a hard requirement for DDC.



Click **NEXT** →.

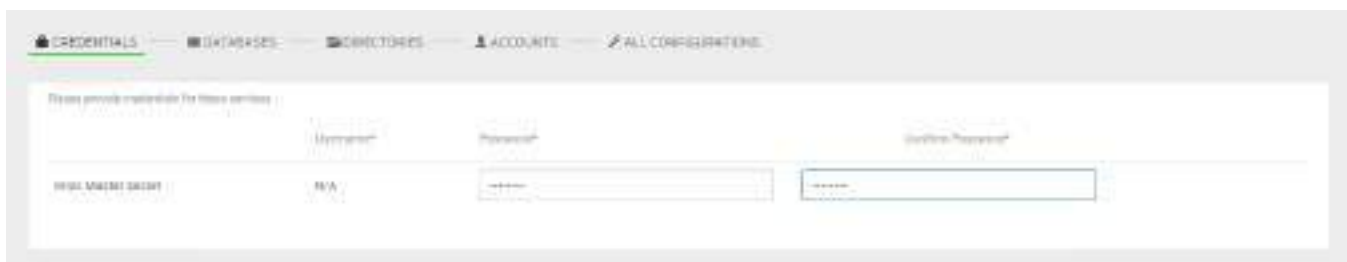
7. Customize Services

Note

The services can be installed during the TDP installation or at any moment. If this is configured during the installation, you will not have to restart the installed services. It is possible to do so when a new service is installed, however in this case you will have to restart the services.

At this point, you only need to provide credentials for different services and review the installation.

CREDENTIALS sub-step



1. Create the Knox master secret password and store it somewhere.

Note

You will need the Knox master secret password when you get to the step [Updating and Exporting the Knox Server Certificate](#). It must have at least 6 characters.

In the **CREDENTIALS** section you will also be asked to set passwords for Grafana and Hive.

2. Click **NEXT** →.

DATABASES sub-step

Click **NEXT** →.

DIRECTORIES sub-step

Click **NEXT** →.

ACCOUNTS sub-step

Click **NEXT** →.

ALL CONFIGURATIONS sub-step

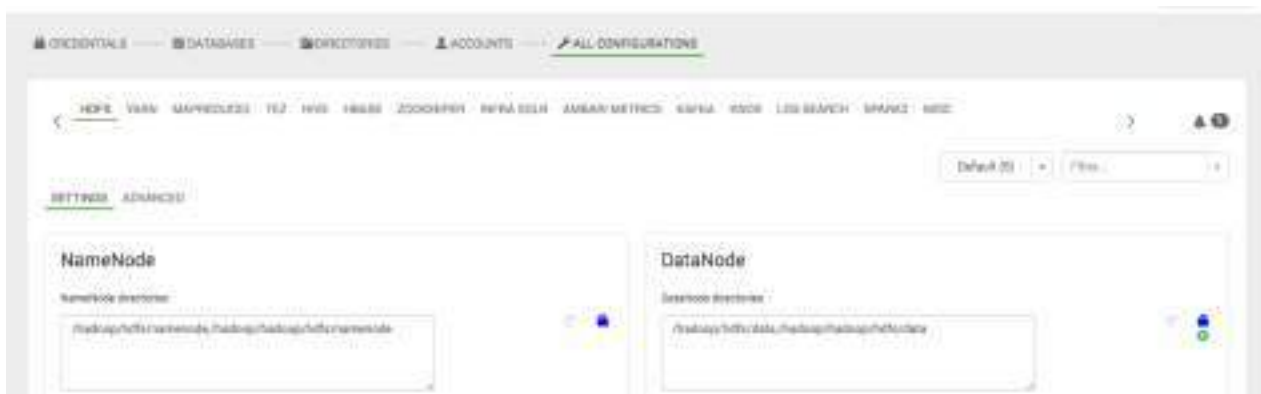
1. **HDFS** tab, **SETTINGS** sub-tab

- a. Set **NameNode directories** in **NameNode** section to:

`/hadoop/hdfs/namenode, /hadoop/hadoop/hdfs/namenode`

- b. Set **DataNode directories** in **DataNode** section to:

`/hadoop/hdfs/data, /hadoop/hadoop/hdfs/data`



2. **HDFS** tab, **ADVANCED** sub-tab

- a. If the **Advanced ssl-client** section is red please expand it and re-confirm `ssl.client.keystore.password` password

3. Click **NEXT** →.

8. Review

1. Review the configuration created
2. Click **DEPLOY** →.

9. Install, Start and Test

Note

If any of the required services fails to start automatically, you may need to start it manually. In case of repeated problems, please check out the service log for additional troubleshooting information. Refer to the article "[Viewing Service Logs](#)" on the Hortonworks documentation pages for details.

When the installation completes, click **NEXT**.

10. Summary

Click **COMPLETE**.

In the next step you will continue with the "Post Install" section.

For more detailed information about the above procedure, refer to the [Hortonworks official webpage](#).

Steps to Set up Additional Nodes in the Hadoop Cluster

Optionally, you may add an additional node (or nodes) to the cluster after the above steps.

Use the same OVA image to create all nodes for a multiple-node cluster, for example 5-nodes. On each node, perform the step [Changing the Initial Password and Network Configuration](#).

Then run the script:

```
/root/setup/node_setup.sh
```

This will prompt you for the public key of the Ambari server. Go to the Ambari server to retrieve the public key. This can be any of the two files:

- `/root/.ssh/authorized_keys`
- `/root/.ssh/id_rsa.pub`

Once the node is set up, log in to the Ambari server and add the node to the cluster with the following steps:

1. Go to the Ambari UI.
2. Click the **Hosts** button in the menu on the left. You should see the list of nodes already present on the cluster.
3. Click **Actions -> Add New Hosts**.

Note

- As in the host names field you can add more than one node, you can add more than one node at the same time. However, the previous steps are required on every node.
- The Ambari server must solve the hostname of each node and the node must be accessible from it.

4. On the ssh private key field you need to paste the private key of the Ambari Server node (typically, the first node of your cluster).
5. Click **Register and Confirm**.
6. Now, you just have to follow the instructions and add the desired services on the new node.

If you are setting up an additional node in the TDP cluster upgraded from a previous version, rename the repo path on the new node to match the repo path of the existing nodes in the cluster and recreate the repo.

For example to add a new node in cluster upgrade from 3.1.5.1 to 3.1.6:

1. Rename the repo:

```
mv /var/repo/hdp /var/repo/hdp-316
```

2. Re-create the repo:

```
createrepo /var/repo/hdp-316
```

Note

Perform these steps on a **new** node before running the `/root/setup/node_setup.sh` script.

5. Post Install

Updating and Exporting the Knox Server Certificate

When the Knox service is installed by Ambari, a self-signed certificate is created internally. However, the certificate created uses SHA1 and with a key size of 1024 bits. The **update_knox_cert.sh** script below will create and replace the certificate with a new one using SHA256 and a key size of 2048 bits.

1. Run this script in an SSH session on the TDP instance:

```
/root/setup/update_knox_cert.sh
```

2. When this script is run, it will prompt for the Knox master secret.

This secret is the one that you set in the step "7. Hadoop Services Installation Through Ambari > Customize Services" (**CREDENTIALS > Knox Master Secret**).

Finally, you need to export the SSL certificate of the Knox server and configure DDC to talk to Hadoop. You need to obtain the certificate where Knox is installed.

1. Put the Knox server certificate in a file, by using this command:

```
echo -n | openssl s_client -connect <IP_OF_KNOX_SERVER>:8443 | \
sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /tmp/hadoop.cert
```

2. Copy the certificate to the system where you will connect to the CipherTrust/DDC GUI.

Issue this command to display the certificate in the terminal window:

```
cat /tmp/hadoop.cert
```

Now, you can copy the certificate from the terminal window and paste it in its own file on your machine.

Tip

You will need to upload this file when configuring Data Discovery and Classification to talk to Thales Data Platform, be sure to save it locally. Refer to [Configuring TDP](#) section in the Data Discovery and Classification documentation to learn more.

You can find the comprehensive procedure at this [link](#) on the Hortonworks documentation pages.

Ambari additional configuration

To execute the instructions below you will need to access Ambari, please refer to [Accessing Ambari](#) for further information.

Enabling Namenode HA

Note

These steps are optional for multiple node cluster, but they are recommended.

1. In the Ambari toolbar on the left, expand **Services**, then click **HDFS**.
2. Go to the **ACTIONS** menu and click **Enable NameNode HA**.
3. Follow each step in the 'Enable NameNode HA Wizzard'.
 - a. In the **Get Started** step, input the Nameservice ID.
 - b. In **Select Hosts**, keep all the defaults and click **NEXT**.
 - c. In the **Review** step, click **NEXT**.
 - d. In **Create Checkpoint**, perform each step as listed. **NEXT** is only clickable after you complete all the steps.
 - e. In **Configure Components**, wait for the deployment of each component, then click **NEXT**.
 - f. In **Initialize JournalNodes**, perform each subsequent step. **NEXT** is only clickable after you complete all the steps.
 - g. In **Start Components**, wait for each process to complete. Then click **NEXT**.
 - h. In **Initialize Metadata**, perform each step, then click **NEXT**. Then click **OK** on the pop-up confirmation window that follows.
 - i. In **Finalize HA Setup**, wait for each process to complete, then click **DONE**.

Livy configuration

Update Livy configuration for Spark2

1. In the Ambari toolbar on the left, expand **Services**, then click **Spark2**.
2. Select the **CONFIGS** tab, then below it click **ADVANCED**.
3. Expand the **Custom spark2-defaults** section, then click the **Add Property...** link. In the **Add Property** popup, click the "multiple tags" icon to enable the **Bulk property add mode**. Then enter the following text, replacing `<zookeeper - node - hostname>` with the zookeeper node IP.

```
spark.yarn.appMasterEnv.ZK_URL_DDC = <zookeeper-node-hostname>:2181
```

Note

You can point to more than one Zookeeper node to ensure high availability on the DDC connection. To do that, use a semicolon (;) to separate the nodes. For example:

```
spark.yarn.appMasterEnv.ZK_URL_DDC = <zookeeper-node1-hostname>:2181;<zookeeper-node2-hostname>:2181
```

4. Expand the **Custom livy2-conf** section, then click **Add Property....** In the **Add Property** popup, click the "multiple tags" icon to enable the **Bulk property add mode**. Enter the following text.

```
livy.server.session.state-retain.sec = 24h
```

5. Expand the **Advanced livy2-conf** section. Update the following entry:

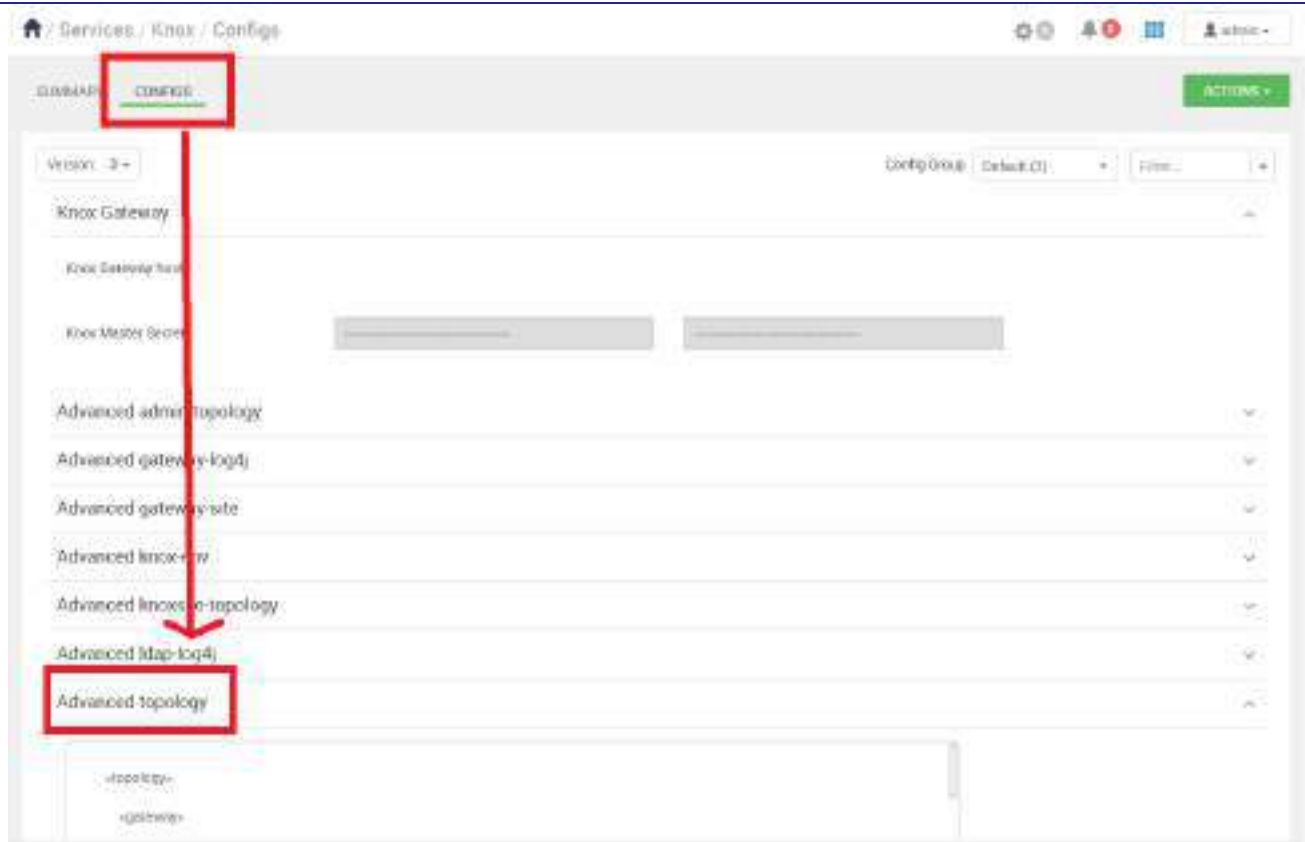
- livy.server.csrf_protection.enabled: `false`

6. Click **SAVE** and then restart **Spark2**.

At the top of the screen it will tell you that a restart is required and there is an orange **RESTART** button. Click that button and select **Restart All Affected**.

Update Livy configuration for Knox

1. In the Ambari toolbar on the left, expand **Services**, then click **Knox**.
2. Select the **CONFIGS** tab.
3. Expand the **Advanced topology** section.



4. For Spark/Livy configuration, add this entry one line before `</topology>`:

- For single node Spark2 Server:

```
<service>
  <role>LIVYSERVER</role>
  <url>http://<HOSTNAME_OF_SPARK2_SERVER>:8999</url>
</service>
```

- For multiple node Spark2 Servers:

```
<service>
  <role>LIVYSERVER</role>
  <url>http://<HOSTNAME_OF_SPARK2_SERVER1>:8999</url>
  <url>http://<HOSTNAME_OF_SPARK2_SERVER2>:8999</url>
  <url>http://<HOSTNAME_OF_SPARK2_SERVER3>:8999</url>
  ...
</service>
```

Tip

This topology will enable you to configure, in Data Discovery and Classification's **Hadoop Services**, **HDFS** with `/gateway/default/webhdfs/v1` as URI and **LIVY** with `/gateway/default/livy/v1` as URI. Refer to [Configuring TDP](#) section in the Data Discovery and Classification documentation to learn more.

5. Click **SAVE** then restart **Knox**.

At the top of the screen it will tell you that a restart is required and there is an orange **RESTART** button. Click that button and select **Restart All Affected**.

Update HDFS configuration for Knox

If your cluster has Namenode HA enabled, it is recommended to properly configure Knox to reach higher availability.

1. In the Ambari toolbar on the left, expand **Services**, then click **Knox**.
2. Select the **CONFIGS** tab.
3. Expand the **Advanced topology** section and add the following sections:

```
<provider>
<role>ha</role>
<name>HaProvider</name>
<enabled>true</enabled>
<param>
<name>WEBHDFS</name>
<value>maxFailoverAttempts=3;failoverSleep=1000;maxRetryAttempts=300;retrySleep=1000;enabled=true</value>
</param>
</provider>
```

4. Add an entry for the Phoenix Query Server.

```
//If the cluster has one PQS installed:
<service>
<role>AVATICA</role>
<url>http://<HOSTNAME_OF_PHOENIX_QUERY_SERVER>:8765</url>
</service>

//If the cluster has multiple PQS installed, you have to list all PQS nodes:
```

```
<service>
<role>AVATICA</role>
<url>http://<HOSTNAME_OF_PHOENIX_QUERY_SERVER1>:8765</url>
<url>http://<HOSTNAME_OF_PHOENIX_QUERY_SERVER2>:8765</url>
<url>http://<HOSTNAME_OF_PHOENIX_QUERY_SERVER3>:8765</url>
</service>
```

5. Click **SAVE** then restart **Knox**.

At the top of the screen it will tell you that a restart is required and there is an orange **RESTART** button. Click that button and select **Restart All Affected**.

HBase configuration

Update Zookeeper configuration for HBase

1. In the Ambari toolbar on the left, expand **Services**, then click **HBase**.
2. Select the **CONFIGS** tab, then below it click **ADVANCED**.
3. Expand the **Advanced hbase-site** section. Update the following entry:

- ZooKeeper Znode Parent: `/hbase`

Note

These properties will be added to the `hbase-site.xml` configuration file.

4. Click **SAVE** then restart all affected components.

At the top of the screen it will tell you that a restart is required and there is an orange **RESTART** button. Click that button and select **Restart All Affected**.

Update PQS configuration for HBase

Apache Phoenix requires additional modification. You need to enable namespace mapping and map the system tables to the namespace. For this procedure we are using the Ambari UI.

1. In the Ambari toolbar on the left, expand **Services**, then click **HBase**.
2. Expand the **Custom hbase-site** section. Click the **Add Property...** link. In the **Add Property** popup, click the "multiple tags" icon to enable the **Bulk property add mode**. Then enter the following text:

```
phoenix.schema.isNamespaceMappingEnabled=true
```

```
phoenix.schema.mapSystemTablesToNamespace=true
```

3. Click **SAVE** then restart all affected components.

Note

After you click **SAVE** you may see a few warnings, out of which this one pertains to the HBase configuration:

Values greater than 6.4GB are not recommended. Maximum ammount of memory each HBase RegionServer can use.

You can disregard this warning and just click **PROCEED ANYWAY** to continue.

At the top of the screen it will tell you that a restart is required and there is an orange **RESTART** button. Click that button and select **Restart All Affected**.

Updating HBase Site Configuration in Spark

To update the HBase site configuration copy the `hbase-site.xml` to Spark.

Run the following command on HBase Master node:

```
cp /etc/hbase/3.1.6-316/0/hbase-site.xml /etc/spark2/3.1.6-316/0/
```

Also run the following command on all nodes where HBase Master server and regionservers are:

```
cp /etc/hbase/conf/hbase-site.xml /etc/spark2/conf/
```

Knox Logging

Changing Knox Log Level

Warning

This is a highly recommended step.

Knox uses Log4j to keep track of its log messages. The default log level, which is INFO, may quickly cause the log file to use up the available space and DDC scans to fail.

Because of that you must change the log level to ERROR by editing the log configuration file. You can do this through Ambari, by following these steps:

1. In the Ambari toolbar on the left, click to expand **Services**, then click **Knox**.

2. Select the **CONFIGS** tab and scroll down to "Advanced gateway-log4j".
3. Open the "Advanced gateway-log4j" section and modify these parameters:
 - Change the log4j.logger.audit parameter to:

```
log4j.logger.audit=ERROR, auditfile
```

- Add these two additional parameters at the end:

```
log4j.appender.auditfile.MaxFileSize=10MB
log4j.appender.auditfile.MaxBackupIndex=10
```

4. Click **SAVE** and then restart Knox.

Purging the Knox Log Directory

If after applying these changes your Knox log grows too quickly, you may have to purge the log directory. First, however, you need to check if this is the case.

1. Check if Ambari is displaying a "NameNode Directory Status" error. This error indicates a failed directory (that is, one or more directories are reporting as not healthy.)
2. Check Ambari for a "Failed directory count" message to find out which directories are reporting problems. If the error message is showing "Failed directory count: 1" it may be the logs directory.
3. In the terminal, check the free disk space for `/var/log/` by issuing the command:

```
df -h
```

If the output of the command is showing no free disk space for the `/var/log/` directory, remove all the log files.

Start LDAP demo server

This step is applicable for demo purposes only.

Warning

It is your responsibility as the system owner to properly configure the authentication to secure the production environment. For demo purposes you can opt for the built-in LDAP server but for everything else you should configure your company's authentication/federation provider.

To set up the embedded LDAP for demo usage, follow these steps:

1. In the Ambari toolbar on the left, expand **Services**, then click **Knox**.

- Go to the **ACTIONS** menu and click **Start Demo LDAP**. Without starting the demo LDAP you cannot use the Knox logins.



Spark Tuning

Warning

This is a highly recommended step.

Spark can be configured by adjusting some properties via Ambari. There are plenty of properties that configure every aspect of the Spark behavior on the [Spark official documentation](#), but this section only just covers some of the most important properties.

Property Name	Default value	Purpose
spark.driver.cores	1	Number of cores to use for the driver process, only in cluster mode.
spark.driver.memory	1 GB	Amount of memory to use for the driver process.
spark.executor.memory	1 GB	Amount of memory to use per executor process, in MiB unless otherwise specified.
spark.executor.cores	1 in Yarn mode	The number of cores to use on each executor. In standalone and Mesos coarse-grained modes, for more detail, see this description .
spark.task.cpus	1	Number of cores to allocate for each task.
spark.executor.instances	2	The number of executors for static allocation. With <code>spark.dynamicAllocation.enabled</code> , the initial set of executors will be at least this large.

Caution

The following instructions are recommended if you have at least 8 CPU / 32GB RAM on board per cluster node.

To increase the resources dedicated to the Spark jobs, you will need to access Ambari, please refer to [Accessing Ambari](#) for further information.

1. In the Ambari toolbar on the left, expand **Services**, then click **Spark2**.
2. Select the **CONFIGS** tab, then below it click **ADVANCED**.
3. Expand **Custom spark2-defaults** and then click **Add Property...**
4. Add the following properties in the text box:

```
spark.driver.cores=3
spark.driver.memory=3g
spark.executor.cores=3
spark.executor.memory=3g
spark.executor.instances=3
```

5. Click **SAVE** then restart all affected components.

At the top of the screen it will tell you that a restart is required and there is an orange **RESTART** button. Click that button and select **Restart All Affected**.

Preparing HDFS for DDC

Warning

This procedure is only needed for a **new TDP cluster** and a **new CM**. You should omit it for any existing working environment (TDP + CM).

Creating HDFS User for DDC

By default, HDFS has permission checking enabled. Please follow the procedure below to properly configure DDC and have it ready to be used by DDC. For this example and throughout the following sections of this document, we will use user **bob** and group **hdfs**.

Tip

You can see if this permission checking is enabled by inspecting the property `dfs.permissions.enabled` in the `hdfs-site.xml` file.

1. SSH to the TDP instance and log in as `root`.
2. Create the user in Linux OS where HDFS is installed, and assign it to the desired group:

```
useradd -g hdfs bob
passwd bob
```

3. Next, you need to configure the new user in HDFS.

- a. Switch to the `hdfs` O.S. user:

```
su - hdfs
```

- b. Run the following commands to setup the user:

```
hdfs dfs -mkdir /user/bob
hdfs dfs -chown -R bob:hdfs /user/bob
hdfs dfs -ls /user
```

The result for the last command should include:

```
drwxr-xr-x - bob hdfs 0 2021-02-29 03:46 /user/bob
```

Tip

You can only use the following characters in the name of the HDFS folder: A to Z, a to z, 0 to 9, `'_'` and `'-'`.

- c. Still using the `hdfs` user, the last step is to refresh the namenode groups:

```
hdfs dfsadmin -refreshUserToGroupsMappings
```

From now on, DDC can use the `bob` user to run jobs that read and write files in the `/user/bob` directory.

- d. Optionally, you can assign a quota for the new user:


```
hdfs dfsadmin -setSpaceQuota 30g /user/bob
```

- e. Return to the `root` prompt

```
exit
```

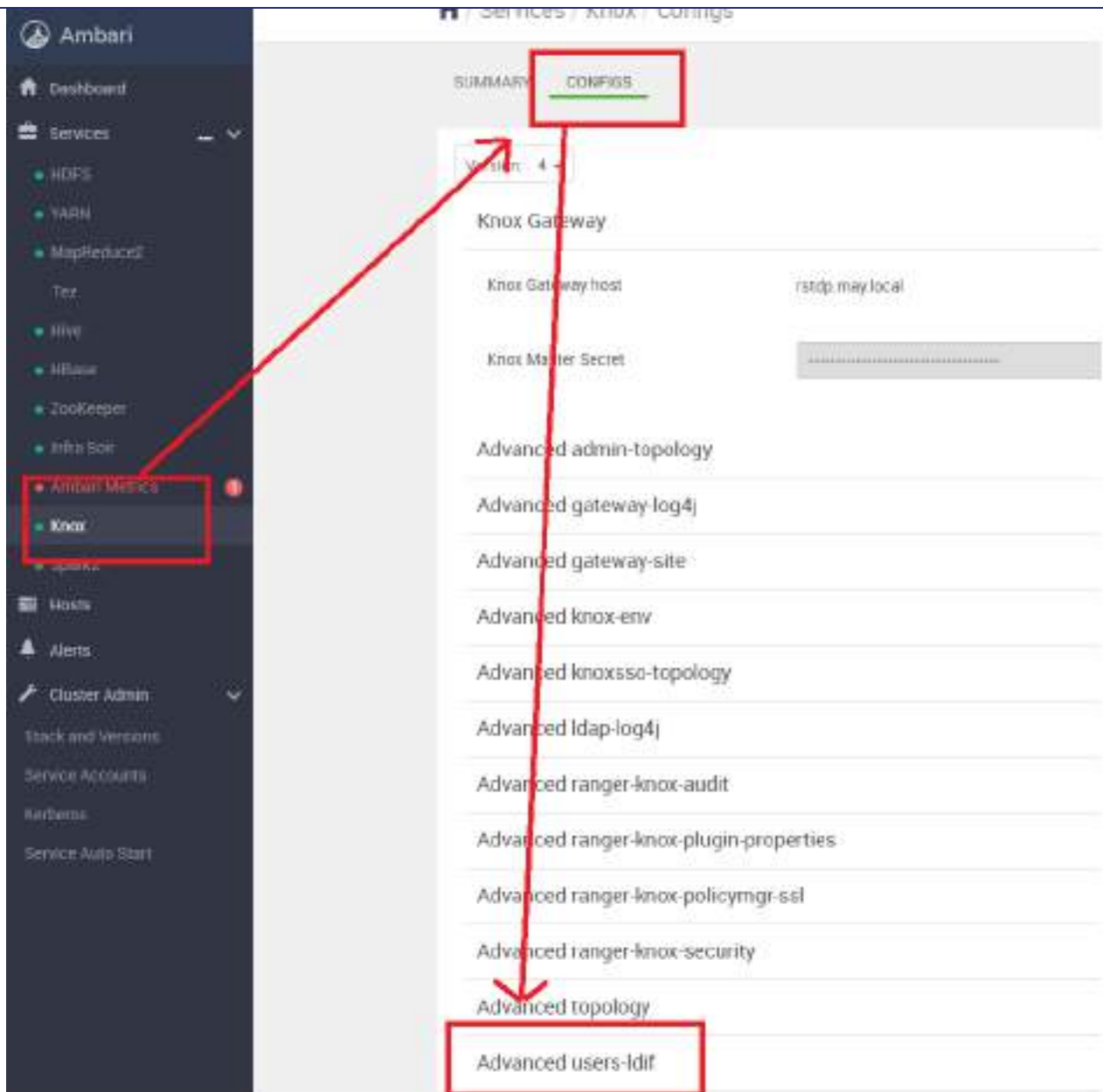
4. Lastly, you need to configure the new user in LDAP.

Warning

It is your responsibility as the system owner to properly configure the authentication to secure the production environment. For demo purposes you can opt for the built-in LDAP server but for everything else you should configure your company's authentication/federation provider.

To set up the user `bob` in the embedded LDAP for demo usage, please follow those instructions. To execute the instructions below you will need to access Ambari, please refer to [Accessing Ambari](#) for further information.

- a. In the Ambari toolbar on the left, expand **Services**, then click **Knox**.
- b. Select the CONFIGS tab and expand the **Advanced users-Idif** section.



- c. Add this text at the bottom of the text box, replacing `<user-password>` for the desired user password:

```
dn: uid=bob,ou=people,dc=hadoop,dc=apache,dc=org
objectclass:top
objectclass:person
objectclass:organizationalPerson
objectclass:inetOrgPerson
cn: bob
sn: bob
uid: bob
userPassword:<user-password>
```

- d. Click **SAVE**.
- e. Click the **ACTIONS** button and then **Stop Demo LDAP** to stop it.
- f. Click the **ACTIONS** button and then **Start Demo LDAP** start it with the new configuration.

Note

This HDFS user must be configured on the Knox connection, inside the DDC Connection Manager (in the CM Connections Management page, select Access Management > Connections Management).

Creating a HDFS Directory for DDC

For DDC to utilize Hadoop Distributed File System (HDFS), you need to create a directory under HDFS. DDC will use this space for storing scan results and reports. You can create it through the command line. You need to do it only on the **primary node**. This directory can have any name, but for this example and throughout the following sections of this document, we will use `/ciphertrust_ddc`.

1. SSH to the TDP instance and log in as root.
2. Switch to the Linux user `hdfs`, who has permissions to create and destroy folders:

```
su - hdfs
```

3. You need to create the `/ciphertrust_ddc` directory in HDFS, by issuing this command:

```
hdfs dfs -mkdir /ciphertrust_ddc
```

4. Because the default permissions will not allow the scans to write over the Hadoop Database, you have to change the access permissions. By issuing this command you will grant full access rights for the DDC user configured in [Creating HDFS User](#), that is `bob`.

```
hdfs dfs -chown -R bob:hdfs /ciphertrust_ddc
```

By running `hdfs dfs -ls /` the result should include:

```
drwxr-xr-x - bob hdfs 0 2021-08-12 03:17 /ciphertrust_ddc
```

5. Return to the `root` prompt

```
exit
```

Tip

Those steps will enable you to configure **HDFS** in Data Discovery and Classification **Hadoop Services** with `/ciphertrust_ddc` as Folder. Refer to [Configuring TDP](#) section in the Data Discovery and Classification documentation to learn more.

Administration tasks

HDFS Administration

Once you have configured HDFS using the instructions above, you can use the user interface to browse through HDFS sub-directories. Please refer to [Browsing HDFS via User Interface](#)

Optionally you can configure HDFS to use an ACL. Please refer to [HDFS ACL's](#)

Configure LDAP Authentication

In production environments it is mandatory to delegate authentication to an external LDAP server. Please refer to [Configuring LDAP Authentication](#) to get instructions on how to configure Thales Data Platform with a LDAP server.

Your TDP is now ready to be used.

THALES

Thales Data Platform

INSTALLATION GUIDE



Document Information

Document Information

Product Version	3.1.5
Document Number	007-000726-001, Rev C
Release Date	17 February 2021

Trademarks

All intellectual property is protected by copyright. All trademarks and product names used or referred to are the copyright of their respective owners. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, chemical, photocopy, recording or otherwise without the prior written permission of Thales.

Disclaimer

Thales HDP (Hadoop Data Platform) is based on Hadoop, an open source software or freeware component. Hadoop is packaged with TDP for your convenience but is provided on an "AS-IS" basis. Thales will only provide support for TDP that will include TDP cluster installation, configuration and set up of DDC. Thales expressly excludes any support for, including but not limited to, optimization, network configuration, hardware configuration or any other Hadoop administration not related to DDC initial setup and such shall be the sole and exclusive responsibility of Licensee.

Each free software or open source component provided with TDP is the copyright of its respective copyright owner. The incorporated open source projects are released primarily under the Apache Software License 2.0 ("ASLv2", full description at <https://apache.org/licenses/LICENSE-2.0>) whereas other software included may be released under the terms of alternative ASLv2 compatible open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Thales makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Thales reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon Thales to notify any person or organization of any such revisions or changes.

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

Thales invites constructive comments on the contents of this document. Send your comments, together with your personal and/or company details to the address below.

Contact Method	Contact Information
Mail	Thales 4690 Millennium Drive Belcamp, Maryland 21017 USA
Email	technical.support.DIS@thalesgroup.com

CONTENTS

Document Information	2
Setting Up the Ambari Server	6
1. The Image	6
2. Initial Login Credentials	6
3. Set the Hostname	7
3.1 Optional - Configure Networking	7
4. Configure Ambari	7
5. Steps to Set up Additional Nodes in the Hadoop Cluster	9
6. Browse the Ambari Server	10
7. Hadoop Services Installation Through Ambari	10
Name the Cluster	10
Select Version	11
Install Options	11
Choose Services	12
Assign Masters	13
Assign Slaves and Clients	14
Customize Services	14
8. Namenode HA Configuration	15
9. Configuring Knox	16
Modifying Embedded LDAP	16
Knox Authentication Information in Ambari	18
10. Configuring Phoenix Query Server (HBase)	20
11. Creating DDC Directory Under HDFS	22
Creating the DDC Directory Using the Command Line	22
Managing the DDC Directory Using the User Interface	22
12. Creating the PQS Schema	23
13. Updating and Exporting the Knox Server Certificate	24
14. Changing the Knox Log Level	25
15. Configuring CipherTrust Manager	25
Upgrading the Thales Data Platform	26
Preparing for the Upgrade	26
Registering and Installing the Target Version on the Ambari-Server Node	26
Performing the Upgrade	27
References	28
Launching the Amazon Instance	29
Installing the Instance	29
Logging in to the Instance and Additional Configuration	30

Securing Hadoop Configuration	31
Hadoop Backup	32
Preparing for the Backup	32
Backup Options	32
Create HDFS Backup	32
Restore HDFS Backup	33
Create and Restore HBase Backup	33
Snapshots	34
How to use the HBase snapshot utility	34
Creating a Backup	34
Restoring the Backup	35
Export/Import the Tables	35
How to use Export	35
How to use Import	36

Setting Up the Ambari Server

Thales Data Platform 3.1.5 (TDP) is a Big Data platform based on Hadoop technology. It can be used for POCs and production in conjunction with DDC. TDP 3.1.5 is available as an OVA and AMI image.

NOTE

- > When installing the TDP OVA on ESXi 6.0 you may see an error regarding the vSphere Client not supporting the SHA256 hashing algorithm. If you see this error, please refer to the following KB article:
<https://kb.portsys.com/help/the-ovf-package-is-invalid-and-cannot-be-deployed-sha256-error>
- > On ESXi 6.0 you may see a warning that the configured guest (CentOS 4/5 or later) for this virtual machine does not match the guest that is currently running (CentOS 7). This is only a warning message. You may click the X button to dismiss it.

1. The Image

The virtual machine must have the following specifications:

- > 64-bit Intel/AMD.
- > OVA: VMware vSphere ESXi 6.0 or later. The OVA can be downloaded from the Thales portal.
- > AMI: You will need to instantiate the AMI in your AWS. For instructions on instantiating the AMI, refer to ["Launching the Amazon Instance" on page 29](#).
- > OS type: Linux, CentOS 7 (64-bit).
- > Minimum of 150 GB disk.
- > At least 4 CPUs.
- > At least 16 GB RAM. At least 64 GB RAM if deployed as a single node.

2. Initial Login Credentials

You should perform this step through ssh/putty because a private key will be generated and you will need to copy-and-paste it from the terminal to a file. You need the **public hostname** of the TDP instance (in VMware or AWS), to be able to ssh to it.

The initial root credentials for the image are:

- > **Login:** root
- > **Password:** thales123

NOTE The user will be prompted to change the root password after the initial login.

3. Set the Hostname

To set a hostname for your Ambari server, run the command:

```
hostnamectl set-hostname <hostname>
```

In the above command, substitute the <hostname> variable for the actual hostname of your Ambari server. For example:

```
hostnamectl set-hostname hadoop-server.company.com
```

NOTE This hostname must be configured for both forward and reverse DNS. The Ambari and Hadoop applications must be able to translate the hostname to its IP address and back in order to work properly.

3.1 Optional - Configure Networking

The Hadoop VM is set to use DHCP by default. If a static IP is desired, you may choose either of these options:

a. Use this utility:

```
/root/setup/network_setup.sh
```

b. Update this file:

```
/etc/sysconfig/network-scripts/ifcfg-eth0
```

Below, is an example of a static IP setting in ifcfg-eth0:

```
BOOTPROTO=static
IPADDR=10.3.16.227
NETMASK=255.255.0.0
GATEWAY=10.3.30.254
DNS1=10.3.110.224
DNS2=10.3.110.104
NM_CONTROLLED=no
```

Restart the network service or reboot the server for the changes to take effect. Command to restart the network:

```
$ sudo systemctl restart network
```

4. Configure Ambari

Run this script to configure the host as an Ambari Server:

```
# /root/setup/ambari_setup.sh
```

This will prompt you to set the admin password. **This will be the password of the Ambari admin.** You can use these credentials to access the Ambari UI.

When this script finishes, it shows you the private key that has been set up for the node. **Please save the key.** You will need it for setting up the Ambari server private key.

```

Aug 05 09:55:09 hadoop-ddc-tdp ambari-server[1985]: Waiting for server start.....
Aug 05 09:55:09 hadoop-ddc-tdp ambari-server[1985]: Server started listening on 443
Aug 05 09:55:09 hadoop-ddc-tdp ambari-server[1985]: DB configs consistency check: no errors
Aug 05 09:55:09 hadoop-ddc-tdp ambari-server[1985]: Ambari Server 'start' completed success
Aug 05 09:55:09 hadoop-ddc-tdp systemd[1]: Started LSB: ambari-server daemon.
-----
https://127.0.0.1:443/api/v1/users/admin
/usr/lib/python2.7/site-packages/urllib3/connectionpool.py:769: InsecureRequestWarning: Unw
thdocs.org/en/latest/security.html
InsecureRequestWarning)
288
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAj3P0jRDRqXDP5A0z8xhB9i3Jr5ezCPmWZK00spD/bg8lKR/
+EKKXQtcQ+DhrU2zMyE4aPHI[K7etjhTex3kby1rLC99oEhslMHQx5IDvxJwIowx
SRPp2DXXAYV4/BBAXqJ1oNyDhC/7lc6iE9d7ChICFEL0pQ0bjcPv1b2khl62RQo
Goy5hIy3AqL2FgywTITwoc225M+7sPkXad7mdOwcFF+NIKz7KoYXLBldjhuIK00
bNAQ+XVgr/g4zTIXGvKzLHUAzEkYA8lfjtXc/XwvjRkKwt00QklevCsInCbt909
wHrsPgdvAP1e8engYw52Rsbva2uXmg/pktuSgQIDAQABAoIBAfd5KH0o4049vdx/
RZA6k2r+NMiLw/T88DiMoAlwLw02xwlv0w0n/ViYFK0ZnY73mzt83g9pFLpUW/B
ajV0eJb+7Yz1l7fSVSQlw03DxCE5Iqyp9Yd6tVvYlqcCbizIdDCP0ghIivvQ7FFn
n5KsA9chrJ8QYl1sRxcapS7y0hQp61sh1BfMDU/aCewMScyZ1a4hwrfg06eyv+0y
w4WUEdug+SDFMgxRLPBgnH0PKc0whfPD3IeASduCunxoKNTcCV/Rkj88ITu2Nhwz
wHEiTanI08nqpXfokATnGRJ0oMHjZqC0aGRO+QuRR0tSk8nkh0CqB80uSuFznp
npC5seECgYEAA6VowkUWuxJajLngA6nApT4g80wLvh80+fzh6FLCxn60iznEwKwT
Fiqk7HyR0HBRXogvyhhFGY+BREx+X7FYMKHB3GekVrZUvz2MFdsDJF4QX2qVskK
vF0n3WKPgIYsNcBy8FucD5ardRwESBu7Fm5TBAX2AYFTSfwwx0dskCgYEAA0LSX
jJ3raDA36TxxYqRIe8BBAhbyYUxMun3zVvKMZFrcvIiafYMSNQ+NjHQUvVap0LH
v1u0dkNct0NB3ygaVJ+dsN7qD1GhtQ43ot49dbcPhleVVQF106fzezP8xBy+2VB
20ajhurdjy8rFYCND02P27v2e05AhhbRRy/66kCgYEApRtvK8V5NzA5ZMB/13Y
n1+G/OnhrvtaHb4YPAZmM9rBk1GpdtKwLEwyf1QYcqhQHRB1pGjKzjo1+2L6WE
Q+CUdFmPv7grk76mEYGenZqge0H9F0oEqQLZ/151vtxn/wE0yYAdYU03Ga6pkbVe
cy3IrcXWjy2b09gJjKAYQ8ECgYAvlv53sVFHj+jLg081Hk600RwtsD0jFgCrmMuf
eBFgfoJG+ePQevLL5qAARv3hw17I++jAM63AMdvp41mNjMVjpoHuz/v/k3BMktM
uTqKfR0tHpmuG115ng+gomVEM57cdFpV8KCIQxocRETGwmlnoFzTv1fe4J2vQ45
1FtvAQK8gBdbqR7h3oJ71y1Sqe40d72wS0a2xa2b4398s2N+ex/1bkt05BASNsTh
GP2/UH2iVxkNFERdSIMi3Upcq1j8Ajad81u0q/Ih9nLRw00iMUoy3P7etnQle
IMD7ujtXUM22y08s7Nkuo3lAhpK0Usn5uM3eq18PBR9cRFT270a
-----END RSA PRIVATE KEY-----

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQc+Pc/5NENgpcnQIA7PzGEH2LcwwL7MM+2bZfQsykP9u0yupH/45Q
50YvrNFCgajLeEjLcCqxyMrIDBMhPBBzbbkz7Uw+RcB3ud0NZwUX48grPscghhcsEt20FQgrTRs0G0Sduav+DjNMhca
-----
HOST: hadoop-ddc-tdp
Ambari GUI: https://hadoop-ddc-tdp
[root@hadoop-ddc-tdp ~]#

```

Private
Key

Public
key



You can always view the private key later by issuing the `cat .ssh/id_rsa` command, as illustrated in the example below:

```
[root@hadoop-ddc-tdp ~]# cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAvj3P0jRDRqXDPSA0z8xhB9i3Jr5ezDPmW2X00spD/bg8lKR/
+EkKXQtc0+DhrUZmHyE4aPHILk7mtjhTeXlkbY1rLC99oEhslWhQxSIDvxJvIowx
SRRPpZDXAYV4/BBAxqJIOnyDhC/7lc6iE9d70hICFEL0pGQbjcPv1b2khmL6zRQo
Goy5hIy3Aql2FqyAwTITwQc225M+7sPkXAd7ndDwcFF+NIKz7KoYXLBLdjhUIK00
bNAQ+XVGr/g4zTIXGvKzLTUAiEkYA8lfjtXc/XwwjRKkwt68QUklevCsInCBt909
mHrsPgdvAPle8engYwSzRsbvaZuXmg/pktuSgQIDAQABAoIBAFd5KH0o4D49vdX/
RZA6k2r+NMilw/T08DiaWoAlvLm02xwlv0m0m/VIYfK0ZnY75nzt03g9pFLpUN/8
ajV0eJb+7YZ1l7f5VSQlW03DxCESiQyp9Yd6tVyViqcCbizIdDCPQgHlIvvG7FFn
n5KsA9chrJ8QYlisRxcaPs7y0hGP6ish1BYWdu/aCmWMScyZ1a4hwfrgG6eyv+Gy
w4WUEDuq+6DfMgxRLPBqnH0PKcowhFD3IeA8duCunxoKNTcCV/Rkj8B1Tu2NhWz
m+HEiTinI08nqXfokATn5RJ0oMHjZqCoaGR0+QumRRDt5k8nkh0CqB8Du5uFaMP
npC5soECgYEAA6VoWkUVmxJajlNgA6nApT4gB0wLvh8G+fZh6FLCXn60iznEWeXwT
Fiqk7HyRdHBrXoqvYhHFGY+BREdx+X7FYMCHB5GmKVrZUvz2WFdsDJF4QX2qVsKK
vF9n3WKPgIYsNcBy8kFwqD5mrdRWEsBu7FWsTBaxZayFTSfxwuxCdxkCgYEA0LSX
jJraDA36TXxYqRIm8m0AHbyYuXMun3zVvKMZFr9cvIIafYMSNQ+NjHQvvsVap0lH
vluDdkNct0MBYsqavJ+dsM7qD1GhtQA3oT49dbcPhmleVVQF10bfzmzP0xBy+2VB
Z0ajhunGDjy8rFYCNDeZP27vzem05AhbRRy/66kCgYEA0rPtvK6V5NzASZWB/i3Y
n1+G/0nhrwtaHb4YPaZWwm9rBk1GpdtKwLEwyfiQYcqHqHRB1pGjkzjo1+ZLG6VE
Q+CUDFmPV7gYk76mEY6emZqgo0H9F0oEq6LZ/iSivtxn/wE0yYAdYU03Ga6pKbVo
cy3IrcXRuy2bD9gJjKAYQ8ECgYAvlv53sVfHJ+jLgm81Hk6D0RetsD0jFgCrnWuf
eBFgfoJG+ePQevLL5qAARv3hwL7I++jAwB3AWdvp41mNjWVjpoHurz/v/k1BWKtW
uTqKfR0tMpmwG1i5ng+qomVBW57cdFpB8KCIQxocREtGwmlnofzTvifm4JZvQ4S
1FtvAQKBgBdbqR7hJoJ71yisQe4Gd7ZwSua2Xa2b4396saN+ex/1bKt0S0ASNsTh
GP2/UlH2IvXkNFeRdSIMij3Upqqlj8Ajad81uUq/Ih9nLRbwQ0iMUoy3P7otnQle
1Mb7ujtxUM22y08s7Nkuog3lAHpK0usn5uW3eqi8PBR9crFT270a
-----END RSA PRIVATE KEY-----
```

5. Steps to Set up Additional Nodes in the Hadoop Cluster

NOTE For a Demo installation this step is optional because a one node Hadoop deployment is sufficient for the demo purposes.

The OVA image can be used to create additional nodes for the cluster.

Perform the step ["2. Initial Login Credentials" on page 6](#) and ["3. Set the Hostname" on page 7](#).

Instead of running the `ambari_setup.sh` script in step 5, run the following script:

```
> /root/setup/node_setup.sh
```

This will prompt you for the public key of the Ambari server. Go to the Ambari server to retrieve the public key. This can be any of the two files:

```
> /root/.ssh/authorized_keys
```

```
> /root/.ssh/id_rsa.pub
```

Once the node is set up, log in to the Ambari server and add the node to the cluster with the following steps:

1. Go to the Ambari UI.
2. Click the **Hosts** button in the menu on the left. You should see the list of nodes already present on the cluster.
3. Click **Actions -> Add New Hosts**.

NOTE

- > As in the host names field you can add more than one node, you can add more than one node at the same time. However, the previous steps are required on every node.
- > The Ambari server must solve the hostname of each node and the node must be accessible from it.

4. On the ssh private key field you need to paste the private key of the Ambari Server node (typically, the first node of your cluster).
5. Click **Register and Confirm**.
6. Now, you just have to follow the instructions and add the desired services on the new node.

6. Browse the Ambari Server

The Ambari server is configured to use TLS (https) at port 443 and using a self-signed certificate.

1. Log in to the Ambari GUI by using the hostname configured in the step ["3. Set the Hostname" on page 7](#).
2. Log in as admin with the password that you set in the step ["4. Configure Ambari" on page 7](#).
3. Click LAUNCH INSTALL WIZARD.

TIP Ambari UI Not responding

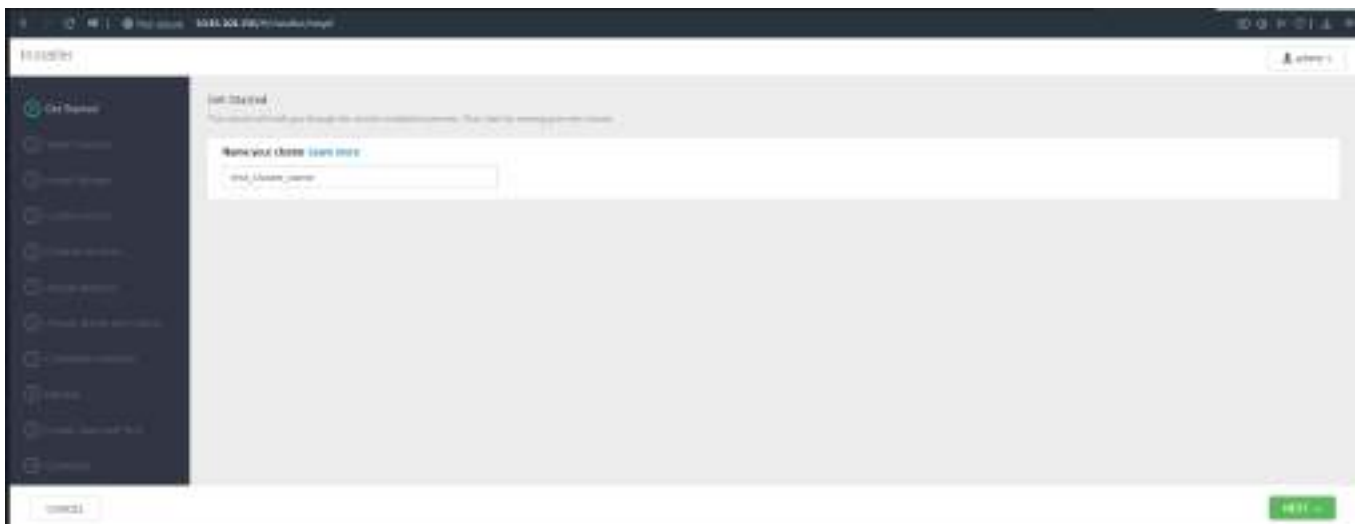
If you cannot access the Ambari UI, try resetting the Ambari service. Open an SSH session on the machine and run the following command:

```
ambari-server restart
```

7. Hadoop Services Installation Through Ambari

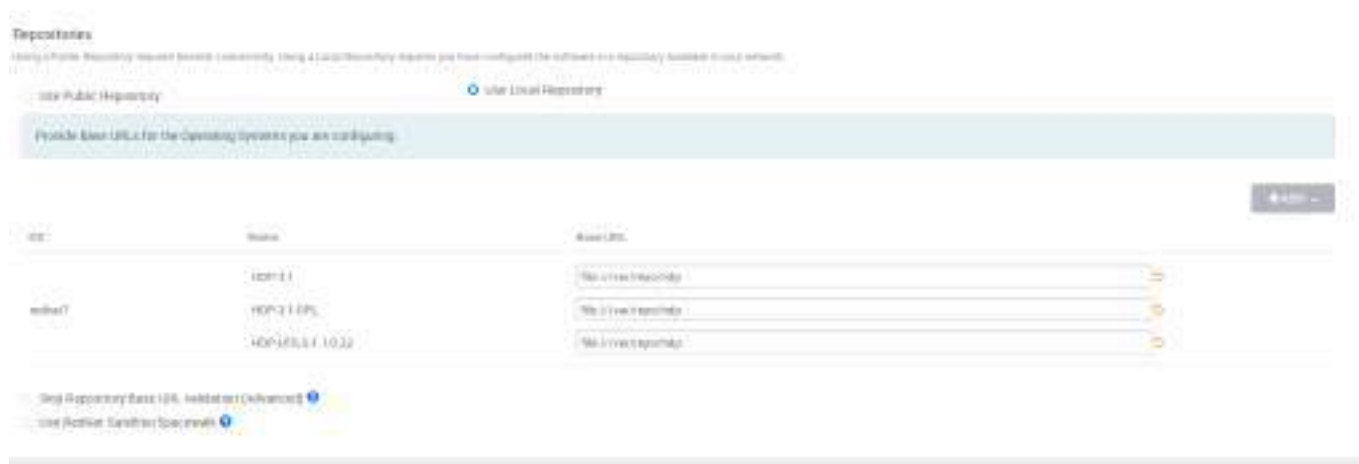
Name the Cluster

Choose a name that you consider more appropriate and descriptive for your cluster:



Select Version

At this point, you should see the UI in this state:

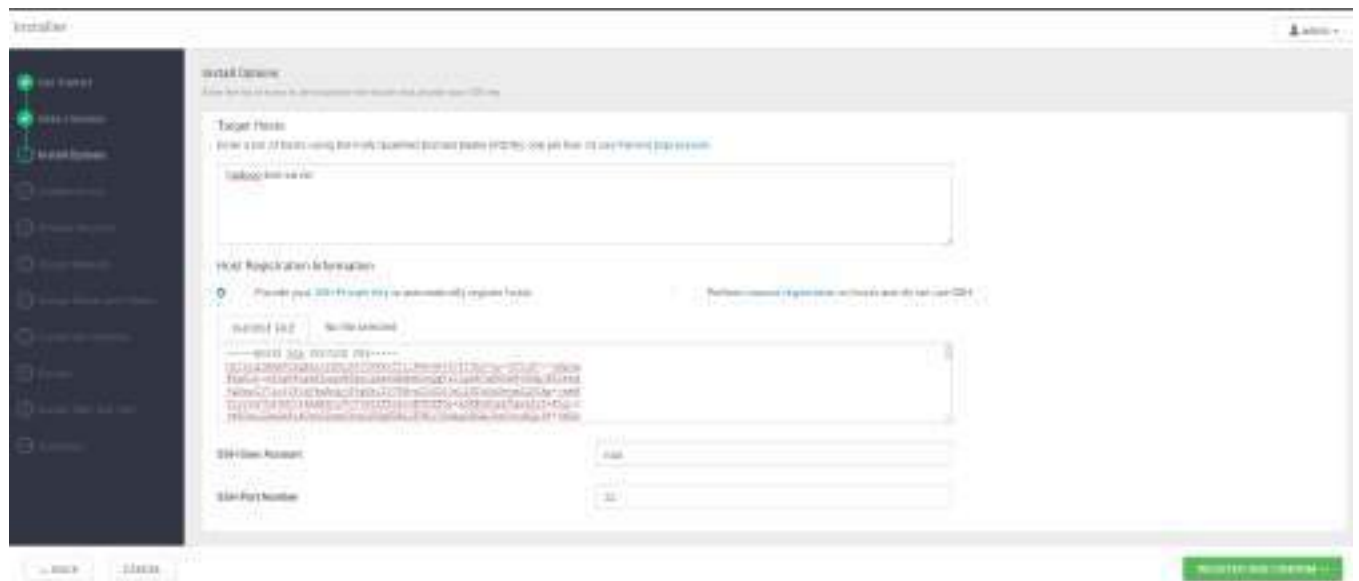


Here, the repositories are pre-filled, so the user doesn't need to know them. The local repository will point to **file:///var/repohdp**.

The only needed OS in the Ambari configuration is Redhat 7, with the string **file:///var/repohdp** as value for the three paths. If you can see any other OS, you can remove it.

Install Options

In this screen the hosts are configured. Put the hostname and private key of all nodes that will be on your cluster. The private key is the one that you saved in the step ["4. Configure Ambari" on page 7](#).



Click **REGISTER AND CONFIRM**. After that, the installation will start for all nodes. This can take a few minutes.

NOTE When this step finishes you may see the message "Some warnings were encountered while performing checks against the registered hosts above". This is a common behavior but usually there is no reason to worry so you can skip it.



Choose Services

Here, we are going to select the services that will be available on our virtual machine. These are the required services for DDC:

- > HDFS
- > Yarn + MapReduce2
- > HBase
- > Zookeeper
- > Ambari_metrics
- > Knox

Choose File System
Choose which file system you want to add to your cluster.

Service	Version	Description
☒ HDFS	3.1.1	Apache Hadoop Distributed File System

Choose Services
Choose which services you want to add to your cluster.

Service	Version	Description
☒ MRJob + MapReduce	3.1.1	Apache Hadoop MapReduce (MR) and MapReduce (MR) + MapReduce (MR)
☐ Tez	0.8.4	Tez is the next generation Hadoop Query Processing framework written on top of MR/PC.
☐ Hive	2.1.1	Data warehouse system for storing, querying & analysis of large datasets and table & storage management service.
☒ HBase	2.1.1	Non-relational distributed database and centralized service for configuration management & synchronization.
☒ Zookeeper	3.4.6	Distributed service which provides highly reliable distributed coordination.
☐ Infra Job	0.1.0	Core shared service used by Ambari managed components.
☒ Ambari Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster.
☐ Atlas	2.0.0	Atlas Metadata and Governance platform.
☐ Kafka	2.0.0	A high-throughput distributed messaging system.
☒ Hsqldb	1.0.0	Provides a single point of authentication and access for Apache Hadoop services in a cluster.
☐ Log Search	0.5.0	Log aggregation, analysis, and visualization for Ambari managed services. This service is Technical Preview.
☐ Sqoop	2.1.0	Apache Sqoop 2.1 is a fast and ground engine for large-scale data processing.

In the list shown in the GUI, we only need to select those. In the next step, we'll add PQS to the cluster.

TIP If any of the required services fails to start automatically, you may need to start it manually. In case of repeated problems, please check out the service log for additional troubleshooting information. Refer to the article ["Viewing Service Logs"](#) on the Hortonworks documentation pages for details.

Assign Masters

In our example, as we're only using one node, so this will be our master for all the services. In production environments, you can select other nodes as the service Master.

Assign Masters

Assign Master Components to the Ambari Server

NameNode	hadoop-test-server [16.7 GB, 8 cores]
SlaveNode	hadoop-test-server [16.7 GB, 8 cores]
Timeline Service V1.5	hadoop-test-server [16.7 GB, 8 cores]
ResourceManager	hadoop-test-server [16.7 GB, 8 cores]
Timeline Service V2.0 Reader	hadoop-test-server [16.7 GB, 8 cores]
YARN Registry DNS	hadoop-test-server [16.7 GB, 8 cores]
History Server	hadoop-test-server [16.7 GB, 8 cores]
HBase Master	hadoop-test-server [16.7 GB, 8 cores]
ZooKeeper Server	hadoop-test-server [16.7 GB, 8 cores]
Metrics Collector	hadoop-test-server [16.7 GB, 8 cores]
GlueJob	hadoop-test-server [16.7 GB, 8 cores]
Knox Gateway	hadoop-test-server [16.7 GB, 8 cores]
Activity Explorer	hadoop-test-server [16.7 GB, 8 cores]
HST Server	hadoop-test-server [16.7 GB, 8 cores]
Activity Analyzer	hadoop-test-server [16.7 GB, 8 cores]

hadoop-test-server [16.7 GB, 8 cores]

[NameNode](#)
[SlaveNode](#)
[Timeline Service V1.5](#)
[ResourceManager](#)
[Timeline Service V2.0 Reader](#)
[YARN Registry DNS](#)
[History Server](#)
[HBase Master](#)
[ZooKeeper Server](#)
[Metrics Collector](#)
[GlueJob](#)
[Knox Gateway](#)
[Activity Explorer](#)
[HST Server](#)
[Activity Analyzer](#)

Assign Slaves and Clients

NOTE This step is really important. Make sure that you select "Phoenix Query Server" because PQS is a hard requirement for DDC.

Assign Slaves and Clients

Assign Slave and Client Components to the Ambari Server

Assign Slave and Client Components to the Ambari Server

Host	all none	all none	all none	all none	all none	all none
hadoop-test-server	☒ Outgroups	☒ Metadata	☒ HadoopManager	☒ HadoopClient	☒ PhoenixQueryServer	☒ Client

Items per page: 25 | 1 | 10 | 20 | 50

Customize Services

At this point, you only need to provide credentials for different services and review the installation.

The screenshot shows the 'KNOX MASTER SECRET' field in the Ambari setup wizard. The field is labeled 'KNOX MASTER SECRET' and has a 'NEXT' button to its right. The field is currently empty, and a password icon is visible on the right side of the input box.

1. Create the Knox master secret password and store it somewhere. Then, keep clicking **NEXT** until the screen that shows **DEPLOY**. Click it.
2. When the installation completes, click **NEXT** and then click **COMPLETE**.

More detailed information about the steps: For further information, please refer to the [Hortonworks official webpage](#).

8. Namenode HA Configuration

This section is only required if you are deploying TDP in more than 1 node.

NOTE It is recommended to perform this step at setup, as any changes to the Namenode HA configuration will result in downtime.

1. Go to the Ambari UI and select the HDFS services (**Services > HDFS** in the Ambari menu on the left).
2. Click **ACTIONS** and then the **Enable NameNode HA** section.
3. Follow the wizard.

The screenshot shows the Ambari UI 'Services / HDFS / Summary' page. The left sidebar shows the 'Services' menu with 'HDFS' selected. The main content area displays the 'Summary' tab for HDFS services. The 'Summary' section shows the status of the services: 'NameNode' is 'Stopped', 'DataNode' is 'Started', and 'SecondaryNameNode' is 'Stopped'. The 'Actions' menu is open on the right, showing the 'Enable NameNode HA' option. The 'Service Metrics' section shows various metrics for the services, including 'NameNode' (1820, 95.91%), 'DataNode' (1820, 95.91%), and 'SecondaryNameNode' (1820, 95.91%).

9. Configuring Knox

The Apache Knox Gateway is a system that provides a single point of authentication and access for Hadoop services in a cluster. While Knox supports many authentication/federation providers this document will only describe configuring LDAP. Please check Apache Knox Gateway's online configuration for details on the other providers:

<http://knox.apache.org/books/knox-1-0-0/user-guide.html#Authentication>

****WARNING**** It is your responsibility as the system owner to properly configure the authentication to secure the production environment. For demo purposes you can opt for the built-in LDAP server but for everything else you should configure your company's authentication/federation provider.

A change in the Knox configuration is needed to provide access from the DDC client. There are two ways a user could configure Knox for DDC. One is to **modify the default topology** and the other is to **create a new topology**. If you want to create a new topology, refer to the Knox documentation.

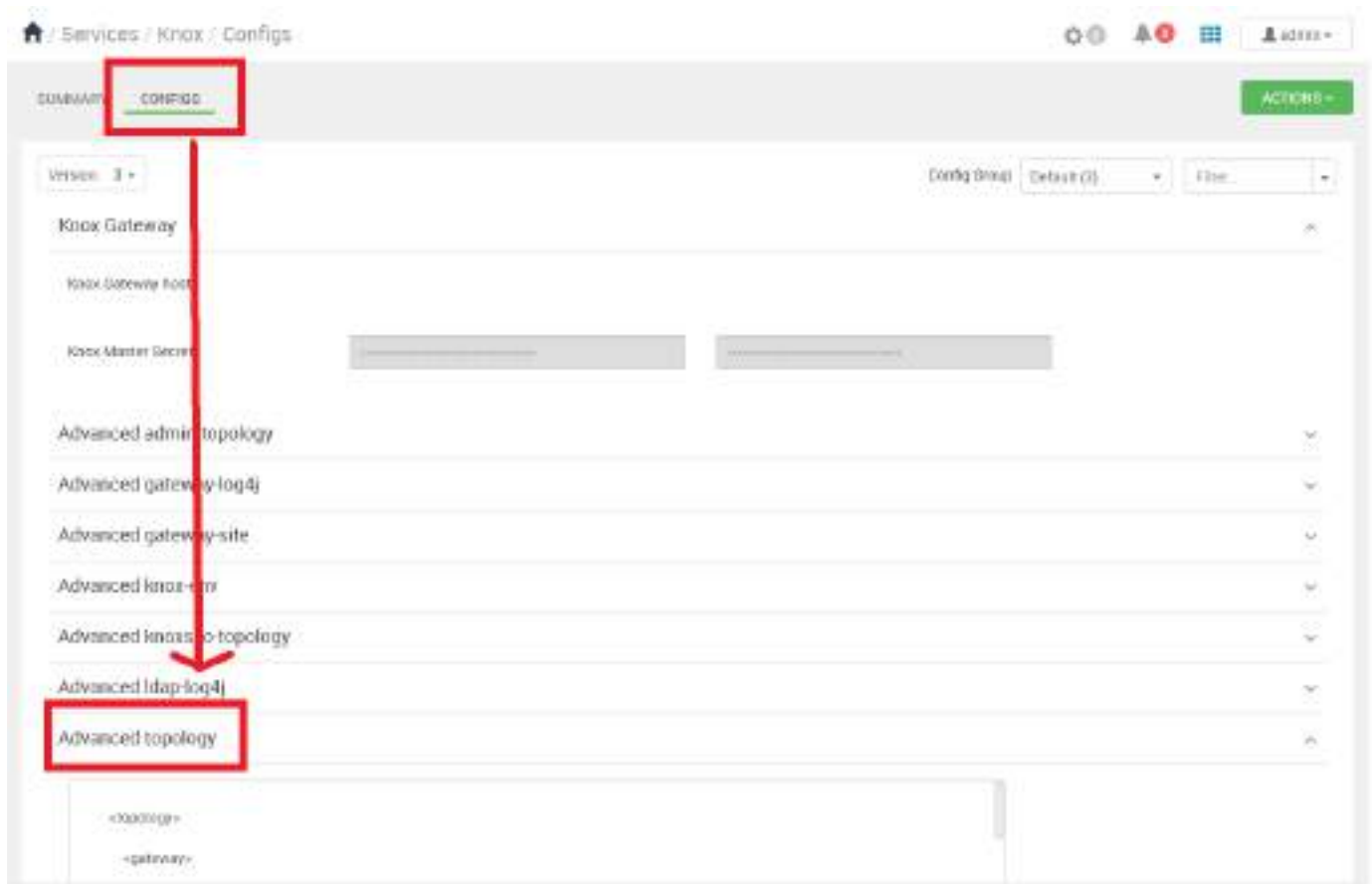
Modifying Embedded LDAP

To set up the embedded LDAP for demo usage, follow these steps:

1. Go to Knox configuration in the Ambari server UI.
2. (applicable for demo purposes only) Go to the **ACTIONS** menu and click **Start Demo LDAP**. Without starting the demo LDAP you cannot use the Knox logins.



3. Expand the **Advanced topology** section.



4. Add an entry for the PQS server.

- Single node configuration:

```
<service>
  <role>AVATICA</role>
  <url>http://<HOSTNAME_OF_PHOENIX_QUERY_SERVER>:8765</url>
</service>
```

- (optional) If the cluster is Namenode HA enabled, you have to list all PQS nodes:

```
<service>
  <role>AVATICA</role>
  <url>http://<HOSTNAME_OF_PHOENIX_QUERY_SERVER1>:8765</url>
  <url>http://<HOSTNAME_OF_PHOENIX_QUERY_SERVER2>:8765</url>
  <url>http://<HOSTNAME_OF_PHOENIX_QUERY_SERVER3>:8765</url>
  ...
</service>
```

5. (optional for creating a cluster) If the cluster is Namenode HA enabled, add the following section:

```
<provider>
  <role>ha</role>
  <name>HaProvider</name>
  <enabled>true</enabled>
  <param>
    <name>WEBHDFS</name>
```

```
<value>maxFailoverAttempts=3;failoverSleep=1000;maxRetryAttempts=300;retrySleep=1000;enabled=true</value>
</param>
</provider>
```

6. (applicable for non-demo purposes) Example for LDAP:

```
<provider>
<role>authentication</role>
<name>ShiroProvider</name>
<enabled>true</enabled>
<param>
<name>main.ldapRealm</name>
<value>org.apache.shiro.realm.ldap.JndiLdapRealm</value>
</param>
<param>
<name>main.ldapRealm.userDnTemplate</name>
<value>uid= {0},ou=people,dc=hadoop,dc=apache,dc=org</value>
</param>
<param>
<name>main.ldapRealm.contextFactory.url</name>
<value>ldap://localhost:33389</value>
</param>
<param>
<name>main.ldapRealm.contextFactory.authenticationMechanism</name>
<value>simple</value>
</param>
<param>
<name>urls./**</name>
<value>authcBasic</value>
</param>
</provider>
```

7. Click **SAVE** then restart all the affected components.

At the top of the screen it will tell you that a restart is required and there is an orange **RESTART** button. Click that button and select **Restart All Affected**.

8. With this configuration, use '/gateway/default/avatica' for the PQS URI and '/gateway/default/webhdfs/v1' for the HDFS URI in "Configuring HDFS" and "Configuring HBase" in the "Thales CipherTrust Data Discovery and Classification Deployment Guide".

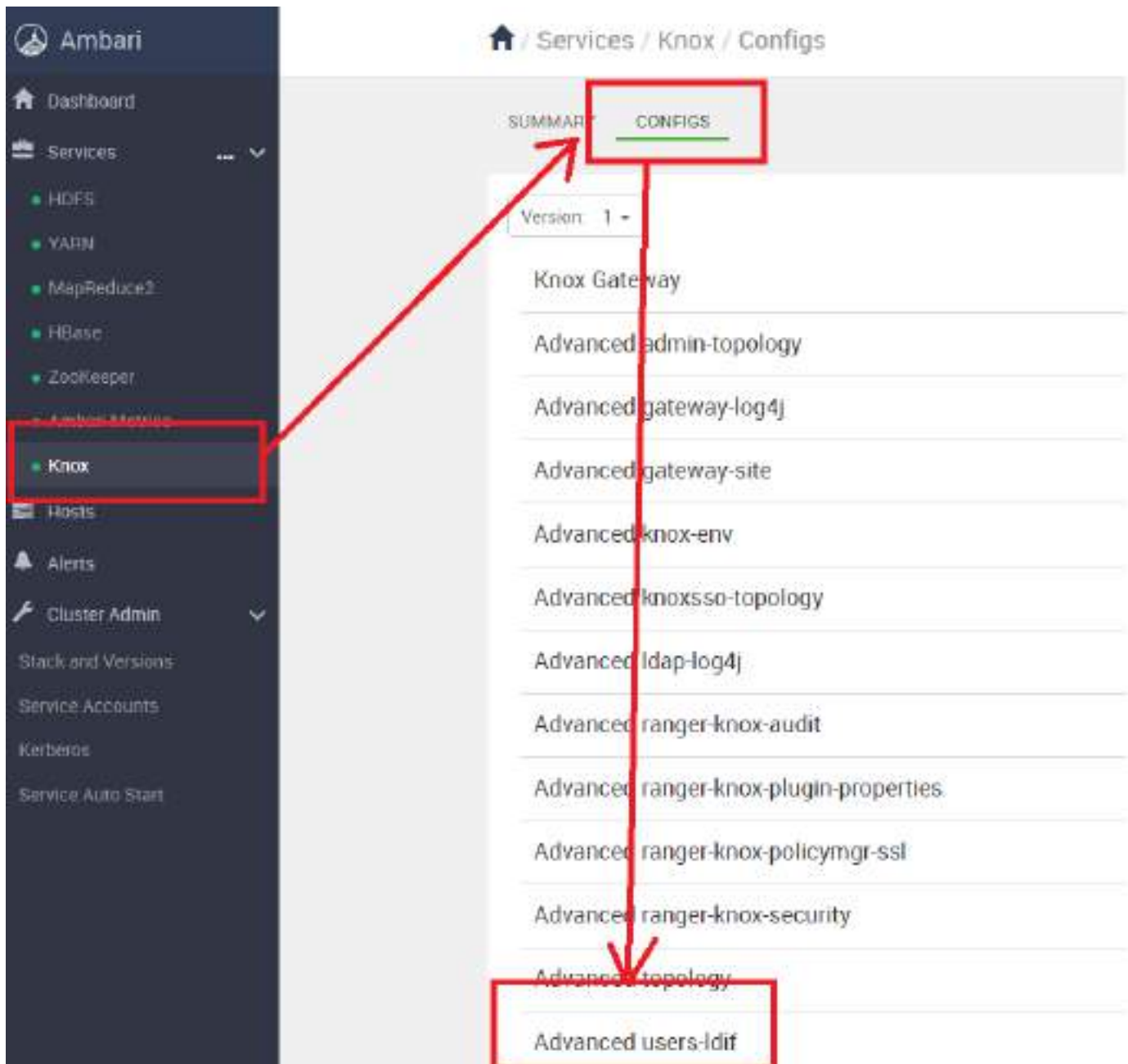
For instructions on configuring an external LDAP or Active Directory refer to the Apache Knox online documentation:

- > [LDAP Configuration](#)
- > [Active Directory Configuration](#)

Knox Authentication Information in Ambari

When you are using the demo LDAP embedded with Knox, there is a file with the users credentials. To view these default credentials, select:

Knox > CONFIGS > Advanced-users-Idif



Scroll down to the '#entry for sample user admin' to find the 'userPassword', which by default is 'admin-password'.

Advanced users-ldif

```

cn: Guest
sn: User
uid: guest
userPassword: guest-password

# entry for sample user admin
dn: uid=admin,ou=people,dc=hadoop,dc=apache,dc=org
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Admin
sn: Admin
uid: admin
userPassword: admin-password

# entry for sample user sam
dn: uid=sam,ou=people,dc=hadoop,dc=apache,dc=org
objectclass: top
objectclass: person

```

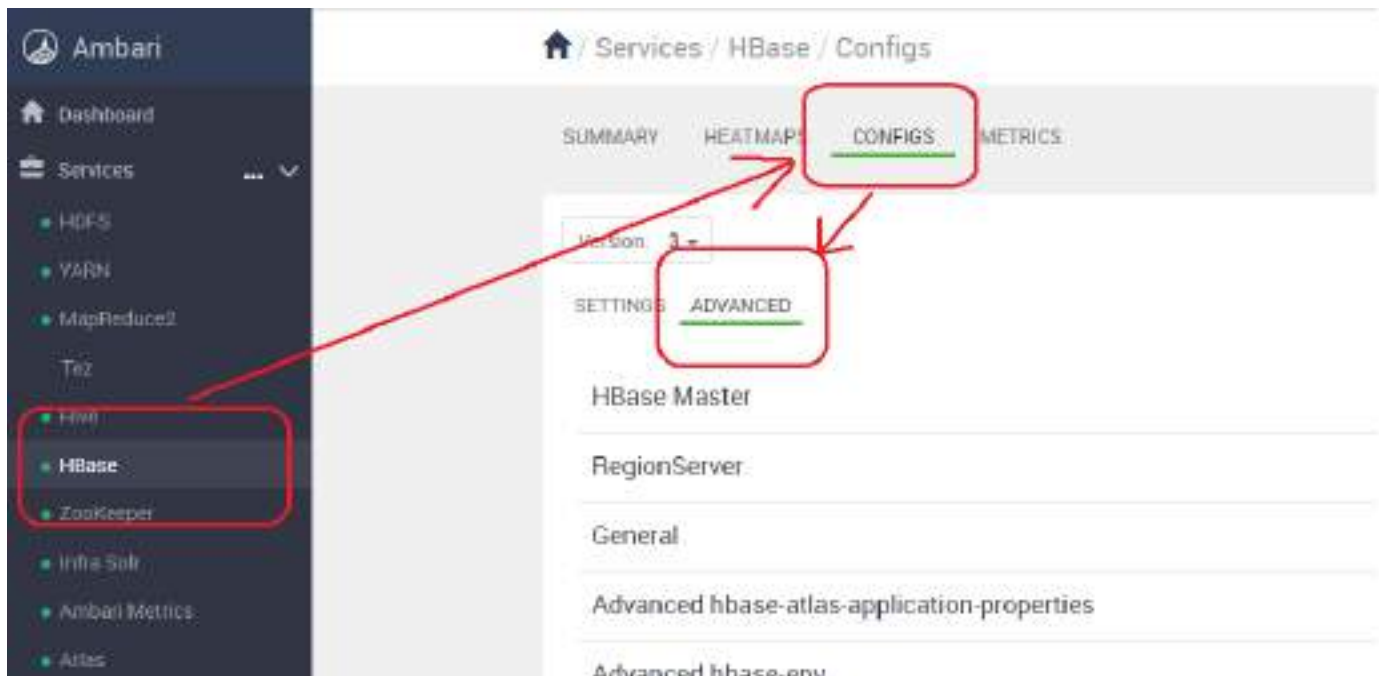
Default admin user**NOTE**

In production environments, when you are using a real LDAP or Active Directory, you have to use a user with admin permissions on the service.

10. Configuring Phoenix Query Server (HBase)

Apache Phoenix requires additional modification before you can use it with DDC. You need to enable namespace mapping and map the system tables to the namespace. For this procedure we are using the Ambari UI.

1. In the advanced HBase configuration, scroll down to **Custom hbase-site (HBase > CONFIGS > ADVANCED > Custom hbase-site)**.



2. Use the **Add Property...** link to add these two additional properties (as illustrated in the image further below):

```
phoenix.schema.isNamespaceMappingEnabled=true
```

```
phoenix.schema.mapSystemTablesToNamespace=true
```

NOTE These properties will be added to the **hbase-site.xml** configuration file.

3. Click **SAVE** then restart all the affected components.

At the top of the screen it will tell you that a restart is required and there is an orange **RESTART** button. Click that button and select **Restart All Affected**.

11. Creating DDC Directory Under HDFS

For DDC to utilize Hadoop Distributed File System (HDFS), you need to create a directory under HDFS. DDC will use this space for storing scan results and reports. You can create it through the command line and using the Ambari UI. You need to do it only on the **primary node**. This directory can have any name, but for this example and throughout the following sections of this document, we will use **/ciphertrust_ddc**.

TIP You will need this path later to configure CipherTrust. See "Configuring HDFS" in the "Thales CipherTrust Data Discovery and Classification Deployment Guide".

Creating the DDC Directory Using the Command Line

1. SSH to the TDP instance and log in as root.
2. Switch to the "hdfs" user, who has permissions to create and destroy folders:

```
su - hdfs
```

3. Create the DDC directory in HDFS by issuing this command:

```
hdfs dfs -mkdir /ciphertrust_ddc
```

4. Because the default permissions will not allow the scans to write over the Hadoop Database, you have to change the access permissions.

Issue this command to grant full access rights to the directory for all users:

```
hdfs dfs -chmod 777 /ciphertrust_ddc
```

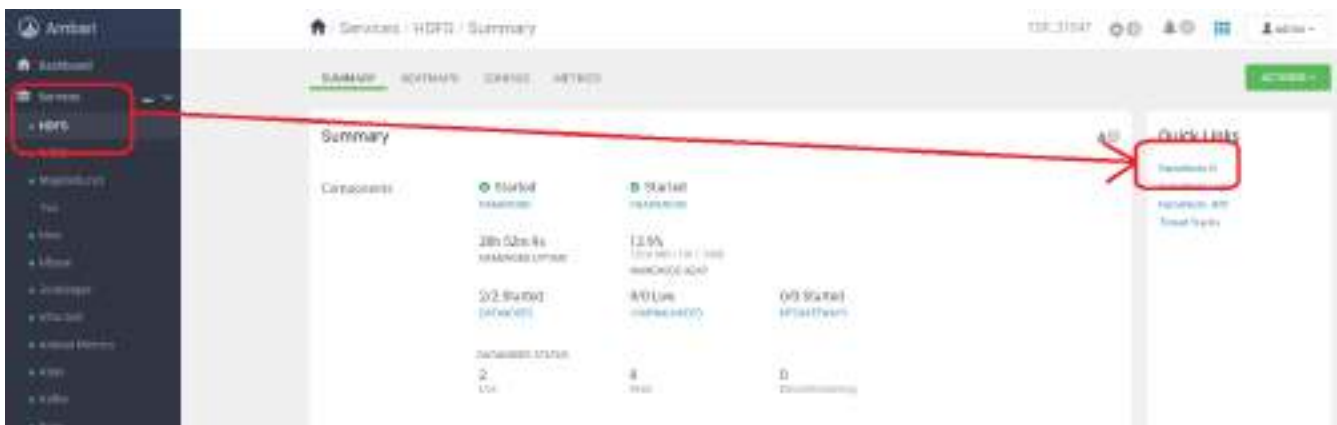
NOTE: After entering the third command, enter 'exit' to return to the root prompt.

Managing the DDC Directory Using the User Interface

NOTE This section is just for your information and optional. You do not have to perform the steps below.

Once you have created the HDFS directory using the CLI, you can use the user interface to create sub-directories, download the files, or simply check the HDFS status.

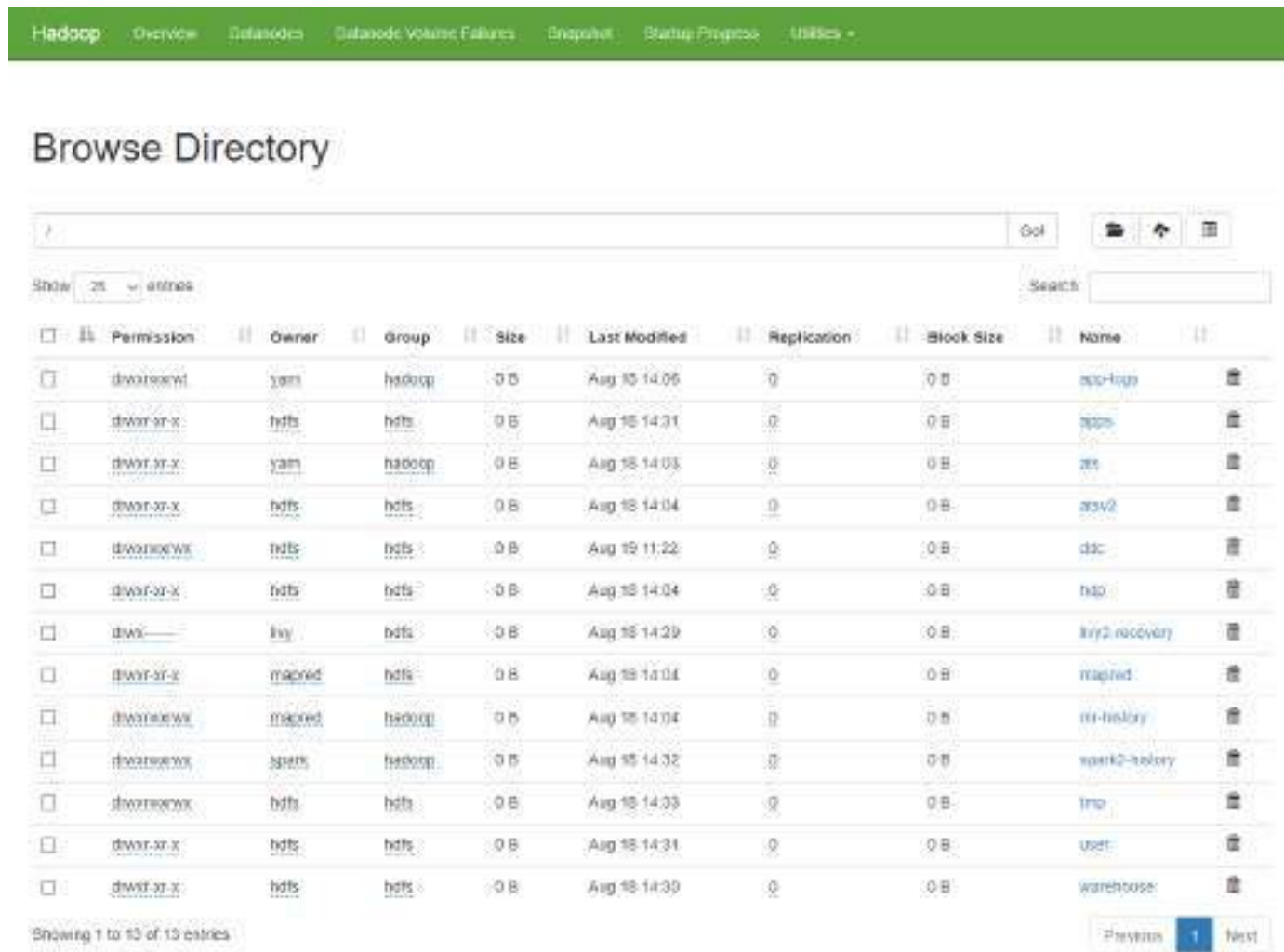
1. Open the Namenode UI:



2. Click to expand the **Utilities** menu and next click the **Browse the file system** link.



From here, you can see all the HDFS directories. In this particular case, you cannot create a directory on the root because the user used by the UI has no permissions to do it.



12. Creating the PQS Schema

DDC uses Hbase to store scan results and reports and uses Phoenix Query Server to store and access this data. To that end, you have to create a PQS schema in the cluster. We will use **ciphertrust_ddc** as the name of the schema.

TIP This is a known issue with the PQS library. It can cause an error when the user tries to configure the PQS service on DDC for the first time. It can be avoided by creating the PQS schema before configuring DDC.

1. Open an SSH session on a Hadoop node which has PQS installed.

If PQS is installed on multiple nodes, choose any one of them. If you don't remember which node has PQS installed, you can go to the **Hosts** tab and see the services installed on each node.

2. Run the **sqlline-thin.py** script to open the SQL interpreter. For example:

```
/usr/hdp/3.1.5.0-316/phoenix/bin/sqlline-thin.py http://localhost:8765
```

(Typically, you will find the script in `/usr/hdp/3.1.5.0-316/phoenix/bin/`).

3. Within the SQL interpreter, run the following command to create the DDC schema:

```
CREATE SCHEMA IF NOT EXISTS ciphertrust_ddc;
```

4. Use Ctrl-D to exit.

Use **ciphertrust_ddc** as the PQS schema in the CipherTrust configuration in the "Configuring HBase" section of the "Thales CipherTrust Data Discovery and Classification Deployment Guide".

You can find more details on creating the PQS schema in the [official Phoenix documentation](#) (section Usage => Client).

13. Updating and Exporting the Knox Server Certificate

When the Knox service is installed by Ambari, a self-signed certificate is created internally. However, the certificate created uses SHA1 and with a key size of 1024 bits. The **update_knox_cert.sh** script below will create and replace the certificate with a new one using SHA256 and a key size of 2048 bits.

1. Run this script in an SSH session on the TDP instance:

```
# /root/setup/update_knox_cert.sh
```

2. When this script is run, it will prompt for the Knox master secret.

This secret is the one that you set in the step "7. Hadoop Services Installation Through Ambari > Customize Services" (**CREDENTIALS > Knox Master Secret**).

Finally, you need to export the SSL certificate of the Knox server and configure DDC to talk to Hadoop. You need to obtain the certificate where Knox is installed.

1. Put the Knox server certificate in a file, by using this command:

```
$ echo -n | openssl s_client -connect <IP_OF_KNOX_SERVER>:8443 | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /tmp/hadoop.cert
```

2. Copy the certificate to the system where you will connect to the CipherTrust/DDC GUI.

Issue this command to display the certificate in the terminal window:

```
# cat /tmp/knox.cert
```

Now, you can copy the certificate from the terminal window and paste it in its own file on your machine. You will need this file when configuring DDC to talk to Hadoop. Refer to "Configuring CipherTrust Manager" in the "Thales CipherTrust Data Discovery and Classification Deployment Guide".

You can find the comprehensive procedure at this [link](#) on the Hortonworks documentation pages.

14. Changing the Knox Log Level

Knox uses Log4j to keep track of its log messages. The default log level is INFO which may quickly cause the log file to use up the available space and DDC scans to fail. Unless you need this level of detail in the logs you should change it to ERROR by editing its configuration file.

1. SSH to the TDP instance and log in as root.

2. Stop Knox by issuing this command:

```
su -l knox -c '$gateway_home/bin/gateway.sh stop'
```

3. Switch to the "hdfs" user:

```
su - hdfs
```

4. Open the `/etc/knox/conf/gateway-log4j.properties` file for editing.

- Change the `log4j.logger.audit` parameter to:

```
log4j.logger.audit=ERROR, auditfile
```

- Add these two additional parameters at the end:

```
log4j.appender.auditfile.MaxFileSize=10MB
```

```
log4j.appender.auditfile.MaxBackupIndex=10
```

5. Start Knox:

```
su -l knox -c '$gateway_home/bin/gateway.sh start'
```

If after applying these changes your Knox log grows too quickly, you may have to purge the log directory. First, however, you need to check if this is the case.

1. Check if Ambari is displaying a "NameNode Directory Status" error. This error indicates a failed directory (that is, one or more directories are reporting as not healthy.)
2. Check Ambari for a "Failed directory count" message to find out which directories are reporting problems. If the error message is showing "Failed directory count: 1" it may be the logs directory.
3. In the terminal, check the free disk space for **/var/log** by issuing the command:

```
df -h
```

If the output of the command is showing no free disk space for the **/var/log** directory, remove all the log files.

15. Configuring CipherTrust Manager

Refer to the "DDC Deployment Guide" to complete the CipherTrust Manager configuration.

Upgrading the Thales Data Platform

This section describes how to upgrade TDP from version 3.1.4 to version 3.1.5.

Preparing for the Upgrade

The first stage is to download the upgrade script and the upgrade tarball and prepare the script for upgrade.

Please follow the steps below:

1. Download the TDP 3.1.5 upgrade script (`ambari_upgrade.sh`) from Thales support and copy it to every node in your TDP cluster.
2. Download the tar file (`rpms_315.tar.gz`) from Thales support. This can be placed on an internal server that is external to the TDP cluster nodes.
3. Place the downloaded tar file (`rpms_315.tar.gz`) into `/var/tmp/`. This way, the path for the tar file is `/var/tmp/rpms_315.tar.gz`.
4. Run the `ambari_upgrade.sh` script on each node in the cluster:
 - On the ambari-server select option 1 (ambari-server and ambari-agent)
 - On all other nodes select option 2 (ambari-agent)

NOTE After the above steps, some of the cluster's components might be in a stopped/stale status as indicated by red alert on the left panel or upper right of the page. It is strongly recommended to restart all services to prepare for the upgrade. This can be done using the Ambari server UI by clicking the 3 dots ("...") to the right of "**Services**" > "**Stop All**" then "**Start All**". As an alternative, for each service (HDFS, HBase, Zookeeper and Knox), go to "**Actions**" then click "**Restart All**".

Registering and Installing the Target Version on the Ambari-Server Node

The next step is to register and install the target version on the ambari-server node only. The steps are as follows:

1. Log in to the ambari-server UI.
2. Browse to **Cluster Admin > Stack and Versions**.
3. Click the **Versions** tab. You see the version currently running, marked as **Current**.

NOTE The full version depends on the HDP version you are actually running. For example, if you are currently running the HDP 3.1.4.0 release, you would see something like HDP-3.1.4.0-315 as the full version number.

4. Click "**Manage Versions**".
5. Proceed to register a new version by clicking "**Register Version**".
6. Put **5.0** in place of Version Number so the Name will look like HDP-3.1.5.0.
7. Check "**Use local repository**" and put the Base URL as `file:///var/repo/hdp-315` in all 3 fields. Click **Save**.

8. Click the **Dashboard**.
9. Browse from **“Cluster Admin”** > **“Stack and Versions”**.
10. Click the **“Versions”** tab.
11. On a registered target version, click **“Install Packages”** and click **OK** to confirm.

The Install version operation starts. This installs the target version on all hosts in the cluster. You can monitor the progress of the install by clicking the **“Installing link”**. When the installation completes, the **“Upgrade”** button will replace the **“Install Packages”** button.

Performing the Upgrade

At this point, you are ready to perform the upgrade, so follow all of the steps below on the ambari-server node only:

1. Log in to the ambari-server UI.
2. Before starting the upgrade you have to disable the Auto Start Settings in the Cluster:
Admin > Service Auto Start
3. Run **“Service Check”** on HDFS, HBASE and KNOX in **“Actions”** > **“Run Service Check”**.
4. Browse to **Cluster Admin > Stack and Versions**.
5. Click the **Versions** tab. The registered and installed target HDP version displays an **“Upgrade”** button.
6. Click **“Upgrade”** on the target version.

Based on your current HDP version and the target HDP version, Ambari performs a set of prerequisite checks to determine if you can perform a rolling or an express upgrade. Note that if any required checks pop up, you will need to perform the recommended actions until all checks pass. A dialog displays the options available.

7. Select **“Express Upgrade”** or **“Rolling Upgrade”** method.
 - A "Rolling Upgrade" orchestrates the TDP upgrade in an order that is meant to preserve cluster operation and minimize service impact during upgrade. This process has more stringent prerequisites (particularly regarding cluster high availability configuration) and can take longer to complete than an "Express Upgrade".
 - An "Express Upgrade" orchestrates the TDP upgrade in an order that will incur cluster downtime but with less stringent prerequisites.

TIP If there are mission critical applications running, a “Rolling Upgrade” is recommended.

8. Click **“Proceed”**.
9. There may be some manual steps needed at this point. Please follow the instructions given and check the box that says “I have performed the manual steps above” before proceeding.
10. When the upgrade stages complete, Click **“Finalize”** to complete the upgrade process.

TIP After the upgrade is completed, you can confirm the upgrade has succeeded by going to 'Stack and Versions' -> 'Versions'. You will see that the current version is HDP-3.1.5.0 (3.1.5.0-316). On the 'Stack and Versions' -> 'Upgrade History' tab, when you click 'Upgrade' and see the drop-down list, you will also see details of each service's versions before and after upgrade, together with other information about the upgrade.

References

- > https://docs.cloudera.com/HDPDocuments/Ambari-2.7.5.0/bk_ambari-upgrade/content/upgrading_HDP_register_and_install_target_version.html
- > https://docs.cloudera.com/HDPDocuments/Ambari-2.7.5.0/bk_ambari-upgrade/content/upgrading_HDP_perform_the_upgrade.html

Launching the Amazon Instance

NOTE Please contact Thales support to get the Amazon Machine Image (AMI) ID that you should use in your Amazon Web Services (AWS) region.

Installing the Instance

1. Select EC2 from you Amazon Management Console. Select a region, for example “Canada (Central) ca-central-1”.
2. Next, create the instance by clicking **Images > AMIs**.
3. Search for the AMI by “AMI Name” or “AMI ID”, then select the desired AMI.
4. Follow the wizard to launch the instance:
 - a. Choose AMI.

Select the desired AMI from the search list and click “**Launch Instance**”.
 - b. Choose Instance Type.
 - i. Select “t2.xlarge”, if you’re installing a node that will be part of a 5 node cluster, and for a single node select “m4.4xlarge”.
 - ii. Click **Next** [Next: Configure Instance Details].
 - c. Configure Instance.
 - i. Key in the number of instances required for the “**Number of instances**”. For example, 5 for a 5-node cluster and 1 for a single node.
 - ii. For the rest of the values go with the default options, if you are not customizing the instance, and click **Next** [Next: Add Storage].
 - d. Add Storage.

Go with the default values and click **Next** [Next: Add Tags].
 - e. Add Tags.
 - i. Click “**Add Tag**”.
 - ii. Add a “**Key**” and a “**Value**”. For example: “Owner demo-user”.
 - iii. Click **Next** [Next: Configure Security Group].
 - f. Configure Security Group.
 - i. Select “**Create a new security group**”.
 - ii. Change “**Type**” to “All traffic”.
 - iii. If a pre-configured group with all traffic is available, choose that by selecting the previous selection as “Select an existing security group”.
 - iv. Click [**Review and Launch**].
 - g. Review.
 - i. Review all the properties of the instance that you have selected in the wizard. If needed, you can edit them in this step.

- ii. When you are done, click **[Launch]**.
- h. Select an existing key pair or create a new key pair.
 - i. Create a new key by choosing “**Create a new key pair**” from the drop down menu, name the key pair and download the same for connecting to the instance.
 - ii. If the key pair was created prior to launching the AMI, use the same. In this case, select “**Choose an existing key pair**” and select the key by name.
 - iii. Click the acknowledge check box and click **[Launch Instance]**.
- 5. Once your instances are launched, on your EC2 Dashboard you will see the details as below. You can rename your instances - these here are examples, but they won't be the hostnames of your instances.

TIP Note down the Private DNS and Public IP of your instances as these will be required later.

Logging in to the Instance and Additional Configuration

The initial login for the image is:

- > **login:** root
- > **password:** thales123

You will be prompted to change the password after initial login.

1. Change the hostname.

Set a hostname for your Ambari server and nodes of the Hadoop cluster. Change the hostname to the Private DNS. Run this command, for example, as root to set a new hostname:

```
# hostnamectl set-hostname ip-172-31-3-54.ca-central-1.compute.internal
```

Repeat this on all hosts!

2. Continue with the configuration by following the steps in "Configure Ambari" under "Setting Up the Ambari Server" till step 13 . Then Configure CiperTrust Manager to complete the Installation.

Securing Hadoop Configuration

Thales strongly recommends appropriately configuring and securing your Hadoop environment according to industry best practices and your organization's security policies.

The following references are not comprehensive, but are intended to provide a starting point for background, tools, and best practices that may be applied to your Hadoop environment:

1. "HDP Security Overview":

https://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.1.5/security-overview/content/hdp_security_overview.html

2. "Introduction to Hadoop Security":

<https://www.bmc.com/blogs/hadoop-security/>

3. "Securing Hadoop: Security Recommendations for Hadoop" (Paper):

<https://securosis.com/blog/securing-hadoop-security-recommendations-for-hadoop-new-paper>

4. "Hadoop Security" (O'Reilly Book):

<https://www.oreilly.com/library/view/hadoop-security/9781491900970/>

Hadoop Backup

At some point, you may want to make a backup of everything in Hadoop that is related to DDC. Such a backup will include the HBase tables and all the generated files (.tar) with the information of the scans (hdfs).

In order to save the DDC data and create a backup you have to perform these two steps (separately):

1. Back up HDFS
2. Back up HBase

Preparing for the Backup

- > Execute the `df -h` command in the hdfs directory to be copied and make sure that there is enough space in the destination location.

NOTE

- > It is not necessary to make the backup in each node, but it is good practice make the backup on the name node.
- > You need to have **root privileges** to make a backup and restore it.
- > To run Export/Import commands, you need to switch to the 'hdfs' user.

Backup Options

There are several options to make and restore a backup:

- > HBASE - Full Shutdown Backup (with stopping of the service)
- > HBASE - Make and restore a snapshot (recommended option)
- > HBASE - Export and Import a table. **WARNING:** This option is not a Clone of the table and inconsistencies could appear
- > HDFS - By using the `distcp` command

For more information about these options, refer to the official Cloudera documentation.

Create HDFS Backup

The best way to create a HDFS backup on a different cluster is to use DistCp. The most common use of DistCp is an inter-cluster copy:

```
hadoop distcp hdfs://nn1:8020/source hdfs://nn2:8020/destination
```

Where:

- > nn1 is the name node where your data is located
- > the source is folder where the .tar files are (that is, the folder indicated in the path field when HDFS is configured in DDC)
- > nn2 is the name node where you want to save your data (nn2 can be the same as nn1)

- > the destination is the folder to which the .tar files will be copied and saved

Example command:

```
hadoop distcp hdfs://sys146116.i.vormetric.com:8020/ddc_demo_83
hdfs://sys146116.i.vormetric.com:8020/ddc_backup/hdfs_dir/ddc_demo_83
```

You can find more information on using DistCp on the official webpage or on the Cloudera blog.

NOTE

- > The destination folder should be created before executing the discp command.
- > Note that DistCp requires absolute paths.
- > These actions are made on ACTIVE NameNode and as 'hdfs' user.

Restore HDFS Backup

To restore a backup you also have to use the DistCP command. Again, the most common use of DistCp is an inter-cluster copy:

```
hadoop distcp hdfs://nn2:8020/destination hdfs://nn1:8020/source
```

Where:

- > nn1 is the name node where your data is located
- > the source is the folder where the .tar files are (that is, the folder indicated in the path field when HDFS is configured in DDC)
- > nn2 is the name node where you want to save your data (nn2 can be the same as nn1)
- > the destination is the folder to which the .tar files will be copied and saved

Example command:

```
hadoop distcp hdfs://sys146116.i.vormetric.com:8020/ddc_backup/hdfs_dir/ddc_demo_83
hdfs://sys146116.i.vormetric.com:8020/ddc_demo_83
```

NOTE

- > If the file already exists, it will be skipped. New files (not yet backed up) will still be there, and deleted files will be restored.
- > If you want to completely restore the folder (and only keep the files that were there when the copy was made), you have to execute a command to delete the files. This action is left to your discretion.
- > These actions must be made on the NameNode and as 'hdfs' user.

Create and Restore HBase Backup

There are different approaches to how to create HBase backups. All of them are described at this link:

<https://blog.cloudera.com/approaches-to-backup-and-disaster-recovery-in-hbase/#export>

Here, we describe the most common scenario - a full backup.

NOTE It is possible to make a complete backup by stopping the services and using the `distcp` command. For more details on a full shutdown backup see the HBase documentation.

Snapshots

If you do not want to stop the services, you can use HBase snapshots by executing the steps as follows:

- > Take a snapshot for each table (hbase doc: Take a Snapshot)
- > Restore the snapshot for the tables that you want (hbase doc: Restore a snapshot)

How to use the HBase snapshot utility

To run these commands, please, open a SSH session on the HBase Master node. Let us start by listing all the tables of the original HBase. Please, note that "DDC_SCHEMA1" is the schema name defined on the PQS configuration (PQS Tab of Hadoop Services in Settings).

```
$ hbase shell
$ hbase(main):001:0> list
TABLE
DDC_SCHEMA1:DATASTORE_SUMMARY_REPORT
DDC_SCHEMA1:DATA_OBJECT_REPORT
DDC_SCHEMA1:SCAN_EXECUTION_REPORT
SYSTEM:CATALOG
SYSTEM:FUNCTION
SYSTEM:LOG
SYSTEM:MUTEX
SYSTEM:SEQUENCE
SYSTEM:STATS
TEST:DATASTORE_SUMMARY_REPORT
TEST:DATA_OBJECT_REPORT
TEST:SCAN_EXECUTION_REPORT
12 row(s)
```

Creating a Backup

Take the snapshot with the command

```
snapshot 'myTable', 'myTableSnapshot-122112'
```

and list the snapshots:

```
$ hbase(main):002:0> snapshot 'DDC_SCHEMA1:DATASTORE_SUMMARY_REPORT', 'myTableSnapshot-
datastore_summary_report'
$ hbase(main):003:0> snapshot 'DDC_SCHEMA1:DATA_OBJECT_REPORT', 'myTableSnapshot-data_object_
report'
$ hbase(main):004:0> snapshot 'DDC_SCHEMA1:SCAN_EXECUTION_REPORT', 'myTableSnapshot-scan_
execution_report'
$ hbase(main):005:0> list_snapshots
SNAPSHOT                                TABLE + CREATION TIME
myTableSnapshot-datastore_summary_report DDC_SCHEMA1:DATASTORE_SUMMARY_REPORT (2020-11-
24 06:36:13 -0800)
myTableSnapshot-data_object_report       DDC_SCHEMA1:DATA_OBJECT_REPORT (2020-11-24
06:37:05 -0800)
myTableSnapshot-scan_execution_report    DDC_SCHEMA1:SCAN_EXECUTION_REPORT (2020-11-24
06:37:12 -0800)
3 row(s)
```

Restoring the Backup

Restore the backup by executing the commands as follows:

```
$ hbase(main):006:0> disable 'DDC_SCHEMA1:DATASTORE_SUMMARY_REPORT'
$ hbase(main):007:0> restore_snapshot 'myTableSnapshot-datastore_summary_report'
$ hbase(main):008:0> enable 'DDC_SCHEMA1:DATASTORE_SUMMARY_REPORT'
$ hbase(main):009:0> disable 'DDC_SCHEMA1:DATA_OBJECT_REPORT'
$ hbase(main):010:0> restore_snapshot 'myTableSnapshot-data_object_report'
$ hbase(main):011:0> enable 'DDC_SCHEMA1:DATA_OBJECT_REPORT'
$ hbase(main):012:0> disable 'DDC_SCHEMA1:SCAN_EXECUTION_REPORT'
$ hbase(main):013:0> restore_snapshot 'myTableSnapshot-scan_execution_report'
$ hbase(main):014:0> enable 'DDC_SCHEMA1:SCAN_EXECUTION_REPORT'
```

Export/Import the Tables

The HBase tables can be exported to HDFS using "Export" Command. After that, we can use DistCP to store the data somewhere else.

To export the tables related to DDC, it's necessary to know the Schema that we are using in DDC. The tables are the same for each schema:

DATASTORE_SUMMARY_REPORT, DATA_OBJECT_REPORT and SCAN_EXECUTION_REPORT

How to use Export

First of all, let's list all the tables of our origin Hbase:

```
$ hbase shell
$ hbase(main):001:0> list
TABLE
DDC_SCHEMA1:DATASTORE_SUMMARY_REPORT
DDC_SCHEMA1:DATA_OBJECT_REPORT
DDC_SCHEMA1:SCAN_EXECUTION_REPORT
SYSTEM:CATALOG
SYSTEM:FUNCTION
SYSTEM:LOG
SYSTEM:MUTEX
SYSTEM:SEQUENCE
SYSTEM:STATS
TEST:DATASTORE_SUMMARY_REPORT
TEST:DATA_OBJECT_REPORT
TEST:SCAN_EXECUTION_REPORT
12 row(s)
$ hbase(main):001:0> quit
```

Export the tables on a HDFS directory executing hbase export command:

```
$ bin/hbase org.apache.hadoop.hbase.mapreduce.Export <tablename> <outputdir>

$ hbase org.apache.hadoop.hbase.mapreduce.Export DDC_SCHEMA1:DATASTORE_SUMMARY_REPORT
hdfs:///ddc_backup/hbase/ddc_schema1_datastore_summary_report
$ hbase org.apache.hadoop.hbase.mapreduce.Export DDC_SCHEMA1:DATA_OBJECT_REPORT hdfs:///ddc_
backup/hbase/ddc_schema1_data_object_report
$ hbase org.apache.hadoop.hbase.mapreduce.Export DDC_SCHEMA1:SCAN_EXECUTION_REPORT hdfs:///ddc_
backup/hbase/ddc_schema1_scan_exectuion_report
```

NOTE The output directory must not exist.

How to use Import

If you want to restore the same tables in the same schema, you can execute the hbase import command.

Import the tables from the HDFS directory where the previous export is. To do that, execute the hbase import command:

```
$ bin/hbase org.apache.hadoop.hbase.mapreduce.Import <tablename> <inputdir>

$ hbase org.apache.hadoop.hbase.mapreduce.Import DDC_SCHEMA1:DATASTORE_SUMMARY_REPORT
hdfs:///ddc_backup/hbase/ddc_schema1_datastore_summary_report
$ hbase org.apache.hadoop.hbase.mapreduce.Import DDC_SCHEMA1:DATA_OBJECT_REPORT hdfs:///ddc_
backup/hbase/ddc_schema1_data_object_report
$ hbase org.apache.hadoop.hbase.mapreduce.Import DDC_SCHEMA1:SCAN_EXECUTION_REPORT hdfs:///ddc_
backup/hbase/ddc_schema1_scan_exectuion_report
```

NOTE

- > The import utility replaces the existing rows, but it does not clone the table keeping the new rows after the export.
- > There may be inconsistencies in the data, especially in the latest reports.

Thales CipherTrust Data Discovery and Classification

DEPLOYMENT GUIDE



Document Information

Document Information

Product Version	2.2.0
Document Number	007-000727-002, Rev A
Release Date	17 February 2021

Trademarks

Thales CipherTrust Data Discovery and Classification is powered by Groundlabs.

All intellectual property is protected by copyright. All trademarks and product names used or referred to are the copyright of their respective owners. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, chemical, photocopy, recording or otherwise without the prior written permission of Thales.

Disclaimer

Thales makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Thales reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon Thales to notify any person or organization of any such revisions or changes.

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

You are responsible for ensuring your own compliance with various laws and regulations, including but not limited to any data privacy or data protection regulation. You are solely responsible for obtaining advice from competent legal counsel to assist you in the identification and interpretation of any relevant laws and regulations that may affect your business and the implementation of any actions you may need to take to ensure you meet your compliance obligations with respect to such laws and regulations.

The software, the products, services, and any other capabilities described or provided herein are not suitable for all situations and may have restricted availability or applicability. Thales does not provide legal, accounting, or auditing advice, nor does it represent or warrant that its software, services, or products will ensure that you are in compliance with any law or regulation.

Thales invites constructive comments on the contents of this document. Send your comments, together with your personal and/or company details to the address below.

Contact Method	Contact Information	
Address	Thales 4690 Millennium Drive Belcamp, Maryland 21017 USA	
Phone	US	1-800-545-6608
	International	1-410-931-7520
Email	technical.support.DIS@thalesgroup.com	
Technical Support Customer Portal	https://supportportal.thalesgroup.com Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the Thales Knowledge Base.	

CONTENTS

Document Information	2
About this Document	6
Audience	6
What's in This Guide	6
Organization	6
Document Conventions	7
Hyperlinks	7
Notifications	7
Command Syntax and Typeface Conventions	8
Related Documents	9
DDC Deployment Architecture	10
Supported Data Stores	10
Where to Install the DDC Agents	11
How DDC Uses Hadoop	11
Software and Hardware Requirements	12
Hardware Requirements	12
Software Requirements	12
Ports Used for Communication	12
Deployment Prerequisites	15
Installing CipherTrust Manager	15
Installing and Configuring Hadoop	15
Configuring DNS Connectivity	16
Running DDC in a Cluster	17
Deploying DDC into a New Environment	17
Deploying DDC into an Existing Cluster	17
Assigning the Active DDC Node	18
Identifying the Active DDC Node	18
Configuring CipherTrust Manager	20
Configuring HBase	20
Configuring HDFS	22
Agent Configurations	24
Agent Compatibility and Installers	24
Installing Agents	26

Installing Agents on RHEL	26
Installing Agents on Debian	27
Installing Agents on Windows	27
Uninstalling Agents	29
Uninstalling Agents from RHEL	29
Uninstalling Agents from Debian	29
Uninstalling Agents from Windows	29
Upgrading Agents	30
Hardening the Deployment	31
General Hardening Recommendations	31
Certificate Security Recommendations	31
Securing the Hadoop Configuration	32

About this Document

This introductory section identifies the audience, provides a brief summary of the contents of this guide, and discusses the documentation conventions used. It contains the following sections:

- > ["Audience" below](#)
- > ["What's in This Guide" below](#)
- > ["Organization" below](#)
- > ["Document Conventions" on the next page](#)

Audience

This document is intended for Thales CipherTrust Data Discovery and Classification (DDC) users responsible for classification of data discovered on data stores. It is assumed that the users of this document are proficient with security and data discovery concepts.

All products manufactured and distributed by Thales are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.

Thales designs data security products for use by file server administrators, network administrators, security engineers, database administrators, application developers, and other technology professionals responsible for daily operations in support of data security.

What's in This Guide

This guide provides information on deploying the Thales CipherTrust Data Discovery and Classification (Thales DDC) solution. The document describes the requirements that must be met before installing and configuring the DDC Agents on host machines. The document also provides list of compatible installers for different types of data stores. Finally, the document provides guidelines on enhancing the performance and security of the DDC solution. The DDC documentation does not include information that is already covered in the CipherTrust Manager documentation, available through the **Product Documentation** link on the CipherTrust Manager GUI login page.

Organization

The *Thales CipherTrust Data Discovery and Classification Deployment Guide* contains the following sections:

1. ["DDC Deployment Architecture" on page 10](#)
Contains a visual overview (a diagram) of a typical DDC deployment.
2. ["Software and Hardware Requirements" on page 12](#)
Software and hardware Requirements for the DDC Server and Agents.
3. ["Deployment Prerequisites" on page 15](#)
Lists requirements that must be fulfilled before you start the DDC deployment.

4. ["Running DDC in a Cluster" on page 17](#)

Information and considerations for deploying DDC in a CipherTrust Manager cluster.

5. ["Configuring CipherTrust Manager" on page 20](#)

Provides instructions to configure DDC so that it can use Hadoop for storing its data.

6. ["Agent Configurations" on page 24](#)

The section lists all supported installers for different types of data stores.

7. ["Installing Agents" on page 26](#)

Provides instructions to install DDC Agents on Windows and Linux hosts.

8. ["Hardening the Deployment" on page 31](#)

Provides guidelines to enhance the performance and security of your DDC deployment.

9. ["Uninstalling Agents" on page 29](#)

Provides instructions to uninstall DDC Agents from Windows and Linux hosts.

10. ["Upgrading Agents" on page 30](#)

Instructions for upgrading DDC Agents on supported platforms.

Document Conventions

This section describes the formatting conventions used in this user guide to indicate hyperlinks, special notes, important information, tips, and warnings.

Hyperlinks

Hyperlinked text will, by default, appear in the shade of purple.

For example: All technical document templates can be found on the [Technical Writing Community page](#).

Notifications

This user guide uses notes, tips, and warnings to alert you to important information that may help you to complete your task, or prevent personal injury, damage to the equipment, or data loss.

Notes

Notes are used to alert you to important or helpful information. These elements use the following format:

NOTE Take note. Notes contain important or helpful information that you want to make stand out to the user.

Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss. These elements use the following format:

CAUTION! Exercise caution. Caution alerts contain important information that may help prevent unexpected results or data loss.

Warnings

Warnings are used to alert you to the potential for catastrophic data loss or personal injury. These elements use the following format:

****WARNING**** Be extremely careful and obey all safety and security measures. In this situation you might do something that could result in catastrophic data loss or personal injury.

Command Syntax and Typeface Conventions

Convention	Description
bold	The bold attribute is used to indicate the following: > Button names (Click Save As.) > Check box and radio button names (Select the Print Duplex check box.) > Dialog box titles (On the Protect Document dialog box, click Yes.) > Field names (User Name: Enter the name of the user.) > Menu names (On the File menu, click Save.) (Click Menu > Go To > Folders.) > User input (In the Date box, type April 1.)
<i>italic</i>	The italic attribute is used for emphasis or to indicate a related document. (See the <i>Thales CipherTrust Data Discovery and Classification Customer Release Notes</i> for more information.)
Double quote marks	Double quote marks enclose references to other sections within the document. For example: Refer to "Disclaimer" on page 2.
<variable>	In command descriptions, angle brackets represent variables. You must substitute a value for command line arguments that are enclosed in angle brackets.
<pre>[optional] [<optional>]</pre>	Square brackets enclose optional keywords or <variables> in a command line description. Optionally enter the keyword or <variable> that is enclosed in square brackets, if it is necessary or desirable to complete the task.
<pre>[a b c] [<a> <c>]</pre>	Square brackets enclose optional alternate keywords or variables in a command line description. Choose one command line argument enclosed within the braces, if desired. Choices are separated by vertical (OR) bars.
<pre>{ a b c } { <a> <c> }</pre>	Braces enclose required alternate keywords or <variables> in a command line description. You must choose one command line argument enclosed within the braces. Choices are separated by vertical (OR) bars.

Related Documents

The following documents contain related or additional information:

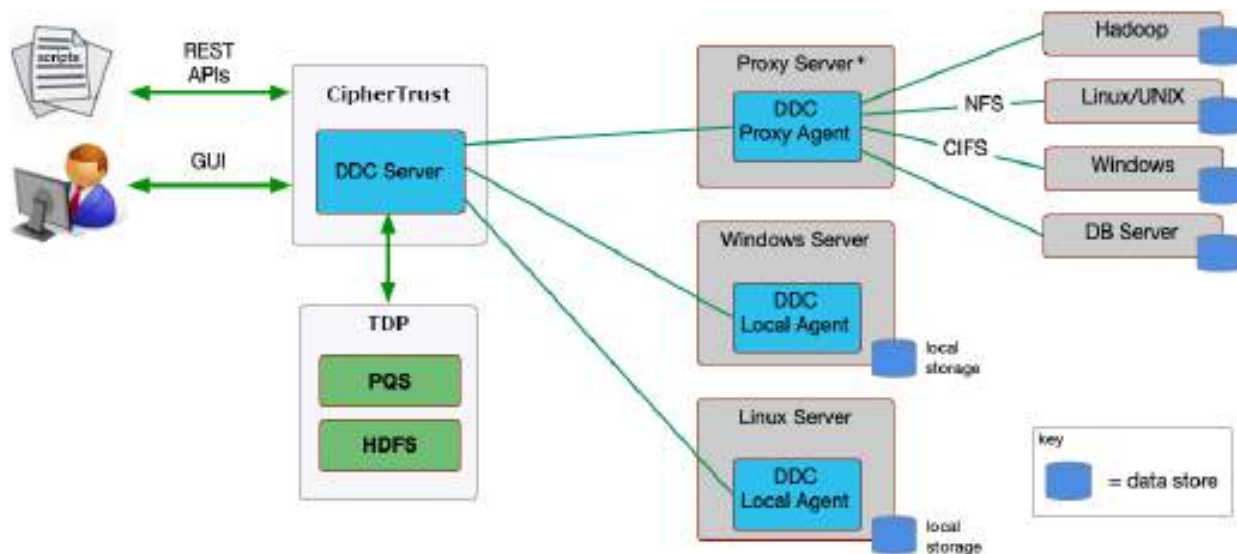
- > *Thales CipherTrust Data Discovery and Classification Administrator Guide*
- > *Thales Data Platform Installation Guide*
- > *Thales CipherTrust Data Discovery and Classification Customer Release Notes*

You can view or download the latest version of the CRN for this release at this location:

<https://supportportal.thalesgroup.com>

DDC Deployment Architecture

This section describes the main components of Thales CipherTrust Data Discovery and Classification (DDC) and how they operate together to provide the DDC solution. Before you go ahead with the actual deployment, review the graphic included in this section to get a feel for what a typical DDC deployment looks like. The concepts used in this diagram are introduced in the later sections of this document and explained at length in the "DDC Administration Guide".



* A Windows Proxy is needed to connect to databases.

At the heart of the DDC solution is CipherTrust Manager on which runs the DDC Server. It is from here that users interact with the DDC GUI or use the DDC APIs to create classification profiles, add data stores, launch scans and generate reports.

Supported Data Stores

DDC supports a number of different data stores:

- > NFS shares
- > CIFS shares
- > Databases
- > Hadoop storage
- > Local storage (Windows and Linux)

In order to access these data stores, the DDC Server communicates with one or more DDC Agents. The DDC Agent is a software component that is used to scan a data store for Infotypes (such as credit card numbers, email addresses and so on) that are part of a classification profile. All data that is collected is sent from the Agent to the DDC Server which stores the data, together with any user requested reports, on an external Hadoop cluster.

Where to Install the DDC Agents

Generally speaking, if you are scanning data stores that are local to Windows or Linux (no network shares), you should install the DDC Agent on the server where the data is located. For all other types of storage (top part of the figure), the DDC Agent should be installed on a proxy server.

As an example, let's assume that you wish to scan an NFS share. In this case, the NFS share should be mounted on the proxy server and the DDC Agent should be installed on the proxy server. To scan the share, specify the mount point of the NFS share when creating the scan. For DDC Agent requirements and the types of data stores supported, see ["Agent Configurations" on page 24](#). For information on securing the deployment, refer to ["Hardening the Deployment" on page 31](#).

How DDC Uses Hadoop

DDC uses Hadoop to generate reports from scans and to store their results (report data). Thales Data Platform (TDP) is the only Hadoop flavor currently supported for this purpose. This is different than the Hadoop cluster that DDC also supports as a data store, that is where the user stores the data.

DDC can directly query HDFS but it requires Phoenix Query Server (PQS) to interface with Hadoop's HBase to benefit from its multitenancy features.

The DDC server retrieves the results of the scan from the DDC Agent and stores this information in TDP together with any reports that are generated. It is imperative that your TDP cluster is highly available to avoid losing any data store scans or reports.

DDC also requires Apache Knox as a single point of access to the TDP cluster (both HBase and HDFS), to ensure all the communications get protected with TLS, and for authentication. Therefore, you only need to connect up DDC to Knox. For information on configuring DDC to use TDP, see ["Configuring CipherTrust Manager" on page 20](#).

Software and Hardware Requirements

Hardware Requirements

CipherTrust Manager requirements

DDC is only supported when running CipherTrust Manager as a Virtual Machine. The CipherTrust Manager VM has the following requirements:

- > **RAM:** 16 GB minimum
- > **CPU:** 4 cores. It is recommended to add extra cores if the average CPU usage is above 50% or CPU load is above 80% for extended periods of time.
- > **Disk space:** at least 256 GB

Agent requirements

Each concurrent DDC scan requires one core and typically less than 1 GB of RAM. Agents do not launch concurrent Local Storage scans. When running Local scans Linux agents require a minimum of 1 core and 1 GB of RAM and Windows agents require 2 cores and 4 GB of RAM.

The above requirements only take the DDC scanning agent requirements into account. The operating system requires additional resources, usually 1-2 cores and 2-4 GB of RAM, and consider the requirements of the other services.

Please note that an agent running in a server can behave as a Local proxy for scanning this server and as a Proxy agent to scan other Data Stores, so you should monitor the agent resource consumption while the scans are running if needed.

Software Requirements

DDC Agents for Debian require Debian kernel versions 3.x and higher.

Ports Used for Communication

This section provides a list of ports that should allow communication among agents, data stores, and CipherTrust Manager. Firewalls should be configured to allow this communication.

The following table lists the ports that are used by agents to connect to data stores:

Initiator	Receiver	Protocol	Port(s)	Connection Type	Description
Agents	CipherTrust Manager	TCP	11117	Persistent	Allow traffic between Agents and the CipherTrust Manager appliance. Agents initiate the communication and keep persistent connections.

Initiator	Receiver	Protocol	Port(s)	Connection Type	Description
Agents	IBM DB2	TCP	50000	Non-persistent	Allow traffic between Agents and the IBM DB2 database store. Agents initiate the communication and need the port during the current session.
Agents	Microsoft SQL	TCP	1433	Non-persistent	Allow traffic between Agents and the Microsoft SQL database store. Agents initiate the communication and need the port during the current session.
Agents	Oracle	TCP	1521	Non-persistent	Allow traffic between Agents and the Oracle database store. Agents initiate the communication and need the port during the current session.
Agents	CIFS/SMB server	TCP	445*	Non-persistent	Allows scanning of Windows remote CIFS file shares.
Agents	NFS server	TCP or UDP	2049**	Non-persistent	Allows scanning of NFS file shares.
Agents	Hadoop Scanning	TCP	8020, 50075 and 50010	Non-persistent	Allow traffic between Agents and Hadoop cluster nodes. Agents initiate the communication and need the ports during the current session.

Apart from Hadoop as data store, CipherTrust Manager uses Hadoop as an external database to store and process the scan results. CipherTrust Manager initiates the communication and needs these ports to be open during the current session:

Initiator	Receiver	Protocol	Port(s)	Connection Type	Description
CipherTrust Manager	Hadoop***	TCP	8443 and 8765	Non-persistent	Allow traffic between TDP cluster nodes and the CipherTrust Manager appliance. DDC supports Hadoop NameNodes and Apache Knox.

* Additional ports. For Windows 2000 and older:

- 137 (UDP)
- 138 (UDP)
- 139 (TCP)

** NFSv4 requires only port 2049 (TCP only). NFSv3 and older must allow connections on the following ports:

- 111 (TCP or UDP)
- Dynamic ports assigned by rpcbind.

*** Thales Data Platform (TDP) is the only Hadoop flavor currently supported.

Deployment Prerequisites

- > CipherTrust Manager must be installed, configured, and accessible through the GUI (also called the console).
- > Hadoop must be installed and configured with a Phoenix Query Server (PQS) mapped to HBase.
- > You must also have Apache Knox installed and configured for Hadoop.

Knox must also be **DNS addressable**, through a network DNS or by adding the DNS entry as described in the "CipherTrust Manager Administration Guide" ("Concepts" > "DNS Hosts" > "Configuring DNS Hosts" section).

Installing CipherTrust Manager

DDC is shipped as a module of CipherTrust Manager with a trial license already installed so no additional installation should be required in CipherTrust Manager. If you don't have CipherTrust Manager already installed, or you cannot find DDC in the list of installed licenses contact Thales Group or refer to the CipherTrust Manager product documentation for instructions.

Installing and Configuring Hadoop

Thales Data Platform 3.1.5 (TDP) is a Big Data platform based on Hadoop technology. We recommend running a **5 node** TDP 3.1.5 cluster that has the following services available:

- > HDFS
- > HBase
- > Phoenix Query Server (PQS) - available on at least one node
- > Knox

We recommend **2 name nodes** and **3 data nodes**. Each node should have the following minimum hardware configuration:

- > 8 CPUs / vCPUS
- > 16 GB RAM
- > 100 GB of disk

For installing TDP 3.1.5 refer to the "Thales Data Platform Installation Guide" and perform all the steps in there before continuing with the DDC installation.

For information about Hadoop, refer to the official HDP documentation page:

<http://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.1.4/index.html>

Configuring DNS Connectivity

To ensure uninterrupted connectivity to the UI, it is recommended to create a DNS entry for the active CipherTrust Manager node and use this entry when configuring the Agents and accessing the CipherTrust Manager UI. This way the Agents will be able to reach CipherTrust Manager even if its IP address changes.

- > **Standalone CipherTrust Manager:** If the CipherTrust Manager appliance is standalone (not in a cluster), configure a DNS entry with its IP address.
- > **Clustered CipherTrust Manager:** In a CipherTrust Manager cluster, configure a DNS entry with the IP address of the cluster's "active" node. See ["Identifying the Active DDC Node" on page 18](#) for details.

Running DDC in a Cluster

Although running a DDC cluster is currently not supported, Agents can be deployed into a CipherTrust Manager cluster. In this configuration, one of the CipherTrust Manager nodes, has to be assigned the active DDC node, so all the Agents report to it. This DDC configuration does not support failover, so if the active node fails so will DDC. DDC will not work, as long as the active node is down, so the only way to retain the DDC operations without any data loss is to restore the original active node.

There are two possible scenarios for running DDC in a cluster:

- > **"Greenfield" deployment** - in this scenario DDC is deployed into a completely new environment. See ["Deploying DDC into a New Environment" below](#) for more information.
- > **"Brownfield" deployment** - in this scenario DDC is deployed into an existing CipherTrust Manager cluster. See ["Deploying DDC into an Existing Cluster" below](#) for more information.

In both these scenarios it is essential to identify the active DDC node as you need an active DDC node for DDC to function in a cluster. For information on assigning the active node and identifying the IP address/hostname of the active node, refer to ["Assigning the Active DDC Node" on the next page](#) and ["Identifying the Active DDC Node" on the next page](#).

Deploying DDC into a New Environment

If you are deploying DDC into an environment where there is no existing CipherTrust Manager server that you could connect the Agents to, you have to start by deploying CipherTrust Manager first as you need a running, active CipherTrust Manager server to complete Agents' installation and configuration. If you're considering having a CipherTrust Manager cluster at some point, and using the cluster with DDC, you need to plan your deployment ahead and think it out carefully.

It is strongly recommended to create any planned cluster of CipherTrust Manager nodes prior to deploying DDC into it. Creating a CipherTrust Manager cluster out of CipherTrust Manager servers already hosting DDC poses a risk of a **loss of data** collected by DDC prior to the cluster creation. This is because any new CipherTrust Manager node added to the cluster gets its DDC **database wiped out** and replaced with a copy of the database of the active node.

This scenario may prove destructive for cases, for example, when you initially planned to have a few independent DDC systems (for example, for monitoring different segments of the network) with unclustered CipherTrust Manager servers but then decided to form a cluster out of them. In such a case, all CipherTrust Manager servers that also act as DDC servers will lose all their DDC data and settings except for the active node.

Deploying DDC into an Existing Cluster

Deploying DDC into an existing CipherTrust Manager cluster is a relatively straightforward task and it should not pose huge problems. After the installation, you just have to make one of the CipherTrust Manager nodes the active DDC node. Next, you have to connect up all the Agents to that active DDC node to complete their configuration.

From then on, the active DDC node will store all the configuration settings in its database. The database of the active node gets replicated over all cluster nodes, so every cluster member has an identical copy of the database. All the copies get synchronized and updated every time new data is inserted into the copy on the active node.

Assigning the Active DDC Node

DDC always requires one active node regardless of the number of CipherTrust Manager nodes (whether it is a single CipherTrust Manager node or a cluster of two or more CipherTrust Manager nodes). It is this active node that the DDC Agents point to (via DNS) and it is only this node that will respond to GUI operations.

To create an active DDC node you have to " earmark" one of the CipherTrust Manager nodes as the active DDC node. This is a manual procedure, which can be performed either through the CipherTrust Manager UI or the command line. In both cases, you have to have the CipherTrust Manager admin rights. The assignment of the active DDC node does not affect the normal CipherTrust Manager cluster operation.

NOTE The assignment of a DDC active node cannot be undone!

- > To assign the active DDC node by using the CipherTrust Manager UI, follow this procedure:
 - a. Log in to the CipherTrust Manager node that you want to make the active DDC node.
 - b. Click the **Data Discovery and Classification** link to open the DDC app.
You should see the "The current node is inactive. The node must be activated to use DDC." message.
 - c. Click the **Activate** button below the message.
The CipherTrust Manager node becomes the active DDC node.
- > To assign the active DDC node by using the CipherTrust Manager command line, you need the **ksctl** tool installed and configured. For information on installing and configuring the tool, refer to the "Interfaces > CLI" section in the "DDC Administration Guide".

Connect to the CipherTrust Manager node that you want to make the active DDC node, and issue this command:

```
ksctl ddc active-node register
```

After you assign an active DDC node, you can perform all DDC related tasks through that node. The other nodes - non-active nodes - will not allow you to work with DDC. When you log in to a CipherTrust Manager node that is a non-active DDC node and enter the **Data Discovery and Classification** application, you will see this message displayed:

"You are currently connected to an inactive node. You must switch to the active node <**active DDC node IP address**> to run DDC."

Identifying the Active DDC Node

In this section, we show how to locate the active node. For this procedure you need the **ksctl** tool. The tool must be configured to access any of the cluster nodes. For detailed information on installing and configuring the **ksctl** tool, refer to the "Interfaces > CLI" section in the "DDC Administration Guide".

- > To find the IP address of the active CipherTrust Manager node, run this command:

```
ksctl ddc active-node info
```

The output shows the IP address of the cluster's active node. Use this IP address to configure a DNS entry for the active CipherTrust Manager. Use that DNS entry to configure the Agents and access the CipherTrust Manager UI.

```
{
  "public_address": "mycluster.thalesgroup.com",
  "host": "10.45.102.101"
}
```

If the CipherTrust Manager appliance is not in a cluster, the command returns the following error:

```
{
  "code": 15,
  "codeDesc": "NCERRBadRequest: Bad HTTP request",
  "message": "oleander is not in cluster mode"
}
```

In this case, just use the IP address (or DNS entry) of this single CipherTrust Manager node.

Configuring CipherTrust Manager

To configure DDC with Hadoop, perform the following steps:

1. Type the CipherTrust Manager URL in the browser and log on to the CipherTrust Manager console as the administrator.
2. Click the **Data Discovery** link to open the **Data Discovery** configuration screen.
3. In the sidebar on the left, click **Settings > Hadoop Services**. The **Hadoop Services** page is displayed. This page contains two tabs, **PQS** and **HDFS**. By default, **PQS** is the active tab.

NOTE Only the users with access to the **root domain** have access to and can modify the Hadoop Services configuration. For more information about domains, refer to the "Thales CipherTrust Manager Administrator Guide".

****WARNING**** Once configured, the Hadoop Services settings must not be modified or you will lose access to all data.

Hadoop Services

PQS **HDFS**

CONNECTION

Knox Gateway Hostname* Port*

URL*

Scheme*

Knox Gateway Server Certificate

No file selected

KNOX AUTHENTICATION

Username*

Password*

Click the lock icon to change the password

Once configured, the Services must not be modified or you will lose access to all your collected data. Make sure the node meets the system requirements for DDC.

Configuring HBase

1. To configure DDC for HBase, click the **PQS** tab and configure the PQS settings:
 - a. **Hostname** and **Port** - the connection details of the Knox server.

You must use the hostname of your Knox server not its IP address as the server certificate that you need to import later on is hostname based. The default port is **8443**.

Knox must also be **DNS addressable**, through a network DNS or by adding the DNS entry as described in the "CipherTrust Manager Administration Guide" ("Concepts" > "DNS Hosts" > "Configuring DNS Hosts" section).

- b. URI** - the PQS path as configured in Knox. The path consists of the Knox server bit and the PQS bit. Here is an example:

/gateway/default/avatica

where "gateway" is the Knox bit, "default" is the topology name, and "avatica" is the service name.

NOTE If you are not using the default topology, use your topology name instead of the "default" bit in the URI.

- c. Schema** - the PQS Schema as mapped to HBase namespace.

TIP To avoid potential issues, it is recommended to have this schema already created prior to performing this configuration step. Refer to the "Thales Data Platform Installation Guide", section "12. Creating the PQS Schema".

- d. Server Certificate** - Knox uses HTTPS for connection that is why DDC will only connect to the Hadoop Services if a valid and trusted certificate is used. If this certificate was signed by a public and recognized CA, DDC will trust it automatically, otherwise you should manually import it here.

Click the **Choose File** button in the **Select Server Certificate** section and import the server certificate.

IMPORTANT: Use the same certificate as you exported earlier in "13. Exporting the Knox Server Certificate" in the "Thales Data Platform Installation Guide".

TIP

- > If the certificate is self-signed, you have to export it first on the Knox server and then import on the machine where you are running the CipherTrust Manager console. The certificate is then permanently stored in DDC. You do not need to repeat this step on another DDC client machine.
- > Binary format certificates are not accepted. You can only import a plain-text certificate (such as for example, the Base-64 encoded X.509 standard).

- e.** In the **AUTHENTICATION** section fill in the **Username** and **Password** string for a Knox authorized user.

IMPORTANT: Use the same authentication details as you configured earlier in "9. Configuring Knox" in the "Thales Data Platform Installation Guide".

- 2.** Click **Save Changes** to save the configuration.

Configuring HDFS

1. To configure DDC for HDFS, click the **HDFS** tab, and configure the HDFS settings:

a. **Hostname** and **Port** - the connection details of the Knox server.

You must use the hostname of your Knox server not its IP address as the server certificate that you need to import later on is hostname based. The default port is **8443**.

Knox must also be **DNS addressable**, through a network DNS or by adding the DNS entry as described in the "CipherTrust Manager Administration Guide" ("Concepts" > "DNS Hosts" > "Configuring DNS Hosts" section).

b. **URI** - the path to HDFS as configured in Knox. The path consists of the Knox server bit and the HDFS bit. For example:

/gateway/default/webhdfs/v1

where "gateway" is the Knox bit, "default" is the topology name, and "webhdfs/v1" is the HDFS bit.

NOTE If you are not using the default topology, use your topology name instead of the "default" bit in the URI.

c. **Folder** - type in the DDC file system directory in HDFS that you created earlier (for example **/ciphertrust_ddc**).

IMPORTANT: Use the same directory name as you configured earlier in "11. Creating DDC Directory Under HDFS" in the "Thales Data Platform Installation Guide".

d. **Server Certificate** - as HTTPS is forced as the protocol for Knox you have to import the server certificate here.

Click the **Choose File** button in the **Select Server Certificate** section and import the server certificate.

IMPORTANT: Use the same certificate as you exported earlier in "13. Exporting the Knox Server Certificate" in the "Thales Data Platform Installation Guide".

NOTE

- > The certificate is typically self-signed so you have to export it first on the Knox server and then import on the machine where you are running the CipherTrust Manager console. The certificate is then permanently stored in DDC. You do not need to repeat this step on another DDC client machine.
- > Binary format certificates are not accepted. You can only import a plain-text certificate (such as for example, the Base-64 encoded X.509 standard).

- e. In the **AUTHENTICATION** section fill in the **Username** and **Password** string for a Knox authorized user.

IMPORTANT: Use the same authentication details as you configured earlier in "9. Configuring Knox" in the "Thales Data Platform Installation Guide".

2. Click **Save Changes** to save the configuration.

The connection was successful if no error is returned and this message is displayed:

"Success. HDFS settings have been updated"

Agent Configurations

DDC supports two types of Agent configurations:

- > **Local:** Agent is installed and configured directly on the machine that contains sensitive data.
- > **Proxy:** Agent is installed and configured on a proxy machine that is used to scan sensitive data on other machines.

The instructions to install and configure Agents in both types of configurations are the same.

Agent Compatibility and Installers

The following table lists supported Agent installers for different types of data stores for different platforms and databases to help you select an appropriate installer depending on your data store requirements.

Data Store Category	Data Store Type	Agent Configuration	Agent Installer Packages
Local Storage	RHEL, CentOS	Local	RHEL (.rpm) - 32 bit, 64 bit
	Debian based distros	Local	Debian (.deb) - 32 bit, 64 bit, 64 bit with database support (compatible with Debian 9 only)
	Windows	Local	Windows (.msi) - 32 bit, 64 bit, 32 bit with database support, 64 bit with database support
Database Storage	IBM DB2 11.1 and higher	Local, Proxy	Windows (.msi) - 32 bit with database support, 64 bit with database support
	Microsoft SQL 2005 and higher	Local, Proxy	Windows (.msi) - 32 bit with database support, 64 bit with database support
	Oracle 9 and higher	Local, Proxy	Windows (.msi) - 32 bit with database support, 64 bit with database support
Network Storage	Unix File Share (NFS)	Proxy	Debian (.deb) - 32 bit, 64 bit, 64 bit with database support (compatible with Debian 9 only) RHEL (.rpm) - 32 bit, 64 bit
	Windows Share (SMB, CIFS)	Proxy	Windows (.msi) - 32 bit, 64 bit, 32 bit with database support, 64 bit with database support
Big Data	Hadoop Cluster	Proxy	Debian (.deb) 64 bit with database support (compatible with Ubuntu 18 only)*

* Linux 3 64-bit "database runtime" version with additional packages for use with Hadoop Clusters only. For Hadoop as data store, Agents must have the following packages installed:

- libaio1

- libgsasl7
- libxml2
- libprotobuf10

Installing Agents

This section provides procedures for installing Agents for all supported operating systems.

1. Download the latest version of a compatible Agent installation file from the [Thales Customer Support Portal](#). The "[Agent Compatibility and Installers](#)" on page 24 lists all the data stores and their matching installer packages.
2. Save the downloaded installer on the host machine where you want to install the Agent.
3. Follow the appropriate installation procedure for your operating system.

TIP Before you begin the installation, make sure that CipherTrust Manager is reachable from the host where you are installing the Agent.

Installing Agents on RHEL

1. Navigate to the location where the Agent installation package (.rpm) is stored.
2. Log in as a user with root privileges (su) and Install the Agent by using the following command:

```
#rpm -ivh er2-2.x.xx-xxxx.rpm
```

For example:

```
#rpm -ivh er2-2.0.31-linux26-rh-x64.rpm
```

The package name that you use with the command may be different and depends on your system's architecture and Agent type.

3. Connect the Agent to the active CipherTrust Manager node:

```
#er2-config -i <hostname|ip_address>
```

where, <hostname|ip_address> represents the IP address or hostname of the CipherTrust Manager node.

4. Test the connection settings (on the data store that is using this host).

```
#er2-config -t
```

If the connection has been correctly configured, you should see the following message:

```
Testing connection setting...
Test SUCCESS. Saving settings
Configuration updated, please restart agent service
The configuration has been saved. Please restart the agent for the changes to take effect.
```

5. Restart the Agent:

- Option 1

```
#!/etc/init.d/er2-agent restart
```

- Option 2

```
#!/etc/init.d/er2-agent stop
#!/etc/init.d/er2-agent start
```

NOTE The installation script creates an **erecon** user in the **erecon** group. Please ensure that this user (or group) is able to read all the files to scan. For security reasons, the account has its password **locked** to ensure that the user is solely used by the DDC scanning agent.

Installing Agents on Debian

1. Navigate to the location where the Agent installation (.deb) package is stored.

2. Install the Agent by using the following command:

```
sudo dpkg -i er2_2.x.xx-xxxx.deb
```

For example:

```
sudo dpkg -i er2_2.0.31-linux26-x64.deb
```

The package name that you use with the command may be different and depends on your system's architecture and Agent type.

3. Connect the Agent to the active CipherTrust Manager node:

```
sudo er2-config -i <hostname|ip_address>
```

where, <hostname|ip_address> represents the IP address or hostname of the CipherTrust Manager node.

4. Test the connection settings (on the data store that is using this host).

```
sudo er2-config -t
```

If the connection has been correctly configured, you should see the following message:

```
Testing connection setting...
Test SUCCESS. Saving settings
Configuration updated, please restart agent service
The configuration has been saved. Please restart the agent for the changes to take effect.
```

5. Restart the Agent:

- Option 1

```
sudo /etc/init.d/er2-agent restart
```

- Option 2

```
sudo /etc/init.d/er2-agent stop
sudo /etc/init.d/er2-agent start
```

NOTE The installation script creates an **erecon** user in the **erecon** group. Please ensure that this user (or group) is able to read all the files to scan. For security reasons, the account has its password **locked** to ensure that the user is solely used by the DDC scanning agent.

Installing Agents on Windows

1. Log in to the host machine where you want to install the Agent as administrator.

2. Run the Agent installer.

3. In the **Welcome** screen of the setup wizard, click **Next** to continue.

4. The **End-User Licence Agreement** (EULA) screen is displayed.
 - a. Read the license agreement and select **I accept the terms in the Licence Agreement**.
 - b. Click **Next** to continue.
5. In the **Choose Setup Type** screen, select the **Install** option for the standard installation and click **Next** to continue.
6. The **Ready to Install** screen is displayed.
 - a. Click **Install** to install the product in the default location.
 - b. If the **User Access Control** dialog box appears, click **Yes** to confirm.

The installation begins and the progress is shown under the **Status** progress bar.
7. During the installation, in a separate **Node Configuration** window, you are asked for the connection details of the active CipherTrust Manager node.
 - a. **Master server IP address or host name** - specify the IP address or host name of the CipherTrust Manager node.
 - b. **Master server public key** and **Target Group** - skip this configuration part as it is optional and currently not used.
 - c. Click **Test Connection** to test the connection between the Agent and CipherTrust Manager.
 - If the connection is properly configured, a confirmation will appear stating "Connectivity test is successful". Click **OK** to close the prompt.
 - If the connectivity test fails, click **OK** to close the prompt, make sure that CipherTrust Manager is reachable from the Agent host, and retry the test.
 - d. Click **Finish** to complete the configuration.
8. After a successful Agent installation, click the **Finish** button to exit the wizard and complete the installation.

NOTE The installer creates a service called **Enterprise Recon 2 Agent** that runs under the Local System user account.

Uninstalling Agents

Uninstalling Agents from RHEL

To uninstall a DDC Agent:

1. Stop the DDC Agent.

```
/etc/init.d/er2-agent stop
```

2. Uninstall the DDC Agent.

```
rpm -e er2
```

Uninstalling Agents from Debian

To uninstall a DDC Agent:

1. Stop the DDC Agent.

```
sudo /etc/init.d/er2-agent -stop
```

2. Uninstall the DDC Agent.

```
sudo dpkg --remove er2
```

Uninstalling Agents from Windows

To uninstall a DDC Agent, you must be logged on as Administrator to the host where the Agent is running.

1. Navigate to the **Control Panel > Programs and Features**.
2. Locate the **Enterprise Recon 2 Agent** in the list of installed programs.
3. Right click the Agent and select **Uninstall**.
4. In the dialog box that is displayed, select to automatically close the **Enterprise Recon 2 Agent** application, and click **OK** to continue.
5. Walk through the wizard.

TIP Alternatively, to uninstall a DDC Agent from CLI, run the following commands as Administrator:

```
net stop "Enterprise Recon 2 Agent (<ARCH>)"  
wmic product where name="Enterprise Recon 2 Agent (<ARCH>)" uninstall
```

Upgrading Agents

Agents do not require an upgrade unless a feature available in an updated version of the Agent is needed. Older versions of the Agent are compatible with newer versions of the Server. To upgrade an Agent, simply re-install it. Prior to reinstalling, you have to uninstall the Agent.

- > For instructions on uninstalling Agents, see:
 - ["Uninstalling Agents" on page 29](#)
 - ["Uninstalling Agents" on page 29](#)
 - ["Uninstalling Agents from Windows" on page 29](#)
- > For instructions on installing Agents, see:
 - ["Installing Agents" on page 26](#)
 - ["Installing Agents" on page 26](#)
 - ["Installing Agents" on page 26](#)

Hardening the Deployment

CipherTrust Manager should be deployed into as secure an environment as possible. Every effort has been made to make CipherTrust Manager as secure as possible, however, additional precautions should be taken especially when CipherTrust Manager is deployed into an untrusted environment. Refer to "Hardening Guidelines" in the "CipherTrust Manager Deployment Guide" for details.

General Hardening Recommendations

When a scan is run, all the data is transferred from data stores to Agents so that they can read and discover info types. As a result, potentially sensitive data is transferred between data stores and Agents. Although we cannot provide the comprehensive list of actions to secure the data being transferred, for security and performance reasons, we offer at least the following recommendations:

- > As a general rule, take all necessary actions to eliminate or minimize the transfer of data between data stores and Agents.
- > It is recommended to install a different Agent in each subnet with sensitive data, and to configure the firewalls to block any connection to the sensitive data stores from Agents in different subnets. This ensures that sensitive data does not leave the subnet and company policies are respected.
- > Agents should not be in a different network security zone. This ensures that data does not cross the network security zone boundaries.
- > Agents should be used in the same subnet as the data stores for Big Data stores. This can prevent firewalls from collapsing.
- > Local Agents should be used, when possible, in data stores with huge amount of data. This ensures that the data never leaves the server.
- > Contact your corporate security teams if data stores reside in different network security zones to determine where to locate the CipherTrust Manager appliance.
- > Agents should be hardened to comply with the company security policies for the sensitivity of the data located in the data stores they will scan, as all the scanned data will be transferred to the Agents.
- > DDC should be configured with secure protocols (for example, using TLS) to connect to data stores whenever possible, to ensure that the data travels protected with channel encryption.
- > Use firewalls to restrict the access to data stores, so only the allowed computers can connect to their endpoints.

Certificate Security Recommendations

- > **Certificate generation**
 - Certificate private keys must not be generated or transformed on public web services.
 - The certificate format must not be transformed on public web services.
 - Cryptographic algorithms and key length must satisfy a minimum security strength of 112 bits.
 - For the symmetric key, we recommend to use at least AES128.

- For the asymmetric key: use DSA-2048, RSA-2048, and ECDSA224-255.
- Certificate validity and Key usage period should not exceed 2 years.
- Digital signatures must use a strong hash function. We recommend using at least SHA256.
- > **Certificate usage**
 - User must keep secret his private key and related activation data.
 - Receivers of a certificate must verify the certificate.
 - Users must respect the PKI Certificate Policy requirements when using a certificate.
 - Do not share Wildcard Certificates for Different Security Contexts or Security Levels.
- > **Certificate revocation**
 - Subscribers must revoke their certificates when necessary (for example, when it's compromised).
- > **Certificate renewal**
 - Subscribers must renew their certificate before it expires.

Securing the Hadoop Configuration

Here are some general guidelines on securing your Hadoop deployment. These recommendations are general and apply to any Hadoop cluster.

- > The Hadoop cluster should only be deployed inside a corporate or trusted network. You must use a VPC network if deploying to a cloud.
- > None of the Hadoop nodes should be exposed to the Internet.
- > To access services from the Internet, it is recommended to use VPN access. If VPN is not available, it is recommended to use jump servers or bastion hosts. For example, to access SSH (port 22), do not expose it directly to the Internet. Instead, use a jump server or a bastion host.
- > To access web services, such as DDC or Ambari, use a reverse proxy or some kind of a load balancer. These devices normally sit in a DMZ (demilitarized zone) and forwards only a specific TCP port to the Hadoop cluster.
- > For Hadoop clusters configure the replication channels to use transport level encryption.

Additionally for the DDC-Hadoop configuration (PQS/HDFS), below you can find some guidelines that will help you set secure credentials and secure TLC configuration. For details and instructions, refer to your Hadoop documentation.

- > **Secure credentials**
 - Hadoop credentials must be rotated regularly.
 - Credentials must be different for every installation.
 - Password must meet the security requirements described below:
 - At least 10 characters in length.
 - Contains both uppercase (A-Z) and lowercase (a-z) alphabetic characters.
 - Has at least one numeric (0-9) character.
 - Has at least one special character (! @ # \$ % ^ & * () _ - + = , . / < > ? ; ' : \ " [] \ \ { } | ~ `).

- Does not contain spaces or tabs.

> **Secure TLS configuration**

- Disable SSL and TLS 1.0 and 1.1.
- Use secure renegotiation or disable renegotiation.
- Disable TLS compression.
- Use Authenticated Encryption cipher suites.
- Use cipher suites with strong key exchange.
- Do not use cipher suites with known vulnerabilities.
- Cipher suites order must be defined by the user.
- Use Perfect Forward Secrecy (PFS).
- Download the tool from <https://testssl.sh/> to check the configuration.

Thales CipherTrust Data Discovery and Classification

ADMINISTRATOR GUIDE



Document Information

Document Information

Product Version	2.0.0
Release Date	10 September 2020

Trademarks

Thales CipherTrust Data Discovery and Classification is powered by Groundlabs.

All intellectual property is protected by copyright. All trademarks and product names used or referred to are the copyright of their respective owners. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, chemical, photocopy, recording or otherwise without the prior written permission of Thales.

Disclaimer

Thales makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Thales reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon Thales to notify any person or organization of any such revisions or changes.

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

You are responsible for ensuring your own compliance with various laws and regulations, including but not limited to any data privacy or data protection regulation. You are solely responsible for obtaining advice from competent legal counsel to assist you in the identification and interpretation of any relevant laws and regulations that may affect your business and the implementation of any actions you may need to take to ensure you meet your compliance obligations with respect to such laws and regulations.

The software, the products, services, and any other capabilities described or provided herein are not suitable for all situations and may have restricted availability or applicability. Thales does not provide legal, accounting, or auditing advice, nor does it represent or warrant that its software, services, or products will ensure that you are in compliance with any law or regulation.

Thales invites constructive comments on the contents of this document. Send your comments, together with your personal and/or company details to the address below.

Contact Method	Contact Information
Address	Thales 4690 Millennium Drive Belcamp, Maryland 21017 USA

Contact Method	Contact Information	
Phone	US	1-800-545-6608
	International	1-410-931-7520
Email	technical.support.DIS@thalesgroup.com	
Technical Support Customer Portal	https://supportportal.thalesgroup.com Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the Thales Knowledge Base.	

CONTENTS

Document Information	2
Preface: About this Document	7
Audience	7
What's in This Guide	7
Organization	7
Document Conventions	8
Hyperlinks	8
Notifications	8
Command Syntax and Typeface Conventions	9
Related Documents	10
Solution Overview	11
Sensitivity Levels	13
Information Types	13
Creating Custom Infotypes	13
Character Type Rules Explained	15
Examples of Custom Infotypes	16
Licensing	18
Trial License	18
Full License Options	18
Viewing the License Status	19
What If My License Stopped Working?	19
Encryption Keys Used by DDC	21
DDC User Groups	22
System Predefined Groups	22
User Defined Groups	23
Accessing and Interacting With DDC	24
Accessing DDC in the Console	24
Managing Branch Locations	25
Viewing Branch Locations	25
Adding Branch Locations	25
Managing Classification Profiles	27
Viewing Classification Profiles	27
Classification Profile Templates	28

Adding Classification Profiles	28
Select Profile Template	28
General Info	28
Select Infotypes	29
Apply Tags	29
Viewing Details of Classification Profiles	30
Editing Classification Profiles	30
Duplicating Classification Profiles	31
Managing Data Stores	32
Viewing Data Stores	32
Adding Local Stores	33
Select Store Type	33
Configure Connection	33
General Info	34
Add Tags & Access Control	34
Adding Network Stores	35
Prerequisites for Network Storage Data Stores	35
Creating a Data Store	35
Creating a Windows Data Store	35
Creating a Linux Data Store	36
Configuring a Data Store - General Information	37
Configuring a Data Store – Tags and Access Control	37
Adding Database Stores	37
Select Store Type	37
Configure Connection	38
General Info	39
Add Tags & Access Control	39
Adding Big Data Stores	40
Select Store Type	40
Configure Connection	40
General Info	40
Add Tags & Access Control	41
Editing Data Stores	41
Automatic Agent Selection	43
Managing Scans	44
Viewing Scans	44
Adding Scans	45
General Info	45
Select Data Stores	45
Add Targets	46
Select Profiles	46
Schedule Scan	47
Running Scans	48
Scan Statuses	48
Potential Problems When Running Scans	49

Removing Scans	49
Managing Reports	50
Viewing Reports	50
Report Types	50
Creating Reports	51
General Info	51
Configure Content	51
Generating Reports	53
Report Details	53
Logging	55
Default Logging Level	55
Identifying DDC Log Messages	55
Security Audit Log Messages	55
Enabling Syslog Logging	56
APPENDICES	57
Error Messages	57
Error Log Messages	63
Reconfiguring Agents	73
Reconfiguring DDC Agents on Windows	73
Reconfiguring DDC Agents on Debian	73
Reconfiguring DDC Agents on RHEL	73
Restarting DDC Agents	74
Restarting Agents on Windows	74
Restarting Agents on Debian	74
Restarting Agents on RHEL	74
Mounting an NFS Share	74
REST API	75
Acquiring an Authorization Token	75
Using the Token	75
Making an API Call	75
CLI	76
Information Types	77
Supported Formats	87
Files	87
Office files	87
Databases	89
Big Data	89
Security Audit Log Event Messages	90

PREFACE: About this Document

This introductory section identifies the audience, provides a brief summary of the contents of this guide, and discusses the documentation conventions used. It contains the following sections:

- > ["Audience" below](#)
- > ["What's in This Guide" below](#)
- > ["Organization" below](#)
- > ["Document Conventions" on the next page](#)

Audience

This document is intended for Thales CipherTrust Data Discovery and Classification (DDC) users responsible for classification of data discovered on data stores. It is assumed that the users of this document are proficient with security and data discovery concepts.

All products manufactured and distributed by Thales are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.

Thales designs data security products for use by file server administrators, network administrators, security engineers, database administrators, application developers, and other technology professionals responsible for daily operations in support of data security.

What's in This Guide

This guide explains data discovery concepts such as data stores, branch locations, classification profiles, and data discovery scans. The document also explains how to generate scan-based reports on the discovered data. Finally, the document describes how to read generated reports.

Organization

The *Thales CipherTrust Data Discovery and Classification Administrator Guide* contains the following chapters:

1. ["Solution Overview" on page 11](#)

Describes data discovery concepts such as branch locations, classification profiles, sensitivity levels, tags, data stores, data discovery scans, and user roles and permissions.

2. ["Licensing" on page 18](#)

Describes different types of DDC licenses.

3. ["DDC User Groups" on page 22](#)

Describes all predefined groups of DDC users with their rights to use various product features.

4. ["Accessing and Interacting With DDC" on page 24](#)

This section introduces the primary channel to interact with DDC , the console.

5. ["Managing Branch Locations" on page 25](#)

Describes how to add and view branch locations.

6. ["Managing Classification Profiles" on page 27](#)

Describes how to add, view, and duplicate classification profiles.

7. ["Managing Data Stores" on page 32](#)

Describes how to add and edit different types of data stores. The types of data stores are local, network, database, and big data.

8. ["Managing Scans" on page 44](#)

Describes how to add, run, and view scans.

9. ["Managing Reports" on page 50](#)

Describes how to configure and run aggregated reports on the discovered data.

10. ["APPENDICES" on page 57](#)

Additional useful information and tools related to system administration, such as system error messages, handy commands, and additional interfaces for interacting with the product.

Document Conventions

This section describes the formatting conventions used in this user guide to indicate hyperlinks, special notes, important information, tips, and warnings.

Hyperlinks

Hyperlinked text will, by default, appear in the shade of purple.

For example: All technical document templates can be found on the [Technical Writing Community page](#).

Notifications

This user guide uses notes, tips, and warnings to alert you to important information that may help you to complete your task, or prevent personal injury, damage to the equipment, or data loss.

Notes

Notes are used to alert you to important or helpful information. These elements use the following format:

NOTE Take note. Notes contain important or helpful information that you want to make stand out to the user.

Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss. These elements use the following format:

CAUTION! Exercise caution. Caution alerts contain important information that may help prevent unexpected results or data loss.

Warnings

Warnings are used to alert you to the potential for catastrophic data loss or personal injury. These elements use the following format:

****WARNING**** Be extremely careful and obey all safety and security measures. In this situation you might do something that could result in catastrophic data loss or personal injury.

Command Syntax and Typeface Conventions

Convention	Description
bold	The bold attribute is used to indicate the following: > Button names (Click Save As.) > Check box and radio button names (Select the Print Duplex check box.) > Dialog box titles (On the Protect Document dialog box, click Yes.) > Field names (User Name: Enter the name of the user.) > Menu names (On the File menu, click Save.) (Click Menu > Go To > Folders.) > User input (In the Date box, type April 1.)
<i>italic</i>	The italic attribute is used for emphasis or to indicate a related document. (See the <i>Thales CipherTrust Data Discovery and Classification Customer Release Notes</i> for more information.)
Double quote marks	Double quote marks enclose references to other sections within the document. For example: Refer to "Disclaimer" on page 2.
<variable>	In command descriptions, angle brackets represent variables. You must substitute a value for command line arguments that are enclosed in angle brackets.
[optional] [<optional>] [a b c] [<a> <c>]	Square brackets enclose optional keywords or <variables> in a command line description. Optionally enter the keyword or <variable> that is enclosed in square brackets, if it is necessary or desirable to complete the task. Square brackets enclose optional alternate keywords or variables in a command line description. Choose one command line argument enclosed within the braces, if desired. Choices are separated by vertical (OR) bars.

Convention	Description
<code>{ a b c }</code> <code>{ <a> <c> }</code>	Braces enclose required alternate keywords or <variables> in a command line description. You must choose one command line argument enclosed within the braces. Choices are separated by vertical (OR) bars.

Related Documents

The following documents contain related or additional information:

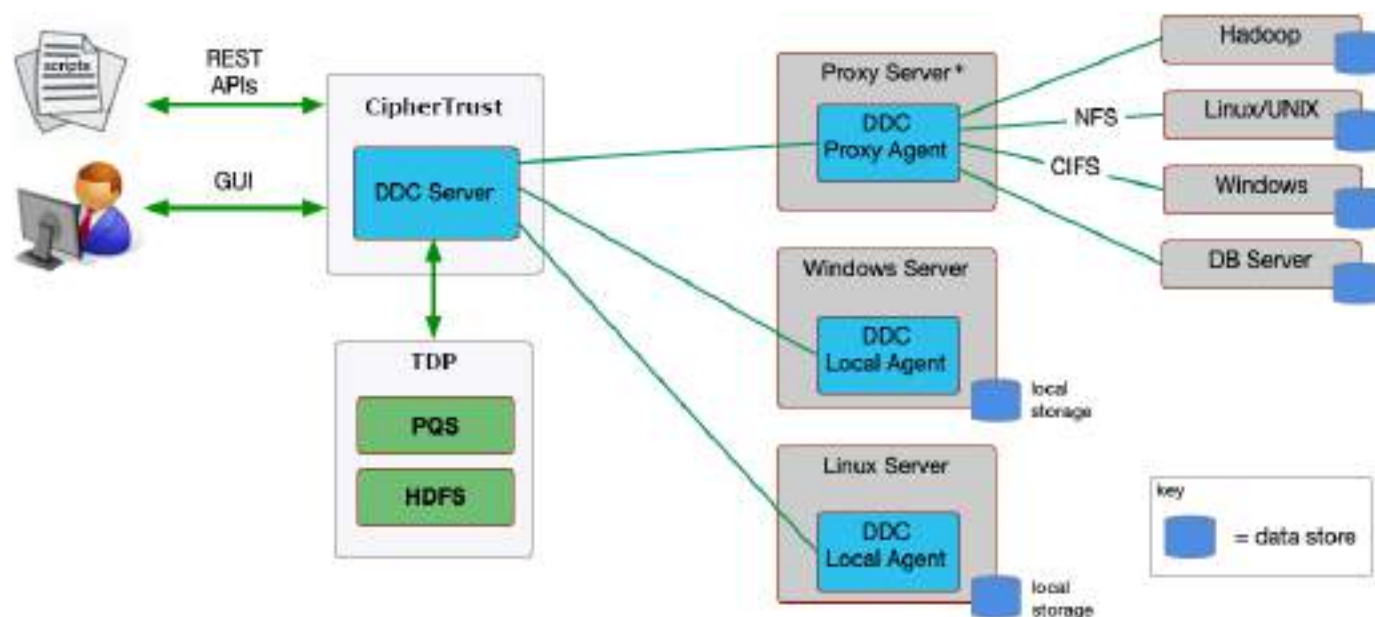
- > *Thales CipherTrust Data Discovery and Classification Deployment Guide*
- > *Thales Data Platform Installation Guide*
- > *Thales CipherTrust Data Discovery and Classification Customer Release Notes*

You can view or download the latest version of the CRN for this release at this location:

<https://supportportal.thalesgroup.com>

Solution Overview

This section describes the main components of the Thales CipherTrust Data Discovery and Classification (DDC) solution. The concepts used in this diagram are briefly discussed in this section and explained at length in the later sections of this document.



> KeySecure, DDC Server

At the heart of the DDC solution is CipherTrust Manager on which runs the DDC Server. It is from here that users interact with the DDC GUI or use the DDC APIs to create classification profiles, add data stores, launch scans and generate reports.

> REST APIs, GUI

Various types of interfaces used to interact with DDC.

> Hadoop, PQS, HDFS

DDC uses Hadoop to generate reports from scans and to store their results (report data). DDC can directly query HDFS but it requires Phoenix Query Server (PQS) to interface with Hadoop's HBase.

> DDC Agent, Proxy Agent, Local Agent

DDC Agents perform the actual scanning jobs and report the results back to the DDC Server for analysis and processing. DDC supports two types of Agent configurations: **Local Agents** are installed and configured directly on the machine that contains sensitive data; **Proxy Agents** are installed and configured on a proxy machine that is used to scan sensitive data on other machines.

> Data Store

A data store is where the data actually resides. It can be a file server, a database, or a Hadoop cluster. For more information see ["Managing Data Stores" on page 32](#).

> Local storage

A type of a data store, a file system (Windows or Linux) that is localized to the same machine where the Agent scanning it is installed.

> NFS, CIFS

A type of data store, a network share (Windows or Linux) that resides on a different machine than that where the Agent scanning it is installed.

> Branch Locations

A branch location specifies a site where the file servers, databases, and data centers that contain data to scan are located. Branch locations are used to indicate where different data stores are physically located. For more information see ["Managing Branch Locations" on page 25](#).

> Sensitivity Levels

A sensitivity level defines how sensitive the data is. Sensitivity levels are required in creating classification profiles and data stores. For more information see ["Sensitivity Levels" on the next page](#).

> Information Types

An information type (or infotype) categorizes data to look for during a scan. A large number of predefined information types are available to better categorize the data. For more information see ["Information Types" on the next page](#).

> Tags

A tag helps group data together. Tags are used to filter data for generating reports. They can be specified when creating data stores and classification profiles.

DDC includes a number of predefined tags. The predefined tags are APA, APPI, CCPA, FINANCIAL, GDPR, HEALTH, HIPAA, KVKK, LEGAL, PCI, PERSONAL, PHI, PII, and SHIELD.

DDC also provides the ability to create custom tags when creating data stores and classification profiles.

> Classification Profiles

A classification profile defines what kind of sensitive information to search for during a scan. It includes information such as a sensitivity level, information types, and tags. Classification profiles can be created based on predefined templates or custom templates. For more information see ["Managing Classification Profiles" on page 27](#).

> Data Objects

A file or a database table stored in a data store is called a Data Object.

> Sensitive Data Objects

A data object that contains any data match is called a Sensitive Data Object.

> Data Matches

A concrete instance of any of the infotypes is called a Data Match.

> Risks

A risk is the presence of a sensitive data object in an unprotected data store.

> Scans

A scan is an entity that helps in scanning data stores. Each scan specifies the location to scan and what to look for during scanning. Findings of scans can be used to generate reports for different purposes. Scans can be either run manually (any time) or scheduled to run and stop at a specified time. For more information ["Managing Scans" on page 44](#).

Sensitivity Levels

A sensitivity level defines how sensitive the data is. Sensitivity levels are required in creating classification profiles and data stores. Prebuilt sensitivity levels are:

- > **None:** The sensitivity level for such data has not yet been specified.
- > **Public:** Specifies the least sensitive data with no specific need for data security. Such data can be shared with anybody.
- > **Internal:** Specifies the data with low sensitivity. Exposure of such data may not affect an organization, but is not meant for public disclosure.
- > **Private:** Specifies that the data is personal. Such data should be protected from public viewing.
- > **Restricted:** Specifies highly sensitive data, for example, customer's personal data and trade secrets etc. This type of data requires the best possible data security. Disclosure of such data can lead to severe financial and legal consequences for an organization. Businesses must prioritize remediation efforts related to this type of data.

Information Types

An information type (infotype) categorizes data to look for during a scan. A large number of prebuilt information types are available to better categorize the data.

Different regions and countries can have different regulatory requirements, so these information types are categorized based on geographical regions. These regions are Global, Africa, Americas, Asia, Europe, and Oceania. The information types can be further categorized into:

- > **Financial:** Financial data such as credit card numbers and bank account details.
- > **Personal Data:** Personal data such as age, gender, race, and religion.
- > **Medical:** Medical data such as history of medical problems and disabilities.
- > **National ID:** National identity documents such as Social Security Number (SSN).

For a list of all available predefined information types, refer to the appendix ["Information Types" on page 77](#).

DDC also allows you to create custom information types. For more information, see ["Creating Custom Infotypes" below](#).

Creating Custom Infotypes

You can create a custom information type, if you require one. This can be achieved from the **Infotypes** screen. To access it, click **Settings** and then **Infotypes** in the sidebar on the left.

1. Click the **+Add Infotype** button in the top right corner of the **Infotypes** screen. The **Add Infotype** wizard is displayed.
 - a. In the **General Info** step of the wizard, provide the following information for your new infotype:
 - **Name:** Choose a name for your infotype.
 - **Category:** Select a category to which your infotype belongs (Financial, Personal Data, Medical, or National ID).

- **Family:** Select a family for your infotype. A family is a subcategory inside the **Category** and the choice of options depends on what you selected in the **Category** menu. The following families are available inside their corresponding categories:

Financial: Credit/Debit Cards; Bank Account Info

Medical: Patient Health Data

National ID: Personal Identification

Personal Data: Email addresses; Login credentials; Card Number; Ethnicity; License Number; Roll Number; Passport Number; Date Of Birth; MAC Address; Mailing Address; Telephone Number; Gender; Religion; IP Address; Phone Number; Name

- **Region:** Select the region for your infotype (Global, Africa, Americas, Asia, Europe, and Oceania).

Click **Next** to go to the next step of the wizard.

- In the **Infotype Definition** step of the wizard, you configure the rules for your new information type. You configure the rules in the **Simple View** tab, and then you can view these rules as translated into an expression of the internal language of the DDC engine in the **Expert View** tab (the expression in the Expert View is **read-only**).

To configure the rules for your new information type, click to expand the **Add Rules** menu in the **Simple View** tab and select one of the following types:

- **Character:** Search for one or more specific characters as specified in the **Select Rule** menu. If the character is found, the location will be returned as a match. For a list of available character type rules, refer to ["Character Type Rules Explained" on the next page](#).

Use the **From** and **To** controls to set the number of consecutive occurrences of the selected character.

- **Phrase:** Search for a specific pattern as defined in the **Phrase** textbox (in layman terms, it is used to look for specific words). Searching for phrases is **case insensitive**.
- **Built-In:** Pre-defined infotypes can be used in combination with other types (Character or Phrase). The complete list of built-in information types is available in the appendix ["Information Types" on page 77](#).

Use the **Apply** button to complete your selection. The selection is displayed in the list of defined infotype rules. You can remove it from the list of rules by clicking the **Remove** link on its right.

You can use each of these types on their own, or combine them to form a more complex rule, involving multiple types in various configurations. See ["Examples of Custom Infotypes" on page 16](#) for some examples.

NOTE

- > Due to a known limitation, it is currently not possible to have two built-in infotypes, one after the other. They must be separated by something, such as another type infotype, or even a plain space. For example:
"american express" " " "english name"
- > Due to a known limitation, when adding a range of characters, all the possible combinations inside the range will appear as matches. For example, for a range of numbers from 1 to 4, when in a document you have a sequence "1234", the search will yield the following matches:
"1", "12", "123" and "1234"
- > When you introduce spaces at the beginning or the end of the phrase, the spaces are removed. Also, when you introduce more than one space between words, only one space is considered.

2. Click **Save** to save your new infotype. Your new information type has been added and is now listed in the **Infotypes** screen, and marked 'Custom' in the **Type** column.

Character Type Rules Explained

Specific predefined characters are used to create custom infotypes using character based rules. They are explained below:

Rule	Expert View Keyword	Usage
Space	SPACE	Matches any ASCII whitespace character such as blank space, horizontal tab, new line, vertical tab, form feed and carriage return.
Horizontal space	HSPACE	Matches an ASCII horizontal space only.
Vertical space	VSPACE	Matches an ASCII vertical space only.
Any	BYTE	Matches any byte within the ASCII 0x00 to 0xFF range.
Alphanumeric	ALNUM	Matches any ASCII alphanumeric character.
Alphabet	LETTER	Matches any ASCII alphabet character.
Digit	DIGIT	Matches any ASCII numeral.
Printable	PRINTABLE	Any printable ASCII character including horizontal and vertical whitespace.
Printable ASCII only	PRINTABLEASCII	Matches a printable ASCII only.

Rule	Expert View Keyword	Usage
Printable non-alphabet	PRINTABLENONALPHA	Matches any printable ASCII characters excluding alphabet characters and including horizontal and vertical whitespace.
Printable non-alphanumeric	PRINTABLENONALNUM	Matches any printable ASCII characters excluding alphanumeric characters and including horizontal and vertical whitespace.
Graphic	GRAPHIC	Matches any ASCII character that is not whitespace or a control character.
Same line	SAMELINE	Matches any printable ASCII character including horizontal whitespace but excluding vertical whitespace.
Non-alphanumeric	NONALNUM	Matches any character that is not an ASCII alphanumeric character.
Non-alphabet	NONALPHA	Matches any character that is not an ASCII alphabet.
Non-digit	NONDIGIT	Matches any character that is not an ASCII numeral.

Examples of Custom Infotypes

Example 1. You want to search for a "Driver License Number" from Illinois, whose format is "M532-4218-1341". You would then create the following rule:

Character Alphabet From 1 to 1 Times

Character Digit From 3 to 3 Times

Phrase -

Character Digit From 4 to 4 Times

Phrase -

Character Digit From 4 to 4 Times

The above example will have the following syntax in the **Expert View**:

```
RANGE LETTER TIMES 1-1
THEN RANGE DIGIT TIMES 3-3
THEN WORD NOCASE '-'
THEN RANGE DIGIT TIMES 4-4
THEN WORD NOCASE '-'
THEN RANGE DIGIT TIMES 4-4
```

In this rule, the expression "**Character Alphabet From 1 to 1 Times**" means that you only expect one alphabetic character, the expression "**Character Digit From 3 to 3 Times**" means that you expect exactly three digits, and the expression "**Phrase -**" means that you expect to find a hyphen (-) in the sequence.

Example 2. You want to search for a name and last name separated by a number of spaces between 1 and 3. To that end you would create the following rule:

Phrase John

Character Space **From** 1 **to** 3

Phrase Gordon

The above example will have the following syntax in the **Expert View**:

```
WORD NOCASE 'John'
THEN RANGE SPACE TIMES 1-3
THEN WORD NOCASE 'Gordon'
```

This rule will allow you to search for a combination of "John" and "Gordon" with one through three spaces between them. By comparison, using the rule **Phrase** John Gordon will only allow you to search for a combination "John Gordon", with only one space. Any additional spaces in the phrase will be truncated.

Example 3. You want to search for the Spanish NIE (foreigners identity number) preceded by the phrase "NIE:" and a number of spaces between 0 and 5. For example, "NIE: X8691474Q".

Phrase NIE

Character Space **From** 0 **to** 5

Built-in Spanish NIE

The above example will have the following syntax in the **Expert View**:

```
INCLUDE 'DEFINE_NID'
WORD NOCASE 'NIE'
THEN RANGE SPACE TIMES 0-5
THEN REFER 'NID_SPAIN_NIE'
```

This rule will find both "NIE: X8691474Q" and "nie: x8691474q" since searching (regardless of the type - Character, Phrase, or Built-in) is case insensitive.

Licensing

Trial License

Thales CipherTrust Data Discovery and Classification (DDC) is deployed with a trial license already installed and activated "out of the box". This allows you to enjoy a fully-functional product for **90 days** and up to the **2 TB** of data allowance.

After the trial license expires, the DDC configuration in CipherTrust Manager becomes read-only. While you still have access to your old reports you are not able to generate new ones, add new targets, or create new scans. You have to contact Thales Group to request a full license and install it in CipherTrust Manager. For more information on obtaining and installing licenses, refer to the "Licensing" section in the "Thales CipherTrust Manager Administrator Guide".

CAUTION! Data allowances of the trial license and full license do not add up! After installing a new full license your data allowance will be that of the new license only (for example, if your trial license has 2 TB data allowance and your full license 50 TB, after installing the full license your data allowance will be 50 TB not 52 TB).

Full License Options

Under the full license you get a fully-functional product with a specified data allowance and for a specific period of time. The license model offers you enough flexibility to choose an option that best suits your needs in terms of the license duration and prospective data allowance. You can choose from among these values:

> Expiration period

- 1 year
- 2 years
- 3 years

> Data allowance

- 15 TB
- 50 TB
- 100 TB
- 150 TB
- 250 TB
- 500 TB
- 1 PB
- 1.5 PB
- 3 PB
- Unlimited

Viewing the License Status

To view the status of your DDC license:

1. Log in to CipherTrust Manager and navigate to the licenses screen (**Admin Settings > Licensing**).
2. Search for **DDC_Data_Allowance** in the list of installed features.

TIP Use the **Search** box if the list is too long and you cannot quickly find the DDC_Data_Allowance entry.

3. Having found the DDC license, you have a few options available:
 - You can quickly check its status by looking at the **State** column in the DDC_Data_Allowance entry. It can be either **Active** or **Expired**.
 - You can check the license expiration date, in the **Expiration** column.
 - If the license is still active, you can view additional details about it. To do that, click the black arrow on the left of DDC_Data_Allowance to expand the whole entry. This displays the **Client Usage** card, with the information about the total data allowance that you have (**Total**), the amount of data allowance that you have already used up (**Used**), and the amount of data allowance that you still have available (**Available**).

NOTE If the **Available** figure has a negative value, it means that you have used up and exceeded your available data allowance. However, you can still run scans and the data from the scans is stored by DDC. The amount of this extra data stored is reflected by the negative value. You can access this data after you install a new license.

NOTE After installing or removing a license you have to wait some time for that action to be reflected in the licenses screen (usually, about one minute).

What If My License Stopped Working?

A license - trial or full - can stop working in one of the following cases:

- > **A trial license expires:** you cannot run new scans. However, the data collected so far is not deleted, so you can still generate reports based on it, and you can access it again when you install a new license.
- > **The data allowance of a trial license is used up:** you can continue scanning but you cannot generate reports. However, the data from scans is stored so after you install a new license you are able to access the data and generate reports.
- > **A license is deleted:** DDC stops working but the data is not deleted so you just have to reinstall the license or install a new one.
- > **You overwrite a trial license with a full one:** the new license takes over the data stored under the trial license.
- > **A full license expires:** it is the same case as the expired trial license above.

- > **The data allowance of a full license is used up:** it is the same case as the trial license with data allowance used up above.

Encryption Keys Used by DDC

DDC uses AES256 encryption to protect sensitive data. For that purpose, DDC creates a number of encryption keys that are stored in CipherTrust Manager. You can find these DDC keys in the **Keys & Access Management** application in CipherTrust Manager:

- > Four encryption keys to protect the Hadoop configuration before storing it inside the DDC Database. Each key is used to protect one configuration parameter (PQS Server, PQS credentials, HDFS Server, and HDFS credentials). These keys have the following format: **citrus-<UUID>** (for example, citrus-6e0cb668-3a3d-4f2c-8687-17092b83b41b).
- > As many encryption keys as there are data stores, and each key is used to encrypt the data store credentials before storing them inside the DDC Database, and to encrypt the results of the scans completed in that data store, before storing them in HDFS. These keys have the following format: **d<UUID>** (for example, d8b2d8404-c9ae-4a34-800a-01258dfaa383).
- > As many encryption keys as there are scans, and each key is used to encrypt the scan data before storing it in HDFS. These keys have the following format: **s<UUID>** (for example, s14912791-bed5-4e73-b733-6a36ecfe338f).

****WARNING**** These keys must never be deleted, or DDC will not be able to process the related scans or data stores properly.

DDC User Groups

System Predefined Groups

DDC has different kinds of users with different responsibilities in administering and using the system. A number of predefined groups are included to ensure that users are granted minimal permissions needed to perform their tasks while ensuring flexibility to meet security requirements across industries.

The table below lists all DDC predefined groups with their rights to use various DDC features. **R/W** in a cell means that the user has **view** and **edit** rights to this aspect of the product. **R** means that the user has only **view** rights.

	DataStores	Scans	Profiles	Reports	SensitivityLevels	BranchLocations	Custom Infotypes ⁷	Hadoop Configuration
Admin ¹	R/W	R/W	R/W	R/W	R/W	R/W		R/W
DDC Admins ²	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DDC Store Viewers	R				R	R		
DDC Store Admins	R/W				R	R/W		
DDC Scan Viewers ⁶		R					R	
DDC Scan Admins	R	R/W	R			R	R	
DDC Profile Viewers			R				R	
DDC Profile Admins		R	R/W		R		R	
DDC Report Admins ^{3, 5}	R	R	R	R/W	R	R	R	
DDC Full Report Admins ^{4, 5}	R	R	R	R/W	R	R	R	
DDC Infotype Viewers							R	
DDC Infotype Admins							R/W	
DDC L3 Support								R/W

1. **Admins** can see their own and other users' reports. Admins can also decrypt scan packages from the Hadoop database.
2. **DDC Admins** can only see their own reports.
3. **DDC Report Admins** can only see their own reports.

4. **DDC Full Report Admins** can only see their own reports.
5. The difference between **Report Admins** and **Full Report Admins** is that **Full Report Admins** do not need access to specific user defined groups to be able to view or generate reports that use data stores restricted to user defined groups. For more information, see "[User Defined Groups](#)" below.
6. Scan Viewers are allowed to run scans.
7. **DDC Store Viewers**, **DDC Store Admins**, and **DDC L3 Support** do not have access to custom infotypes.

The users belonging to the "L3 Support" group are DDC Support Administrators. These users can help identify and troubleshoot issues you may encounter when using DDC. They can also can also decrypt scan packages from the Hadoop database.

User Defined Groups

Apart from system predefined groups, DDC also allows you to create **user defined groups**. User defined groups are used to prevent certain users from viewing sensitive information in reports. These groups are defined by the CipherTrust Manager Admin in Keys & Access Management -> Groups in CipherTrust Manager. In DDC, they are applicable when creating a Data Store, in the Data Store creation wizard when you are granting access to selected groups. See "[Managing Data Stores](#)" on page 32 for details (the **ACCESS: Selected group/s** setting in "Configuring a Data Store – Tags and Access").

In other words, a data store that is restricted to a specific user defined group is visible to all the groups with permissions to see data stores, the same goes for scans. However, a user without a permission to see a data store which is restricted to a group, but with a permission to create and generate reports, will not be able to generate reports for those data stores. For this user, the scan executions will not be visible in the New Report wizard.

NOTE The only users that do not have to belong to a specific user defined group to be able to see reports for all the data stores are **Full Report Admins** and **Admins**.

For example, if a user created a report that has Data Store "DS1" but the data store is restricted to a specific group, that user will see the report template, but when he tries to access the report he will get an "insufficient permissions" error.

Accessing and Interacting With DDC

The primary channel to interact with Thales CipherTrust Data Discovery and Classification (DDC) is through the CipherTrust Manager's GUI, also called the **console**. The console allows you to perform the management operations, such as managing data stores and scans.

You can also interact with DDC by using the CLI tool or REST API.

- > **CLI tool** - The CipherTrust Manager includes a CLI tool, named **ksctl**, that can be downloaded and run locally to control a remote CipherTrust Manager appliance. For more information, refer to ["CLI" on page 76](#).
- > **REST API** - You can use the REST interface from the *API playground*, or via any REST client such as curl. For more information, refer to ["REST API" on page 75](#).

In this guide we provide instructions to perform all management functions such as creating branch locations and data stores only through the CipherTrust Manager console (GUI).

Accessing DDC in the Console

Use this procedure to get access to the DDC features in the GUI.

1. Open the CipherTrust Manager URL in a browser. The log in page is displayed.
2. Enter **Username** and **Password**.
3. Click **Log In**. The GUI of the CipherTrust Manager is displayed.
By default, the **Applications** page is displayed with links to various applications.
4. Click the **Data Discovery** link to open the Data Discovery application.



Managing Branch Locations

You manage branch locations through the **Branch Locations** page, which is accessed by clicking the **Settings > Branch Locations** link in the **Data Discovery** sidebar on the left.

From the **Branch Locations** page you can:

- > View all currently available branch locations. See "[Viewing Branch Locations](#)" below.
- > Create a new branch location. See "[Adding Branch Locations](#)" below.

Viewing Branch Locations

The **Branch Locations** page lists available branch locations. This view also shows the total number of existing branch locations. The **Branch Locations** page shows the following details:

Item	Description
Site Name	Name of the branch location.
Country	Name of the country.
State/Province	Name of the state/province. This field is applicable to the United States. For other countries, the field is unavailable, indicated by N/A.
City	Name of the city.

TIP Use the **Search** text box to filter branch locations. Search results display branch locations that contain the specified text in their names.

Adding Branch Locations

Adding a branch location requires specifying the country and city where the branch is located. To add a branch location:

1. Click **+ Add Location** on the right of the **Branch Locations** page.
2. In the **Add Branch Location** dialog box, enter the following details:

Item	Description
Site Name	Specify a unique name for the branch location. The name must be longer than two characters and up to 64 characters. This field is mandatory.
Country	Select the country from the drop-down list. This field is mandatory.
State/Province	Select the state/province from the drop-down list. This field is applicable to the United States. For other countries, the field is unavailable.

Item	Description
City	Specify name of the city. This field is mandatory.
Description	Describe the branch location (up to 250 characters).

3. Click **Save.**

The newly created location appears on the Branch Locations page. By default, branch locations are displayed in alphabetic order by name. Depending on the number of entries per page, you might need to navigate to other pages to view the newly created branch location.

Managing Classification Profiles

You manage classification profiles through the **Classification Profiles** page, which is accessed by clicking the **Classification Profiles** link in the **Data Discovery** sidebar on the left.

From the **Classification Profiles** page you can:

- > View all the available classification profiles. See ["Viewing Classification Profiles" below](#).
- > Create a new classification profile. See ["Adding Classification Profiles" on the next page](#).
- > View details of a selected classification profile. See ["Viewing Details of Classification Profiles" on page 30](#).
- > Modify an existing classification profile. See ["Editing Classification Profiles" on page 30](#).
- > Create a new classification profile from an existing one. See ["Duplicating Classification Profiles" on page 31](#).

Viewing Classification Profiles

The **Classification Profiles** page lists available classification profiles. Initially, the page shows prebuilt classification profile templates only. Newly created and duplicate classification profiles are also shown on this page. Duplicating classification profiles is the process of creating copies of existing profiles with identical properties. Additionally, the page shows the total number of available profiles.

The list view of the **Classification Profiles** page shows the following details:

Item	Description
Name	Name of the classification profile.
Infotypes	Number of infotypes linked with the profile.
Sens. Level	Sensitivity level applied to the profile.
Modified	Time when the profile is modified.
Tags	Number of applied tags.

TIP

- > Use the **Search** text box to filter classification profiles. Search results display classification profiles that contain specified text in their names.
- > By default, classification profiles are listed in ascending alphabetic order of their names. Classification profiles can be sorted by their names and percentage of sensitivity levels.
- > Classification profiles can be filtered using the **Sens. Level** filter.

Classification Profile Templates

Classification profiles can be created based on predefined templates or custom templates. You can use the following predefined templates:

- > **Blank:** Provides the ability to specify custom sensitivity level, information types, and tags for the profile. The Blank template does not contain preselected infotypes.
- > **CCPA (California Consumer Privacy Act):** Affects organizations that process the personal data of a California resident, regardless of where the organization is headquartered.
- > **GDPR (General Data Protection Regulation):** Affects organizations that process the personal data of EU citizens, regardless of where the organization is headquartered.
- > **HIPAA (Health Insurance Portability and Accountability Act):** Covers the healthcare information in the US. HIPAA relates to protection, encryption, and key management.
- > **PCI (Payment Card Industry):** Affects organizations that play a role in processing credit and debit card payments. These organizations must comply with the strict PCI DSS (Data Security Standard) compliance requirements for the processing, storage, and transmission of data.
- > **Privacy Shield:** Regulates transatlantic exchanges of personal data for commercial purposes between the European Union and United States.

Adding Classification Profiles

Use the Add Classification Profile wizard to add a classification profile. Adding a classification profile involves the following steps:

Select Profile Template

1. In the **Classification Profiles** page, click the **+ Add Profile** button on the right. The **Add Classification Profile** wizard is displayed.
2. In the **Select Profile Template** step, select the required profile template from these options:
 - **Blank**
 - **CCPA**
 - **GDPR**
 - **HIPAA**
 - **PCI**
 - **Privacy Shield**

See "[Classification Profile Templates](#)" above for more information about these classification profile templates.

3. Click **Next** to go on to the **Name and Describe** screen.

General Info

1. Specify a **Profile Name**. The name must be longer than two characters and up to 64 characters.

2. Provide a **Description** for the profile (up to 250 characters).
3. Select a **Sensitivity Level** from the drop-down list. A sensitivity level suggests DDC what level of sensitivity is OK to find in this data store. For details, see ["Sensitivity Levels" on page 13](#).
4. Click **Next** to go on to the **Select Infotypes** screen.

Select Infotypes

1. The **Select Infotypes** screen is displayed. The screen shows the list of available information types. The screen shows details such as Infotype Name, Category, and Region.

NOTE Based on the selected profile template, certain information types may already be applied/selected. The applied information types can be viewed by turning on the **Selected only** toggle switch.

2. Search for the required infotypes. You can use the following options:
 - **Search** text box: Enter text to filter information types. Search results display information types that contain specified text in their names.
 - **Category** filter: Click the filter icon, select or clear categories, and click **OK**.
 - **Region** filter: Click the filter icon, select or clear categories, and click **OK**.
3. Click **Next** to go on to the **Apply Tags** screen.

Apply Tags

1. The **Apply Tags** screen is displayed.

NOTE Based on the selected profile template, certain tags may appear already applied. You can select existing tags, enter new tags, and remove existing tags, as appropriate.

2. Select a tag from the **Add Tags (optional)** drop-down list. The prebuilt tags are APA, APPI, CCPA, FINANCIAL, GDPR, HEALTH, HIPAA, KVKK, LEGAL, PCI, PERSONAL, PHI, PII, and SHIELD.

TIP

- > New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down list.
- > Add as many tags as needed.
- > To remove a tag, click the close icon in the tag name.

3. Click **Save**.

The newly created classification profile appears on the Classification Profiles page. By default, profiles are displayed in alphabetic order by name. Depending on the number of entries per page, you might need to navigate to other pages to view the newly created profile.

Viewing Details of Classification Profiles

The default view of the Classification Profiles page lists prebuilt profile templates, created profiles, and duplicate profiles. Names of profiles and their modification times are also shown. Additionally, the view shows infotypes, sensitivity levels, and tags applied to available classification profiles.

The edit view of the page shows additional details of each classification profile. The details include the number of linked scans, profile name, profile description, applied sensitivity level, list of linked infotypes, and applied tags.

To view details of a classification profile:

1. In the left pane of the **Data Discovery** application, click **Classification Profiles**. The **Classification Profiles** page is displayed. This page lists available classification profiles.
2. Click the overflow icon () corresponding to the desired profile. A shortcut menu appears.
3. Click **View**. The **Classification Profiles** page shows additional details of the profile.

NOTE

- > For new and duplicate classification profiles, the button name is **View/Edit**. Clicking this button shows the edit view of the **Classification Profiles** page. The details can be viewed and edited on this page. Refer to ["Editing Classification Profiles" below](#) for details.
- > Only the users with appropriate rights can see the **View/Edit** button. For all other users, only the **View** button is visible.

Editing Classification Profiles

Newly created and duplicate classification profiles can be modified to suit your requirements. Use the edit view of the page to modify individual classification profiles. You can edit the profile name, profile description, applied sensitivity level, linked infotypes, and applied tags.

NOTE Prebuilt classification profiles cannot be edited. However, you can duplicate them and edit the copy to suit your requirements.

To edit a new or a duplicate classification profile:

1. In the left pane of the **Data Discovery** application, click **Classification Profiles**. The **Classification Profiles** page is displayed. This page lists available classification profiles.
2. Click the overflow icon () corresponding to the desired profile. A shortcut menu appears.
3. Click **View/Edit**. The edit view of the **Classification Profiles** page appears.

NOTE

- > For new and duplicate classification profiles, the button name is **View/Edit**. Clicking this button shows the edit view of the **Classification Profiles** page. The details can be viewed and edited on this page.
- > Only the users with appropriate rights can see the **View/Edit** button. For all other users, only the **View** button is visible.

4. Expand **GENERAL**. General details are displayed.
5. Modify the required details.
6. Expand **INFOTYPES**. The list of infotypes is displayed.
7. Select or clear infotypes, as required.
8. Expand **TAGS**. The applied tags, if any, are displayed.
9. Add new tags or modify existing tags, as required.
10. Click **Save Changes**.

The list view of the Classification Profiles page shows updated information.

Duplicating Classification Profiles

Duplicating classification profiles is the process of creating copies of existing profiles with identical properties. This process simplifies the creation of new profiles. Duplicate profiles can be modified later to suit your requirements, if needed.

To duplicate a classification profile:

1. In the left pane of the **Data Discovery** application, click **Classification Profiles**. The **Classification Profiles** page is displayed. This page lists available classification profiles.
2. Click the overflow icon () corresponding to the desired profile. A shortcut menu appears.
3. Click **Duplicate**.

A message appears stating that the profile has been duplicated successfully. The duplicate profile with the name **<original_profile_name> - Copy** appears on the Classification Profiles page. For example, if the profile **APA - Australia Privacy Amendment** is duplicated, a profile named **APA - Australia Privacy Amendment - Copy** is created.

Managing Data Stores

You manage data stores through the **Data Stores** page, which is accessed by clicking the **Data Stores** link in the **Data Discovery** sidebar on the left.

From the **Data Stores** page you can:

- > View all the available data stores. See ["Viewing Data Stores" below](#).
- > Create a new local type data store. See ["Adding Local Stores" on the next page](#).
- > Create a new network type data store. See ["Adding Network Stores" on page 35](#).
- > Create a new database type data store. See ["Adding Database Stores" on page 37](#).
- > Create a new Big Data type data store. See ["Adding Big Data Stores" on page 40](#).
- > Edit an existing data store. See ["Editing Data Stores" on page 41](#).
- > Select an Agent for a data store. See ["Automatic Agent Selection" on page 43](#).

Viewing Data Stores

The list view of the Data Stores page shows the number of:

- > Existing data stores with the number of scanned and unscanned data stores.
- > Supported data types with the number of configured data stores of each type.
- > Scanned data stores with the number of data stores containing sensitive and nonsensitive data.

Click the refresh button to refresh the displayed information.

The list view of the **Data Stores** page shows the following details:

Item	Description
Name	Name of the data store.
Type	Type of the data store.
Sens Level	Sensitivity level applied to the data store.
Location	Location of the data store.
Tags	Number of applied tags.
%Sens. Info	Percentage of data objects in the data store that are considered as sensitive data objects. A hyphen "-" indicates that a data store is not yet scanned.
Status	Status of the data store - enabled or disabled. During a scan, DDC searches for agents in enabled data stores. Click the toggle switch to change the status. The status of a data store could be disabled while it waits for an Agent or if it fails to select an Agent. Disabled data stores are skipped during the scan.

TIP

- > Use the **Search** text box to filter data stores. Search results display data stores that contain specified text in their names.
- > By default, data stores are listed in ascending alphabetic order of their names.
- > Data stores can be sorted by their names, types, sensitivity levels, locations, and percentage of sensitive information.

Adding Local Stores

Use the Add Data Store wizard to add a local data store. Adding a data store involves the following steps:

Select Store Type

1. In the left pane of the **Data Discovery** application, click **Data Stores**. The **Data Stores** page is displayed. This page lists available data stores.

2. On the right, click **+ Add Data Store**. The **Add Data Store** wizard is displayed.

The **Select Data Store** screen displays options to filter data store types:

- **Filter by Data Store category:** Shows categories of data stores. Click a category to filter available options under the **Select Type** drop-down list.
- **Select Type:** Shows types of data storage. By default, the drop-down list shows all types of data stores. When a category is selected under **Filter by Data Store category**, the label **Select Type** is changed to reflect the selection. For example, for **Local Storage**, the label becomes **Select Local Storage Type**.

NOTE This document uses **Filter by Data Store category** to filter data stores.

3. Under **Filter by Data Store category**, click **Local Storage**.
4. From the **Select Local Storage Type** drop-down list, select **Local Storage**.
5. Click **Next** to go on to the **Configure Connection** screen.

Configure Connection

1. The **Configure Connection** screen is displayed.
2. Specify **Hostname/IP** of the machine where the local data store resides. Specify a valid hostname, IP address, or Uniform Resource Identifier (URI). The hostname must be longer than two characters. This is a mandatory field.

NOTE Local data stores need a DDC Agent installed on the same host.

3. Click **Next** to go to the **General Info** screen.

General Info

1. The **General Info** screen is displayed.
2. Specify a unique **Name** for the data store. The name must be longer than two characters and up to 64 characters.
3. Provide a **Description** for the data store (up to 250 characters).
4. Select a **Branch Location** from the drop-down list.
5. Select a **Sensitivity Level** from the drop-down list. A sensitivity level suggests to DDC what level of sensitivity is OK to find in this data store. For details, see ["Sensitivity Levels" on page 13](#).

NOTE The **Enable Data Store** check box is selected by default. This means that this data store is available for scans. If the check box is cleared, the data store is disabled (not available) for scans.

6. Click **Next** to go on to the **Add Tags & Access Control** screen.

Add Tags & Access Control

1. The **Add Tags & Access Control** screen is displayed.
2. Under **ACCESS**, select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store.

The available options are:

- **All groups:** All groups of users can access the data store through reports. This is the default setting.
- **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group.

If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.

3. Under **TAGS**, select a tag from the **Add Tag** drop-down list. The prebuilt tags are APA, APPI, CCPA, FINANCIAL, GDPR, HEALTH, HIPAA, KVKK, LEGAL, PCI, PERSONAL, PHI, PII, and SHIELD.

TIP

- > New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down list.
- > Add as many tags as needed.
- > To remove a tag, click the close icon in the tag name.

4. Click **Save**.

The newly created data store appears on the Data Stores page. By default, data stores are displayed in alphabetic order by name. Depending on the number of entries per page, you might need to navigate to other pages to view the newly created data store.

Adding Network Stores

DDC supports two types of Network Storage types as data stores: Linux Network File Share (**NFS**) and Windows share (**SMB/CIFS**).

NOTE SMB/CIFS is supported for Windows only. Currently, the SMB implementation on Linux (Samba) is not supported. Also, we cannot guarantee that NFS type data stores on MAC will work properly.

Prerequisites for Network Storage Data Stores

To create a **Windows Network Storage** data store:

- > Use a Windows Proxy Agent.
- > Ensure that the target storage is accessible from the Proxy agent host.

To create a **Linux Network Storage** data store:

- > Use a Linux Proxy Agent.
- > The target storage path must be mounted on the Proxy agent host.

For both types of these data stores, the credentials to access the target storage must have the minimum permissions required to scan it. Bear in mind that data discovery or scanning of data requires **read access**.

Creating a Data Store

To create a new data store, navigate to the **Data Stores** screen (**Data Discovery > Data Stores**). Click the **+Add Data Store** button to open the **Add Data Store** wizard.

In the wizard, you have to go over four configuration steps for each data store that you create:

1. **Select Store Type** - Select a data store type that you want to create. Refer to individual procedures for each data store type for details.
2. **Configure Connection** - provide the connection details for the data store that you selected in the previous step. This step is different for every data store type. Refer to individual procedures for each data store type for configuration details.
3. **General Info** - specify the name, description, branch location, and sensitivity level for your data store. These settings are shared by all data store types. See ["Configuring a Data Store - General Information" on page 37](#) for details.
4. **Add Tags & Access Control** - grant access rights to your data store and add tags. These settings are shared by all data store types. See ["Configuring a Data Store – Tags and Access Control" on page 37](#) for details.

Creating a Windows Data Store

1. To create a Windows Data Store, click **Network Storage > SMB/CIFS Share** in the **Select Store Type** screen in the **Add Data Store** wizard. For details, refer to ["Configuring a Data Store – Tags and Access Control" on page 37](#).

2. In the **Configure Connection** screen of the wizard, provide the following configuration details for your data store:
 - > **Hostname/IP** - a valid hostname, IP address, or URI of the data store.
 - > **Share Name** - a valid Windows share name. These characters are not allowed in the Share Name: = * ? , < > | ; : + [] " / \

CAUTION! Do not confuse the **Share Name** with the **Network Path**. In Windows, the **Share Name** is typically set in the **Advanced Sharing** settings in the folder sharing properties.

- > **Credentials** - provide a valid username and password. Use the appropriate user name format for the target Windows hosts credentials:
 - `<domain\username>` - target host resides in the same Active Directory domain as the Windows proxy agent.
 - `<target_hostname\username>` - target host does not reside in the same Active Directory domain as the Windows proxy agent.
3. In the **General Info** screen of the wizard, specify the name, description, branch location, and sensitivity level for your data store. See ["Configuring a Data Store - General Information" on the next page](#) for details.
 4. In the **Add Tags & Access Control** screen of the wizard, grant access rights to your data store and add metadata. See ["Configuring a Data Store – Tags and Access Control" on the next page](#) for details.
 5. Click **Save** to create the data store. At any time during the configuration you can click **Back** to go to any of the previous wizard screens to update the configuration.

Creating a Linux Data Store

1. To create a Linux Data Store, click **Network Storage > NFS Share** in the **Select Store Type** screen in the **Add Data Store** wizard. For details, refer to ["Configuring a Data Store – Tags and Access Control" on the next page](#).
2. In the **Configure Connection** screen of the wizard, provide the following configuration details for your data store:
 - > **Hostname/IP** - a valid hostname, IP address, or URI of the data store.
 - > **Share Path** - a valid NFS path, it must begin with a slash ("/"). The path must be set to the mount path on the Proxy host.
 - > **Agent Hostname/IP** - a valid hostname, IP address, or URI of the host where the DDC agent resides.
 - > **Mount Point (On Proxy Agent)** - the mount path on the Proxy host (for the **Share Path** above). See also ["Mounting an NFS Share" on page 74](#).
3. In the **General Info** screen of the wizard, specify the name, description, branch location, and sensitivity level for your data store. See ["Configuring a Data Store - General Information" on the next page](#) for details.
4. In the **Add Tags & Access Control** screen of the wizard, grant access rights to your data store and add metadata. See ["Configuring a Data Store – Tags and Access Control" on the next page](#) for details.
5. Click **Save** to create the data store. At any time during the configuration you can click **Back** to go to any of the previous wizard screens to update the configuration.

Configuring a Data Store - General Information

The **General Info** screen in the **Add Data Store** wizard allows you to specify the name, description, branch location, and sensitivity level of your data store. More details below:

- > **Name** - the name of your data store. The name must be longer than two characters and up to 64 characters.
- > **Description** - the description for the data store (up to 250 characters).
- > **Branch Location** - select a branch location from the drop-down list. If no branch location is available, you have to create it. See ["Managing Branch Locations" on page 25](#) for details.
- > **Sensitivity Level** - select a sensitivity level from the drop-down list. A sensitivity level suggests to DDC what level of sensitivity is acceptable to find in this data store. For details, see ["Sensitivity Levels" on page 13](#).
- > **Enable Data Store** - when selected it means that this data store is available for scans. The **Enable Data Store** check box is selected by default. If the check box is cleared, the data store is disabled (not available) for scans.

Configuring a Data Store – Tags and Access Control

The **Add Tags & Access Control** screen in the **Add Data Store** wizard allows you to grant access rights to your data store and add tags. More details below:

- > **ACCESS** - select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store. The available options are:
 - **All groups**: All groups of users can access the data store through reports. This is the default setting.
 - **Selected group/s**: Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group. If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.
- > **TAGS** - select a tag from the **Add Tag** drop-down list. The predefined tags are: APA, APPI, CCPA, FINANCIAL, GDPR, HEALTH, HIPAA, KVKK, LEGAL, PCI, PERSONAL, PHI, PII, and SHIELD.

TIP New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down list. Add as many tags as needed. To remove a tag, click the close icon in the tag name.

Adding Database Stores

Use the **Add Data Store** wizard to add a database type data store. Adding a data store involves the following steps:

Select Store Type

1. In the left pane of the **Data Discovery** application, click **Data Stores**. The **Data Stores** page is displayed. This page lists available data stores.

2. On the right, click **+ Add Data Store**. The **Add Data Store** wizard is displayed.

The **Select Data Store** screen displays the following options to filter data store types:

- **Filter by Data Store category**: Shows categories of data stores. Click a category to filter available options under the **Select Type** drop-down list.
- **Select Type**: Shows types of data storage. By default, the drop-down list shows all types of data stores. The label **Select Type** changes to reflect the category selected under **Filter by Data Store category**. For example, for **Database**, the label becomes **Select Database Type**.

NOTE This document uses **Filter by Data Store category** to filter data stores.

3. Under **Filter by Data Store category**, click **Database**.
4. From the **Select Database Type** drop-down list, select a database. The available options are:
 - **IBM DB2**: Select to add an IBM DB2 database.
 - **Oracle**: Select to add an Oracle database
 - **Microsoft SQL**: Select to add a Microsoft SQL database.
5. Click **Next** to go on to the **Configure Connection** screen.

Configure Connection

1. The **Configure Connection** screen is displayed.
2. Specify **Hostname/IP** of the database server. Specify a valid hostname, IP address, or Uniform Resource Identifier (URI). The hostname must be longer than two characters. This is a mandatory field.
3. Specify **Port** of the database server. The port must be a number between 1 and 65535. The default ports are 50000 for IBM DB2, 1521 for Oracle, and 1433 for Microsoft SQL. This is a mandatory field.
4. Specify name of the **Database** service. This is a mandatory field. For an Oracle database, specify **Database or SID**.

NOTE If you are using **Oracle 12x**, or if the Oracle database displays a “**TNS: protocol adapter error**”, you must specify a **SERVICE_NAME** in the **Database or SID** field. For example: HR(SERVICE_NAME=XE)

5. Select an authentication method to connect to the database. The available options are:
 - **Credentials**: Select for password-based authentication. This is the default setting. Specify valid user credentials (**User** and **Password**) to access the network storage. For password-based authentication, valid user credentials are mandatory.
 - **Certificate**: Select for certificate-based authentication. Click **Choose File** to upload a valid certificate file. For certificate-based authentication, uploading a valid certificate is mandatory.

NOTE

- > The certificate must be in either **PEM** or **.p12** format.
- > The certificate size must be less than **30 KB**.

6. Click **Next** to go on to the **General Info** screen.

General Info

1. The **General Info** screen is displayed.
2. Specify a unique **Name** for the data store. The name must be longer than two characters and up to 64 characters.
3. Provide a **Description** for the data store (up to 250 characters).
4. Select a **Branch Location** from the drop-down list.
5. Select a **Sensitivity Level** from the drop-down list. A sensitivity level suggests to DDC what level of sensitivity is OK to find in this data store. For details, see ["Sensitivity Levels" on page 13](#).
6. Click **Next** to go on to the **Add Tags & Access Control** screen.

Add Tags & Access Control

1. The **Add Tags & Access Control** screen is displayed.
2. Under **ACCESS**, select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store.

The available options are:

- **All groups:** All groups of users can access the data store through reports. This is the default setting.
- **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group.

If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.

3. Under **TAGS**, select a tag from the **Add Tag** drop-down list. The prebuilt tags are APA, APPI, CCPA, FINANCIAL, GDPR, HEALTH, HIPAA, KVKK, LEGAL, PCI, PERSONAL, PHI, PII, and SHIELD.

TIP

- > New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down list.
- > Add as many tags as needed.
- > To remove a tag, click the close icon in the tag name.

4. Click **Save**.

The newly created data store appears on the Data Stores page. By default, data stores are displayed in alphabetic order by name. Depending on the number of entries per page, you might need to navigate to other pages to view the newly created data store.

Adding Big Data Stores

Use the Add Data Store wizard to add a big data type data store. Adding a data store involves the following steps:

NOTE In a Hadoop cluster:

- > Nodes where data blocks distributed by HDFS are stored are called DataNodes. DataNodes are treated as “slaves” in a Hadoop cluster.
- > A node that maintains the index of directories and files and manages data blocks stored on DataNodes is called a NameNode. A NameNode is treated as “master” in a Hadoop cluster.

Select Store Type

1. In the left pane of the **Data Discovery** application, click **Data Stores**. The **Data Stores** page is displayed. This page lists available data stores.
2. On the right, click **+ Add Data Store**. The **Select Data Store** screen of the **Add Data Store** wizard is displayed.

The **Select Data Store** screen displays the following options to filter data store types:

- **Filter by Data Store category**: Shows categories of data stores. Click a category to filter available options under the **Select Type** drop-down list.
- **Select Type**: Shows types of data storage. By default, the drop-down list shows all types of data stores. The label **Select Type** changes to reflect the category selected under **Filter by Data Store category**. For example, for **Big Data**, the label becomes **Select Big Data Type**.

NOTE This document uses **Filter by Data Store category** to filter data stores.

3. Under **Filter by Data Store category**, click **Big Data**.
4. From the **Select Big Data Type** drop-down list, select **Hadoop Cluster**.
5. Click **Next** to go on to the **Configure Connection** screen.

Configure Connection

1. The **Configure Connection** screen is displayed.
2. Specify **Hostname/IP** of the Hadoop cluster's active NameNode. Specify a valid hostname, IP address, or Uniform Resource Identifier (URI). The hostname must be longer than two characters. This is a mandatory field.
1. Click **Next** to go on to the **General Info** screen.

General Info

1. The **General Info** screen is displayed.

2. Specify a unique **Name** for the data store. The name must be longer than two characters and up to 64 characters.
3. Provide a **Description** for the data store (up to 250 characters).
4. Select a **Branch Location** from the drop-down list.
5. Select a **Sensitivity Level** from the drop-down list. A sensitivity level suggests to DDC what level of sensitivity is OK to find in this data store. For details, see ["Sensitivity Levels" on page 13](#).
6. Click **Next** to go on to the **Add Tags & Access Control** screen.

Add Tags & Access Control

1. The **Add Tags & Access Control** screen is displayed.
2. Under **ACCESS**, select user groups that can access the data store. Access to a data store provides ability to see reports that include scans of that data store.

The available options are:

- **All groups:** All groups of users can access the data store through reports. This is the default setting.
- **Selected group/s:** Specified user defined groups can access the data store through reports. When this option is selected, select a group from the drop-down list. This list shows existing user defined groups. The user defined groups must already exist on CipherTrust Manager. If no user defined groups exist, ask the administrator to create a group.

If needed, you can select multiple groups. Start typing the name of the desired group and select from the suggested groups.

3. Under **TAGS**, select a tag from the **Add Tag** drop-down list. The prebuilt tags are APA, APPI, CCPA, FINANCIAL, GDPR, HEALTH, HIPAA, KVKK, LEGAL, PCI, PERSONAL, PHI, PII, and SHIELD.

TIP

- > New tags can also be added. Start typing a new tag, and click the **New: <new_tag>** link that appears below the drop-down list.
- > Add as many tags as needed.
- > To remove a tag, click the close icon in the tag name.

4. Click **Save**.

The newly created data store appears on the Data Stores page. By default, data stores are displayed in alphabetic order by name. Depending on the number of entries per page, you might need to navigate to other pages to view the newly created data store.

Editing Data Stores

Existing data stores can be modified to suit your requirements. Use the edit view of the page to modify properties of data stores. You can edit the data store name, description, linked branch location, and applied sensitivity level. Additionally, connection settings, access rights, and tags can be modified.

To edit a data store:

1. In the left pane of the **Data Discovery** application, click **Data Stores**. The **Data Stores** page is displayed. This page lists available data stores.
2. Click the overflow icon () corresponding to the desired data store. A shortcut menu appears.

TIP Alternatively, to open the edit view a data store, click the **Name** link of the desired data store. Only the users with appropriate rights can see edit data store settings. All other users can only the settings.

3. Click **View/Edit**. The edit view of the **Data Stores** page appears.

NOTE Only the users with appropriate rights can see the **View/Edit** button. For all other users, only the **View** button is visible.

4. Expand **GENERAL**. General details are displayed.
5. Modify the required information.

NOTE The current data store type, which is displayed under **Select Type**, cannot be changed.

6. Expand **CONNECTION**. Connection settings are displayed. Based on the storage type, the displayed fields can be different.
7. Modify the required information.

NOTE When using the **Authentication** method, specify valid credentials in **User** and **Password**. To change the existing password, unlock the **Password** field by clicking the lock icon and enter the new password.

8. Click **Test Connection** to test the modified connection settings. If any error occurs, correct the connection settings and retry.

NOTE The **Test Connection** button is available only if a compatible Agent is found.

9. Expand **ACCESS**. The granted access rights are displayed.
10. Modify access rights under **Grant Access to**, if required.
11. Expand **TAGS**. The applied tags, if any, are displayed.
12. Add new tags or modify existing tags, as required.
13. Click **Save Changes**.

The list view of the Data Stores page shows updated information.

Automatic Agent Selection

Data stores that do not have a DDC Agent installed on the same host require using a DDC Agent as a proxy to get from the CipherTrust Manager appliance to the data store endpoint. To achieve this, data stores select agents automatically.

When a data store is added, the following situations can occur:

- > **DDC searches for a compatible agent:** When DDC searches for a compatible Agent, a rotating spinner next to the data store's name is displayed. If you hover the mouse over the spinner, "Waiting for Agent" is shown.
- > **DDC finds a compatible agent:** When a compatible agent is found, no spinner is seen next to the name. You can now test its connectivity with the Agent by clicking the "Test Connection" button inside the data store's settings. Refer to ["Editing Data Stores" on page 41](#) for details.
- > **DDC does not find a compatible agent:** DDC retries the agent selection for seven days. If cannot find a compatible agent in seven days, an error icon is displayed. If you hover the mouse over the icon, it states "Agent not available". The "Find Agent" button to relaunch the Agent selection is visible on clicking the overflow icon () next to the data store.

To relaunch automatic agent selection for a data store:

1. In the **Data Discovery** application, click the overflow icon () corresponding to the desired data store. A shortcut menu appears.
2. Click **Find Agent**.

NOTE

- > Instructions to install and configure DDC Agents can be found in the Thales CipherTrust Data Discovery and Classification Deployment Guide.
- > Port 11117 on the CipherTrust Manager appliance must be accessible from DDC Agent hosts.
- > Data store endpoint needs to be accessible from DDC Agent hosts.
- > To proxy requests to database stores, a Windows-based DDC Agent is required.
- > To proxy requests to Hadoop data stores, a Linux-based DDC Agent is required.
- > When the DDC Agent is properly identified, the data store status changes to ready. At this point, it is now possible to run scans against this data store.

Managing Scans

You manage scans through the **Scans** page, which is accessed by clicking the **Scans** link in the **Data Discovery** sidebar on the left.

From the **Scans** page you can:

- > View all currently available scans. See ["Viewing Scans" below](#).
- > Create a new scan. See ["Adding Scans" on the next page](#).
- > Execute a scan. See ["Running Scans" on page 48](#).
- > Delete a scan. See ["Removing Scans" on page 49](#).

Viewing Scans

The list view of the **Scans** page shows the number of:

- > Scans with the number of executed and unexecuted scans.
- > Executed scans with the number of scans containing sensitive and non-sensitive data.
- > Scanned data objects with the number of sensitive and other data objects.

Click the refresh button to refresh the displayed information.

The list view of the **Scans** page shows the following details:

Item	Description
Name	Name of the scan.
Profile	Number of classification profiles.
Schedule	Schedule of the scan.
Last Scan	Time when the scan last ran.
Duration	Time taken to complete the run.
Status	Status of the scan. The status could be Completed, In progress, Failed, Disabled, Stopped, Unscanned, Validating, or Pending.

TIP

- > Use the **Search** text box to filter scans. Search results display scans that contain specified text in their names.
- > By default, scans are listed in ascending alphabetic order of their names.
- > Scans can be sorted by their name, last scan time, duration, and status.

Adding Scans

To add a scan, navigate to the **Scans** screen (**Data Discovery > Scans**). Click the **+Add Scan** button to open the **Add Scan** wizard.

In the wizard, you have to go over these configuration steps for each scan that you add:

1. **"General Info" below** - Name the scan and give a short description.
2. **"Select Data Stores" below** - Select which data stores will be scanned.
3. **"Add Targets" on the next page** - Narrow down the scan scope by selecting specific scan targets.
4. **"Select Profiles" on the next page** - Choose which Classification Profile you want to scan for.
5. **"Schedule Scan" on page 47** - Configure when you want your scan to run.

General Info

1. In the **General Info** screen, the wizard asks you to specify a unique name for the scan and to give it a short description:
 - **Name** - The name must be longer than two characters and up to 64 characters.
 - **Description** - optional description of up to 250 characters.
2. Click **Next** to move on to the **Select Data Stores** screen.

Select Data Stores

The **Select Data Stores** screen lists all data stores in tabular form. By default, no data stores are selected. The table has three columns:

- **Data Store Name**: Lists available data stores (with their number).
- **Type**: The type of the data store, such as Local Storage, Network Share, etc.
- **Agent**: Displays the Agent that is connected to that data store. In this column, you can also see if the Agent is ready (that is, if the data store is ready).

To select a data store to scan:

1. Search for the desired data stores by specifying the search criteria in the search box. The search results will be displayed in the table under it.
2. Select a data store for the scan by selecting the corresponding check box. Similarly, select multiple data stores, if needed.

TIP Use the **Selected only** toggle switch to display only the selected data stores or all data stores (if the switch is 'off' all data sources are displayed).

3. Click **Next** to move on to the **Add Targets** screen .

Add Targets

In the **Add Targets** screen you can review a list of the data stores that you selected for the scan. By default, the scan will scan the entire data store, and this wizard step allows you to narrow down the scan scope by selecting specific targets for your selected data stores. The **Add Targets** screen is divided into three columns:

- **Data Store:** The list of selected data stores.
- **Targets:** Any selected specific target for the listed data store. "Full DS" indicates that no specific target has been selected, that is, the entire data store will be scanned. If you have added a scan target for the data store, it will be listed after you expand the data store row (by clicking the arrow button next to the data store name, on the left).
- **Add Target:** In this field you can type in a specific target and add it to the scan parameters. Scanning of this data store will be limited to the added target only.

NOTE

Any scan target that you add must be **valid**, otherwise the scan will fail. What is a valid scan target depends on the data store type, but here are a few tips to have in mind:

- > When adding scan targets for database data sources (IBM DB, Oracle, and MS-SQL):
 - Note that table names are case sensitive but schema names are not case sensitive.
 - Oracle data stores accept only tables as scan targets.
 - IBM DB and MS-SQL data stores accept schemas or tables as scan targets.
- > For Hadoop type data stores, you can configure a scan to use a specific Hadoop **file** as a scan target.

- > To add a scan target for a selected data store:
 - Type your scan target in the **Add Target** field.
 - Click the Apply button on the right of **Add Target** to add the target.
- > To remove a scan target for a selected data store:
 - Click the arrow button next to the data store name for which you want to remove a scan target.
 - Click the **Remove** link next to the scan target to remove it.
- > To move on to the **Select Profiles** screen, click **Next**.

Select Profiles

The **Select Profiles** screen lists all classification profiles in tabular form. By default, no profiles are selected. The table has three columns:

- **Classification Profile Name:** Lists available profiles.
- **Infotypes:** Displays the number of information types associated with the profile.
- **Sensitivity:** Displays the sensitivity level assigned to this classification profile. See "[Sensitivity Levels](#)" on page 13 for more information.

To select a classification profile for the scan:

1. Search for the desired profiles by specifying the search criteria in the search box. The search results are displayed in the table under it.

2. Select profiles for the scan by selecting the check boxes corresponding to desired profiles.

TIP Use the **Selected only** toggle switch to display only the selected classification profiles or all classification profiles (if the switch is 'off' all classification profiles are displayed).

3. Click **Next** to move on to the **Schedule Scan** screen.

Schedule Scan

1. In the **Schedule Scan** screen select the frequency with which you want the scan to run. The options are:

- **Manual:** Select to run the scan manually. This is the default setting.

NOTE If you select **Run Now**, the scan will be run just once after the scan is added successfully.

- **Scheduled:** Select to specify a schedule for the run. The scan will be run automatically on the specified schedule. When **Scheduled** is selected, the following fields appear on the screen:
 - **Increment:** Select the increment pattern of the run. This is a mandatory field. The options are **Daily**, **Weekly**, and **Monthly**. By default, **Daily** is selected.
 - **Every:** Specify when the run should repeat. This is a mandatory field.
For example, if Daily is selected as Increment, enter 2 to run the scan once every two days. If Weekly is selected as Increment, enter 2 to run the scan once every two weeks. Similarly, if Monthly is selected as Increment, enter 2 to run the scan once every two months.
 - **Time:** Specify the time when the run should start. This is a mandatory field. Specify the time in 12-hour format.
 - **Time Zone:** Select a time zone from the drop-down list.
 - **Starting:** Specify the day when the schedule should start. This is a mandatory field. By default, **Today** is selected. To specify a particular start date, select **On this date**, click the calendar icon, and select the date.
 - **Ending:** Specify the day when the schedule should end. This is a mandatory field. By default, **No End** is selected. To specify a particular end date, select **On this date**, click the calendar icon, and select the date.

NOTE A scan cannot run unless there is an identified Agent for every data store included in the scan. If it fails to run, check the status of different data stores included in the scan.

2. Click **Save** to complete adding the scan.

As a result, the newly created scan appears on the **Scans** page. By default, scans are displayed in alphabetic order by name. Depending on the number of entries per page, you might need to navigate to other pages to view the newly created scan. By default, the Status of a newly created scan is Unscanned.

NOTE If your CipherTrust Manager system clock does not match the Agent's system clock, your scans will not run as scheduled, so it is highly recommended to set up a NTP server to synchronize the clocks. This can be achieved in CipherTrust Manager through the **Admin Settings -> System -> NTP**. For details, refer to the "Thales CipherTrust Manager Administrator Guide".

Running Scans

To run a scan, navigate to the **Scans** screen (**Data Discovery > Scans**). Scans can be run either manually or automatically at a scheduled time.

- > To run a scan manually:
 - a. Search for the scan to run.

TIP

- > Use the **Search** text box to filter scans. Search results display scans that contain specified text in their names.
- > By default, scans are listed in ascending alphabetic order of their names.
- > Scans can be sorted by their name, last scan time, duration, and status.

- b. Move the mouse pointer to the row that contains the scan. The **Run Now** button appears. This button disappears as soon as the mouse pointer is moved out of the row.
- c. Click **Run Now**.

As soon as the scan is initiated, its status becomes **Pending**.

- > To configure a scan to run automatically, refer to the information in ["Schedule Scan" on the previous page](#).

Scan Statuses

The status of the scan changes in the sequence: Validating > Pending > Running now / Paused / Stopped > Processing > Completed / Failed.

Status	Description
Validating	Checking if all the data stores are ready.
Pending	Scan is pending and the linked data stores are being contacted. Depending on factors such as the network connectivity, this stage may: • Complete in a flash. You may not see it on the Scans page. • Remain for some time in this state.
Running now / Paused / Stopped	Scan is running, or is paused or stopped.
Processing	Scan is processing the collected data.

Status	Description
Completed / Failed	Scan run is successful or has failed.

Potential Problems When Running Scans

- > **Ready/Not Ready data store:** A scan cannot run unless there is an identified Agent for every data store included in the scan. Such a data store has the status **Ready**. A scan that has at least one data store that is **Not Ready** will fail to run, and display an error. If more than one data stores associated with a scan are **Not Ready** the system will fail on the first scanned data store that is **Not Ready** and will not check the remaining data stores.
- > **Disabled/Enabled data store:** You can manually deactivate a data store. Such a data store has a status **Disabled** and it will not be scanned. A scan that has several data stores associated will still run (without an error) even if one or more data stores are **Disabled** as long as at least one data store is enabled, but it will only scan the enabled data stores. A scan with all data stores **Disabled** will not run at all.
- > **Hadoop file access rights:** You get a **"data store path not accessible"** error when scanning a Hadoop data store that has a Hadoop file configured as its scan target, if you do not have access rights to that file.
- > **IBM, Oracle and MS-SQL - empty table or schema:** You get a **"table or schema not accessible"** error when scanning an empty table or schema.
- > **IBM, Oracle y MS-SQL - case sensitive table name:** In these data stores database schema names are not case sensitive, but table names are case sensitive.

Removing Scans

1. In the **Scans** screen, use the **Search** text box to filter scans and search for the scan that you want to remove.
2. Click the overflow icon () corresponding to the desired scan. An overflow menu is displayed, with a **View** and **Remove** options available.

NOTE The **Remove** option is not always available in the menu, only if a scan is **Failed**, **Completed**, **Stopped**, or **Disabled**.

3. Click **Remove** in the menu. As a result, a warning message **"Remove Scan? Are you sure you want to remove this scan?"** is displayed.
4. Click the **Remove** button in the warning message window to confirm the removal of the selected scan.

Managing Reports

You manage reports through the **Reports** page, which is accessed by clicking the **Reports** link in the **Data Discovery** sidebar on the left.

From the **Reports** page you can:

- > View all existing reports. See ["Viewing Reports" below](#).
- > Create a new report. See ["Creating Reports" on the next page](#).
- > Generate a report. See ["Generating Reports" on page 53](#).
- > View details of a selected report. See ["Report Details" on page 53](#).

Viewing Reports

The **Reports** page displayed lists available reports. Initially, the page shows no reports. Newly configured reports are shown on this page. Additionally, the page shows the total number of available reports.

By default, reports are listed in ascending alphabetic order of their names. The list view of the **Reports** page shows the following details:

Item	Description
Name	Name of reports.
Type	Type of reports.
Analysis	Analysis type is Aggregated.
Last Run	Time when the report was run.
Schedule	Schedule of the report run is Manual.
Status	Status of the report. The status could be Pending, Running now, Stopped, Processing, Completed, or Failed.

- > Use the **Search** text box to filter reports. Search results display reports that contain specified text in their names.
- > Reports can be sorted by their name, type (Scans), analysis (Aggregated), the last time that the scan was run, schedule, and status.
- > Click the link embedded in the report name to display the details of that report. For more information, see ["Report Details" on page 53](#).

Report Types

There are two different types of reports:

- **Static report:** This report is based on a scan run on a specific date.

- The report is 'frozen' with the last successful scan that has been executed up to now.
- In the case of multiple executions of the scan for the selected day, the report will include the information for the latest execution on that day.
- **Dynamic report:** This report is always showing the information found on the latest scan execution, so the results will change if the underlying data in the Data Store or the Classification Profile is modified. In other words, such a report is dynamic, it will be regularly updated whenever:
 - There are changes in the machine, such as the quantity of data has changed (increased or decreased).
 - There are changes in any aspect of the DDC configuration in DDC, such as when the classification profiles have been modified.

NOTE

- > For one report, it is possible to select scans in a dynamic and static manner. This would result in one report that is 'frozen' and another that is not 'frozen'.
- > In order to see an updated report you must run the scan since the report reflects the scan information.

Creating Reports

To create a report you aggregate data from multiple sources. When a report is generated it contains the results of executed scans.

To create a report use the **New Report** wizard described in the following sections. To launch the wizard, click the **+ Add Report** button in the **Reports** page on the right.

General Info

Provide the following information in the **General Info** screen of the **New Report** wizard:

- > A unique **Name** for the report. The name must be longer than two characters and up to 64 characters. This field is mandatory.
- > An optional **Description** for the report (up to 250 characters).

Click **Next** to go on to the **Configure Content** step of the wizard.

Configure Content

This **Configure Content** screen shows available scans with their number and the number of selected scans.

1. Use the **Search** text box to filter available scans. Search results display scans that contain specified text in their names.
2. Select the **Scan Name** check boxes corresponding to desired scans.
3. You can create two different types of reports:
 - A **static report** is based on a scan run on a specific date. For a static report, click "Latest Execution" and select the date of the scan that you wish to use.

- A **dynamic report** can change if the underlying data store or protection profile is modified and a scan is run again. For a dynamic report leave "Scan Execution" as "Latest Execution".

See ["Report Types" on page 50](#) for more information on these reports.

4. Click **Save** and a message will appear stating that the report has been created successfully.

The content identified in this step will be merged in a single report. This is called an **aggregated report**.

To create a report use the New Report wizard described in the following sections.

Name and Describe

1. In the **Reports** page on the right, click **New Report**. The **Name and Describe** screen of the **New Report** wizard is displayed.
2. Specify a unique **Name** for the report. The name must be longer than two characters and up to 64 characters. This field is mandatory.
3. Provide a **Description** for the report (up to 250 characters).

Configure Content

NOTE Content identified in this step will be merged in a single report. This is called an aggregated report.

1. Click **Next**. The **Configure Content** screen is displayed. This screen shows available scans with their number and the number of selected scans.
2. Use the **Search** text box to filter available scans. Search results display scans that contain specified text in their names.
3. Select the **Scan Name** check boxes corresponding to desired scans.
4. Create a static report or dynamic report:
 - For a static report - click **Latest Execution** and select the date of the scan that you wish to use.
 - For a dynamic report - leave **Scan Execution** as **Latest Execution**.

For information on these report types, see ["Report Types" on page 50](#).

5. Click **Save** and a message will appear stating that the report has been created successfully.

As soon as a new scan-based report is created, it is automatically run and results are visible. For information on the details of the report, see ["Report Details" on the next page](#).

NOTE

- > Scan-based reports are for personal use only. Users can only access the scan-based reports that they create. No user can access the scan-based reports created by others.
- > Selecting a scan forces DDC to retrieve the results discovered during the last execution of the given scan. As different scans may be taken into account, different runs of the same scan-based report may provide different results.

Generating Reports

After you have configured a report, it can be generated at anytime. Configured reports can be generated any number of times.

To generate a report:

1. In the **Reports** page, search for the report that you want to generate.

TIP

- > Use the **Search** text box to filter reports. Search results display reports that contain specified text in their names.
- > By default, reports are listed in ascending alphabetic order of their names.
- > Reports can be sorted by their name, type (Scans), analysis (Aggregated), last run time, schedule, and status.

2. Click the overflow icon () corresponding to the desired report. A shortcut menu appears.

3. Click **Run report**.

As soon as the report starts to run, its status becomes **Pending**. The status of the report changes in the sequence: Pending > Running now / Stopped > Processing > Completed / Failed.

NOTE Permissions to access the data stores accessed by the scans included in a scan-based report are checked every time the report is run. If the current user no longer has the correct permission for any of them, an error is displayed.

Report Details

The report details page displays such information about the report as the report name, the number of scans, data stores, and data objects. The page also shows the total data objects scanned, sensitive data objects found, sensitive data matches, and selected infotypes found.

NOTE There may be a mismatch between the number of objects scanned as shown in the **Total Data Objects Scanned** info card, top left of the **Data Objects** page and the number of objects listed in the table at the bottom ("**Showing _ of _**"). This is because the table also lists all so called "inaccessible" items that have been found in the scan.

Inaccessible items are data objects that could not be scanned because the Agent could not access their contents. The main reasons for this is a lack of permissions at the OS level (that is, the Agent is not able to read the file) and/or issues encountered when extracting the text data, such as file contents protected with a password, corrupted files, and the like.

The table in the report details lists the following findings distributed among these columns:

- **Object Name** - The name of the data object scanned and listed in the report details.
- **Risk** - The number of risks founds by the report in the given scanned data object. A risk is the presence of a sensitive item of data.

- **Type** - The type of the scanned data object listed in the report details, such as "File" or "Folder".
- **Path** - The path to the object that is listed in the report details.
- **Store** - The name of the data store where the object listed in the report details was found.
- **Owner** - The owner of the listed object.
- **Modified** - The date of the last modification of the scanned and reported data object.
- **Infotypes** - The number of information types found in the data object that is listed in the report.

NOTE Due to a known limitation of the processing engine, the information on the **Owner** and **Modified** is usually not listed in the report details.

To return to the **Reports** page, click **All Reports** at the top of the report details page.

Logging

DDC prints out its log messages to the CipherTrust Manager logs. CipherTrust Manager logs are located in the **/opt/keysecure/logs** directory. The CipherTrust Manager System Administrator (ksadmin) can log in using ssh to retrieve CipherTrust Manager logs. Also the DDC Application Administrators have access to the logs.

For more details on collecting DDC logs, see "Troubleshooting Issues in Conjunction with Customer Support" in the "Thales CipherTrust Manager Administrator Guide".

Default Logging Level

By default, log level setting for DDC is INFO. With this log level set DDC prints out the INFO and ERROR level messages to the log. Among the various messages that DDC prints to the logs, the error messages and security audit messages are the most useful for troubleshooting DDC issues and securing the deployment.

Identifying DDC Log Messages

The microservices behind DDC are **Oleander** and **Sundew** and the messages coming to the CipherTrust Manager log from DDC can be identified by those names.

Additionally, Oleander has these three modules:

- > Clustering
- > Agent_Selection
- > Scan_watcher

Each of these modules will generate its own error messages, each in its separate log.[] log file.

The logging service responsible for collecting and processing these messages is FLUENTD. It is capable of displaying those messages to the terminal through the log command. Here's an example of such a command:

```
> log | grep oleander | grep "clustering"
```

This command would display all messages coming from the Oleander's Clustering module.

For a complete list of error messages that DDC sends to the CipherTrust Manager log, see the appendix ["Error Log Messages" on page 63](#).

Security Audit Log Messages

The DDC security audit messages can be identified by the **Oleander | INFO [security]** bit that they contain. The full format of such a log message (or log line) is:

```
<date> | Oleander | INFO [security] <event> <error (if any)> <details (if any)>
```

For example:

```
2020-06-29 | Oleander | INFO | [security] DDCScanClientUnexpectedErrorProbe "error: error  
probing scan client" "details: [scan_id:5432-5432-543254-2-5432]"
```

Usually, only the event type is printed out to the log (in the example above, it would be DDCScanClientUnexpectedErrorProbe). You can find the full list of events with explanations in the appendix ["Security Audit Log Event Messages" on page 90](#).

Enabling Syslog Logging

Audit records are logged to a local database by default. This is suitable for production systems and clusters with a limited load. However, for clusters that support a large number of transactions, it is recommended to configure the CipherTrust Manager Manager to disable logging to a local database and enable logging using a remote Syslog server. This significantly reduces cluster traffic and disk usage. For more information, refer to the following sections in the “Thales CipherTrust Manager Administrator Guide”:

- > “Disabling local database audit logging”
- > “Configuring remote Syslog server”
- > “Configuring Connection to a Syslog Server”

APPENDICES

Error Messages

This section lists the various error messages that the system can display, with explanations and solutions (if available).

Error Message	Explanation
Branch Locations	
"Branch Location name already exists" (message on toast)	You tried to create a branch location with a name that is already taken by another branch location. SOLUTION: Choose another name.
Data Stores	
"Data Store name already exists" (message on toast)	You tried to create a data store with a name that is already taken by another data store. SOLUTION: Choose another name.
"A valid agent could not be found" (Agent selection - on mouse-over on the data store)	There is no active agent for this data store that the automatic agent selection process has been able to detect. SOLUTION: This requires additional research, such as checking if the agent is installed on the data store, if it has the right type (local/proxy), if it is of the right OS "flavor" (Linux, Windows), or of the right type (e.g. database). Refer to the "Deployment Guide" for more information on troubleshooting this issue.
Scans	
"Scan name already exists" (message on toast)	You tried to create a scan with a name that is already taken by another scan. SOLUTION: Choose another name.
"All Data Stores are disabled" (message on toast)	You attempted to run a scan that has all data stores disabled. SOLUTION: Enable at least one data store for the scan. Refer to the "Managing Scans" on page 44 chapter for instructions.

Error Message	Explanation
"The following Data Stores are not accessible: <xyz>" (message on toast)	You tried to scan a data store that is not accessible. The scan is marked as Failed, and includes a warning icon with the message "The data store <xyz> included in the scan is not accessible" on mouse-over. SOLUTION: Verify the connectivity from the agent to the data store. Verify the data store configuration.
"One or more Data Stores are not accessible." (On mouse-over on the scan fail icon)	The scan failed because the data store that is configured is inaccessible. The data store failed after the scan was launched. SOLUTION: There may be a number of reasons for this. To troubleshoot a failed data store refer to "Managing Data Stores" on page 32.
"The following Data Stores have no agent available: <xyz>" (On mouse-over on the scan fail icon)	You tried to scan a data store that had no agent available when the scan was executed. There is a problem with the agent. The data stores that failed are listed. SOLUTION: This requires additional research, such as checking if the agent is installed on the data store, if it has the right type (local/proxy), if it is of the right OS "flavor" (Linux, Windows), or of the right type (e.g. database). Refer to the "Deployment Guide" for more information on troubleshooting this issue.
"Data Store has incorrect credentials" (on mouse-over on the scan fail icon)	Data store credentials provided are incorrect so the scan cannot be executed. These data stores are listed. SOLUTION: Update the server credentials for the data store.
"One or more Data Stores have incorrect credentials" (message on toast)	The credentials for one or more data stores are no longer valid (credentials modified, user deleted, and so on) preventing the scan from completing. SOLUTION: Reconfigure the data store and re-launch the scan.
"The scanner service is not available" (message on toast)	You tried to run a scan with the scan engine unavailable. SOLUTION: Check the status of the scan engine (the KeySecure server).

Error Message	Explanation
"The following Data Stores have missing agents: <xyz>" (message on toast)	This happens when an agent was assigned to the listed data store(s) and then when a scan was launched, for some reason the assigned agent could not be found on the server. SOLUTION: Try to re-assign the agent in the data stores screen. If this does not work, check the agent assigned to the <xyz> data store.
"The following Data Stores have agent errors: <xyz>" (message on toast)	This happens when a management request for an agent fails (for example, at verification or when setting it as a proxy) during the scan execution. SOLUTION: It is usually a transient issue. Wait a few minutes and run the scan again. If it still fails, check the agent status.
"Error processing scan" (message on toast)	This happens when the scan fails in the processing stage, that is when the scan results are being processed by DDC. SOLUTION: Run the scan again. If the error persists, contact Thales support.
"Error connecting to HDFS" (message on toast)	This happens when the scan fails because there is no HDFS connectivity. SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the "Deployment Guide" for information on configuring the DDC-HDFS connection.
"Error connecting to PQS" (message on toast)	This happens when the scan fails because there is no PQS connectivity. SOLUTION: Check the HDFS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the "Deployment Guide" for information on configuring the DDC-PQS connection.
"Error checking the data allowance" (message on toast)	This happens when DDC is not licensed. DDC sends a request for data allowance to the license server and the server responds that there is no license. SOLUTION: Obtain and install a valid DDC license.

Reports

Error Message	Explanation
"Report name already exists" (message on toast)	You tried to create a report with a name that is already used in another report. SOLUTION: Choose another name.
Classification Profiles	
"Classification Profile name already exists" (message on toast)	You tried to create a classification profile with a name that is already taken by another classification profile. SOLUTION: Choose another name.
Licensing	
"DDC License not found - try again in a few minutes if you recently inserted one" (message on toast)	Any action performed in the UI results in this message, because there is no valid DDC license installed. SOLUTION: Obtain and install a valid license.
"DDC License expired" (message on toast)	Any action performed in the UI results in this message, because your DDC license has expired. SOLUTION: Obtain and install a valid license.
Hadoop Configuration	
"Hadoop is not active. Please go to DDC Settings --> Hadoop" (message on toast)	Problem communicating with Hadoop or DDC has not been configured with Hadoop. SOLUTION: Assuming that you have Hadoop deployed in your environment, configure DDC to use it (DDC Settings --> Hadoop in the KeySecure UI). For a detailed procedure, refer to the "Deployment Guide".
"Error connecting to the PQS database" (message on toast)	Problem communicating with the Phoenix Query Server database (i.e. HBase). SOLUTION: Check the PQS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the "Deployment Guide" for information on configuring the DDC-PQS connection.

Error Message	Explanation
"Error creating the PQS database schema" (message on toast)	Problem communicating with the Phoenix Query Server database. SOLUTION: Check the PQS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the "Deployment Guide" for information on configuring the DDC-PQS connection.
"Error using the PQS database schema" (message on toast)	Problem communicating with the Phoenix Query Server database. SOLUTION: Check the PQS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the "Deployment Guide" for information on configuring the DDC-PQS connection.
"Error connecting to HDFS" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the "Deployment Guide" for information on configuring the DDC-HDFS connection.
"Invalid HDFS directory path: Not a directory" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the "Deployment Guide" for information on configuring the DDC-HDFS connection.
"Incorrect credentials in the HDFS connection" (message on toast)	SOLUTION: 1. Check the HDFS configuration in DDC (Hadoop Services). Refer to the "Deployment Guide" for information on configuring the DDC-HDFS connection. 2. Check that the authentication service is up and running.
"Incorrect HDFS URI" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the "Deployment Guide" for information on configuring the DDC-HDFS connection.
"Invalid HDFS folder: the path to the folder does not exist" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the "Deployment Guide" for information on configuring the DDC-HDFS connection.
"Invalid server certificate in the HDFS request" (message on toast)	SOLUTION: Check the HDFS configuration in DDC (Hadoop Services). Refer to the "Deployment Guide" for information on configuring the DDC-HDFS connection.

Error Message	Explanation
"Incorrect credentials in the PQS connection" (message on toast)	SOLUTION: 1. Check the PQS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the "Deployment Guide" for information on configuring the DDC-PQS connection. 2. Check that the authentication service is up and running.
"Invalid server certificate in the PQS request" (message on toast)	SOLUTION: Check the PQS configuration in DDC (Hadoop Services) or PQS/Hadoop configuration in your Hadoop deployment. Refer to the "Deployment Guide" for information on configuring the DDC-PQS connection.
"Your system does not meet the 16GB RAM minimum" (message displayed across the bottom of all DDC screens)	DDC requires at least 16GB of RAM to be able to run properly. SOLUTION: Increase the RAM memory on board to at least the required minimum of 16GB.

Error Log Messages

This section lists the various error messages that DDC sends to the KeySecure log.

Error Message	Comment/Explanation
OLEANDER ERRORS	
2020-02-20 19:53:08 oleander Error connecting to the scan service	This error means a connectivity issue between the oleander and the sundew/ER2 .
2020-02-13 08:50:17 oleander DDC Error creating the database schema.: error executing http request. Code: 500 - Body:<html>	To check the connectivity with Hadoop external data base a schema is created. So this means that there is no connectivity with Hadoop. (also, possibly related to "Error creating the PQS database schema" in the UI)
"CLIENT_CREDENTIAL_PARTITION is not set"	CLIENT_CREDENTIAL_PARTITION variable is not set in the config object.
"[Background-Processes] Error retrieving license from DMV", "error", err	Oleander GetLicenses request against DMV has failed. This error could have been caused by DMV being down.
"[Background-Processes] Error killing all agents selections"	When the oleander instance loses its license or the current license expires, all ongoing agent selections will stop. This error is caused by an internal Oleander issue while this agent selections are being shut down.
"[Background-Processes] Error killing all scan watchers"	When the oleander instance loses its license or the current license expires, all ongoing scan tracking will stop. This error is caused by an internal Oleander issue during this scan tracking shut down.
"[Background-Processes] Error removing all scan schedules due to DDC not licensed"	When the oleander instance loses its license or the current license expires, all scheduled scans will be stopped. This error is caused by an internal Oleander issue during this scheduled scans stopping.
"[Background-Processes] Error starting agent selection for datastores"	The listed Data Stores have no agent available.
"[Background-Processes] Error starting scans watchers"	When the oleander instance receives a valid license from DMV, all stopped agent selections must be resumed. This error is caused by an internal Oleander issue during this agent selections resuming.

Error Message	Comment/Explanation
"[Background-Processes] Error starting cron schedules"	When the oleander instance receives a valid license from DMV, all stopped scheduled scans must be resumed. This error is caused by an internal Oleander issue during this scheduled scans resuming.
"[Background-Processes] Error trying to migrate the PQS database"	During the PQS configuration, an error occurred trying to apply the changesets to update the Database, create the tables, etc...
"[Background-Processes] Error creating cron", "error", err	Background_processes service constantly creates crons for license status checking against DMV, this error is caused by an internal Oleander issue during the creation of one of this crons.
"[Background-Processes] Error with unmarshal."	Any "unmarshal/unmarshalling" error is caused by an internal Oleander issue converting a golang object to JSON format or vice versa.
"[Background-Processes] Error while trying to update status to FAILED for datastore.", "name", ds.Name, "error", err	When background_process services gets an invalid or expired license from DMV all running scans must be stopped and set as FAILED, this message indicates an internal Oleander error changing the scan status for some scan.
"[Background-Processes] Error while trying to retrieve scans from background processes table"	When the oleander instance receives a valid license from DMV, all stopped normal scans must be resumed. This error is caused by an internal Oleander issue accessing the background processes table, which contains all the information for the scans resuming.
"[Background-Processes] Watcher has failed updating scan status for scan", "name", sc.Scan.Name, "error", err	background_processes service has the responsibility to track the running scans and update their Oleander status, this error message indicates an internal Oleander issue updating the scan_process table.
"Cannot retrieve the HDFS settings", "error", err	Connectivity or internal Oleander error trying to retrieve HDFS settings from Citrus.
"Cannot retrieve the PQS settings", "error", err	Connectivity or internal Oleander error trying to retrieve PQS settings from Citrus.
"Cannot find the country", "error", err	When a user creates a BranchLocation and he sets a country that is not registered in our DB.
"Cannot find the state", "error", err	When a user creates a BranchLocation and he sets a state ID that is not registered in our DB.

Error Message	Comment/Explanation
"Trying to verify if the country has states", "error", err	When a user creates a BranchLocation and he sets a state "Name" that is not registered in our DB.
"Missing tag %s for the default classification profiles", r	When inserting default classification profiles, there is no tag that matches the correct Regulation
"[Datastores] Error encrypting connection for datastore: ", "Name", d.Name	Error when calling Scrim Helper for the encryption of the Connection field for a datastore.
"[Datastores] Error while trying to create background process resource for datastore: ", "Name", d.Name	When a datastore is created, a new row is inserted into the Background Processes table for further tracking, this error message indicates an internal Oleander issue inserting that row.
"[Datastores] Error selecting agent for datastore ", "Name", dsAAS.Name, "error", err	No suitable agent has been found for this datastore.
"[Datastores] Error while trying to retrieve datastore for background process, agent selection might fail: ", "error", err	Error retrieving a datastore from DB for background processes purposes in a datastore.update operation.
"[Datastores] Error running automatic agent selection", "error", errAgentUpdate	An internal Oleander error has occurred while trying to update the status of a datastore during the automatic agent selection.
"[Datastores] Error while trying to unmarshal datastore from background processes"	An internal Oleander error has occurred while trying to translate a JSON object to a goLang object while recovering the automatic agent selections.
"Error closing the json file", "error", err	Oleander failed trying to close a JSON file.
"Error trying to close the families json file"	Oleander reads a families JSON file for startup DB population. This error message indicates an internal Oleander error while closing this file.
"Error initializing the account", "error", *errPtr	Oleander failed trying to set the initialization status to the accounts map.
"Error closing oleander", "error", err	Oleander service could not be closed.
"Error starting background processes", "error", err	Some error has occurred trying to execute all background processes.
"Error connecting to HDFS", "error", err	This happens when the scan fails because there is no HDFS connectivity.

Error Message	Comment/Explanation
"Error trying to close the info types json file"	Oleander reads an infotypes JSON file for startup DB population. This error message indicates an internal Oleander error while closing this file.
"Error connecting to PQS", "error", err	This happens when the scan fails because there is no PQS connectivity.
"Error closing the temporary file", "FileName", f.Name(), "error", err	Oleander failed closing the temporary file used for decrypting the raw data file.
"Error deleting the temporary file", "FileName", f.Name(), "error", err	Oleander failed deleting the temporary file used for decrypting the raw data file.
"Error trying to close the zip file", "error", merry.Wrap(err).WithHTTPCode(http.StatusBadRequest)	Oleander failed closing the zip file used for decrypting the raw data file.
"Error changing the scan status", "error", err	Internal Oleander error while trying to update the scan status in the Scan Process table.
"[Scan-Launcher] Error while trying to create background process resource for datastore", "name", rsc.Name	Oleander inserts a row in the Background Processes table for further scan tracking - this message indicates an internal Oleander error while performing this insert.
"[Scan-Launcher] The scan watcher returned an error", "error", err	Generic error message for any issue during the scan watcher process.
"[Scan-Launcher] Agent not found for datastore", "DS name", s.ScanDatastores[i].Datastore.Name, "error", err	The listed Data Stores have no agent available.
"[Scan-Launcher] Error getting absolute paths", "DS name", s.ScanDatastores[i].Datastore.Name, "error", err	Oleander was unable to retrieve the absolute paths of the mentioned datastore for further scan execution.
"[Scan-Launcher] Error getting connection path", "DS name", s.ScanDatastores[i].Datastore.Name, "error", err	Oleander was unable to retrieve the connection paths of the mentioned datastore for further scan execution.
"[Scan-Launcher] Error retrieving the oleander context"	Error while generating the Oleander service user. (this is the context for executing actions on behalf of the itself, instead of a specific user).
"[Scan-Launcher] Error while validating the scan"	This is a generic error, one of the steps of the validation has failed. (could be that some datastores are not ready, a probe has failed, etc)

Error Message	Comment/Explanation
"[Scan-Launcher] Error while retrieving extensions"	The extensions necessary for the scan execution could not be retrieved from the DB.
"[Scan-Launcher] Unable to initialize background process"	The background process object could not be initialized so the scan watcher cannot start.
"[Scan-Actions] Error found while trying to delete background process resource"	When Oleander has finished tracking a scan it removes the corresponding row from background processes table - this error message indicates an internal Oleander issue removing that row.
"[Scan-Actions] Error trying to execute the Scan scheduled run", "error", err	Oleander failed trying to add a scan schedule for a scan.
"Error changing the scan status", "error", err	Generic error message for an issue during the update of the scan status.
"[Scan-Watchers] Watcher has failed getting scan status from scanned service", "scan", sc.Scan.Name, "error", err	This indicates an underlying connectivity issue with Sundew.
"[Scan-Watchers] Watcher has failed updating scan status", "scan", sc.Scan.Name, "error", err	Error updating the scan status in the scan process table.
"[Scan-Watchers] Error processing the report: ", "error", err	Oleander scan collector has failed trying to create the scan collector background process for the mentioned scan.
"[Scan-Watchers] Scan aborted, maximum wait for scan results exceeded", "MaxTimeInterruptedState (minutes)", o.Config.Er2MaxInterruptedTime/60, "scan", sc.Scan.Name, "current status", sc.Scan.ScanProcess.Status, "scan service status", er2MappedStatus	The scan has been in INTERRUPTED er2 state for too long, when it exceeds the timeout the status is changed to FAILED in DDC.
"Error trying to parse er2 polling frequency from env"	Oleander failed trying to read the ER2 POLLING FREQUENCY env variable from the config object.
"Error while trying to retrieve scans from background processes table", "error", err	Oleander tried to retrieve all SCAN rows in the background processes table but it failed.
"Error trying to recover the scan watchers", "error", err	One of the scan watchers instantiated by the recovery system has returned an error and stopped working.
"Wrong datastore credentials"	Oleander was unable to reach a datastore due to credentials failure.

Error Message	Comment/Explanation
"Wrong target path defined"	A scan has failed because the target path defined was not valid.
"Wrong db schema in target path"	A scan has failed because the db schema defined was not valid.
"Wrong db table in target path"	A scan has failed because the DB defined was not valid.
"Wrong defined file extension in target path"	A scan has failed because a file was specified without an extension.
"Error connecting to the scan service"	Oleander is unable to connect with ER2.
"Error processing the scan reports"	Generic error while trying to process a scan report.
"No Data Allowance Licensing detected"	Oleander does not have a Data Allowance record.
"Error reading the scan report"	Generic error while trying to read a scan report.
"Probing path can be launched only on folders. Files are not supported"	Scan failed because a file was specified as target instead of a folder.
"Probing a File or Directory that does not exist"	Scan failed because the specified directory does not exist.

OLEANDER INFO

"[Background-Processes] This node was NOT selected as active node. Turning off background processes"	Related to clustering, self-explanatory error.
"[Background-Processes] This node was selected as active node. Turning on background processes"	Related to clustering, self-explanatory error.
"[Background-Processes] Recovering collectors"	Oleander has received a valid license and recovers the collector processes that were stopped.
"[Background-Processes] Updating license from DMV"	Oleander is requesting from DMV the licenses available for DDC.
"[Background-Processes] Global license status set to nil"	Oleander is unlicensed.
"[Background-Processes] Global license status set to ", "newLicenseStatus", *newLicenseStatus"	Oleander license status is whatever <i>newLicenseStatus</i> is.

Error Message	Comment/Explanation
"[Background-Processes] Checking HDFS connectivity"	Oleander is sending a ping to HDFS to check the connectivity.
"[Background-Processes] Cannot connect with HDFS", "error", err	Oleander's ping against HDFS has failed. Oleander has no connectivity with HDFS.
"[Background-Processes] HDFS connectivity successful"	Oleander has successfully performed a ping against HDFS. Oleander has connectivity with HDFS.
"[Background-Processes] Checking PQS connectivity"	Oleander is sending a ping to PQS to check the connectivity.
"[Background-Processes] Cannot connect with PQS", "error", err	Oleander's ping against PQS has failed. Oleander has no connectivity with PQS.
"[Background-Processes] PQS connectivity successful"	Oleander has successfully performed a ping against PQS. Oleander has connectivity with PQS.
"[Background-Processes] License status has changed"	Oleander license status has changed since the last license cron execution.
"[Background-Processes] Initializing fast cron: Oleander unlicensed", "UnlicensedCronFrequency", o.Config.UnlicensedCronFrequency	Oleander is unlicensed so the license cron (which asks DMV for a DDC license) increments its frequency. The frequency is defined in the docker-compose file.
"[Background-Processes] Initializing fast cron: waiting for Hadoop connectivity", "HadoopConnectivityCronFrequency", o.Config.HadoopConnectivityCronFrequency	Oleander has a valid license but does not have connectivity with Hadoop.
"[Background-Processes] Initializing slow cron: Oleander licensed and Hadoop connectivity successful", "RunningCronFrequency", o.Config.RunningCronFrequency	Oleander has a valid license and has connectivity with Hadoop, the license cron decrements its frequency. The frequency is defined in the docker-compose file.
"[Background-Processes] About to run NO LICENSE scenario"	Oleander has no valid license so it will stop all running scan watchers, automatic agent selections and scan schedules.
"[Background-Processes] About to run VALID LICENSE scenario"	Oleander didn't have a valid license but now it has one so all scan watchers, automatic agent selections and scan schedules that were stopped are being resumed.
"[Background-Processes] Recovering automatic agent selection"	Part of the VALID LICENSE scenario above.

Error Message	Comment/Explanation
"[Background-Processes] Starting automatic agent selection for pending datastores", "pending datastores", strings.Join(ds, ", ")	Part of the VALID LICENSE scenario above.
"[Background-Processes] Recovering watchers"	Part of the VALID LICENSE scenario above.
"[Background-Processes] Starting watchers for ongoing scans", "ongoing scans", strings.Join(ss, ", ")	Part of the VALID LICENSE scenario above.
"[Background-Processes] Recovering scan schedules"	Part of the VALID LICENSE scenario above.
"[Background-Processes] Migrating PQS database"	Self-explanatory message.
"[Background-Processes] Deleting background process for scan", "name", sc.Scan.Name	Oleander has no valid license so all scans are being stopped as well as their corresponding background processes.
"[Background-Processes] Updating status to FAILED for scan", "name", sc.Scan.Name	Oleander has no valid license so all scans are being stopped (with status FAILED).
"Cannot connect to HDFS", "settings", hdfsSettings, "error", err	Oleander's ping against HDFS has failed. Oleander has no connectivity with HDFS.
"Cannot connect to PQS", "Settings", pqsSettings, "error", err	Oleander's ping against PQS has failed. Oleander has no connectivity with PQS.
"[Datastores] Agent selected: ", "Name", a.Name	Oleander has found and assigned a suitable agent for the mentioned datastore.
"Unable to connect with datastore", "datastore name", d.Name, "error", err	A probe against the mentioned datastore has failed during a test connectivity check.
"Instantiating new scrim helper"	Oleander is instantiating a new ScrimHelper object which is used for communication with Scrim, Minerva, Sallyport and DMV.
"Instantiating new hdfs scan collector"	Oleander is instantiating a new HDFSCollector object which is used for communication and processing with HDFS.
"Instantiating scheduler cron"	Oleander is instantiating a new SchedulerCron and starting the background processes.
"[WARNING] PQS connector can not be closed in GetSummaryReport service"	Oleander was unable to close the PQS connector while executing a GetSummaryReport operation.

Error Message	Comment/Explanation
"[WARNING] PQS connector can not be closed in GetDatastoresDetailsReport service"	Oleander was unable to close the PQS connector while executing a GetDatastoresDetailsReport operation.
"[WARNING] PQS connector can not be closed in GetReportTemplate service"	Oleander was unable to close the PQS connector while executing a GetReportTemplate operation.
"[Scan-Actions] [WARNING] PQS connector can not be closed in GetScanExecutions service"	Oleander was unable to close the PQS connector while executing a GetScanExecutions operation.
"[Scan-Watchers] Stop watcher signal received", "scan", sc.Scan.Name	A running scan watcher has received a STOP signal and the current execution is cancelled.
"[Scan-Watchers] Watcher has failed stopping scan from scanned service"	The scan has been in an INTERRUPTED status for too long so Oleander has tried to stop the scan in ER2 but the request was unsuccessful.
"[Scan-Watchers] Watcher has detected different status", "scan", sc.Scan.Name, "current status", sc.Scan.ScanProcess.Status, "scan service status", er2MappedStatus	The scan watcher has detected a change in the scan status, scan status will be updated in DDC.
"[Scan-Watchers] Interrupted status received", "MaxTimeInterruptedState", o.Config.Er2MaxInterruptedTime, "scan", sc.Scan.Name, "current status", sc.Scan.ScanProcess.Status, "scan service Mapped status", er2MappedStatus, "scan service status", er2Status, "InterruptedTimestamp", sc.InterruptedTimestamp	The scan watcher has received an INTERRUPTED status for the scan, the scan watcher will continue asking until the status changes or the timeout is exceeded.
"[Scan-Watchers] Scan has finished... collector starting", "scan", sc.Scan.Name	The scan watcher has received a COMPLETED status for the scan, the scan status is set to PROCESSING in DDC and the scan collector starts.

SUNDEW INFO

"Asking for license status"	Sundew is requesting from Oleander the DDC license status.
"Active license information received", "http ret code", c	Sundew has received a valid license status from Oleander.
"Starting scan service..."	Sundew is launching ER2.
"Connectivity test successful, scan service is up"	Sundew confirms that ER2 is up-and-running.

Error Message	Comment/Explanation
"Generating scan service license...", "ID", s.er2Status.lastIssuedLicenseID	Sundew is generating a license for ER2.
"Injecting license into scan service"	Sundew is injecting the generated license into ER2.
"License won't be generated/refreshed in this iteration", "ID", s.er2Status.lastIssuedLicenseID, "checks (intervals)", s.Config.GeneratingLicenseInterval-s.er2Status.lastIssuedLicense	No action is required from Sundew for this iteration regarding licensing.
"Not active license or product not licensed response received", "http ret code", c	Sundew has received an invalid license status from Oleander.
"Scan service stopped"	Sundew is stopping ER2.
"Unexpected error code in response for /license/status", "http ret code", c	Sundew asked Oleander for the license status but got an unexpected response.

SUNDEW ERROR

"CLIENT_CREDENTIAL_PARTITION is not set."	CLIENT_CREDENTIAL_PARTITION variable is not set in the config object.
"Error trying to ask for license status", "error", err	Sundew failed trying to request DDC status from Oleander.
"Error trying to start the scan service", "error", err	Sundew failed trying to start ER2.
"Connectivity test to the scan service failed", "error", conErr	Sundew failed trying to ping ER2. ER2 is down.
"Error during validation of received license", "error", err	Sundew failed trying to validate the DDC license retrieved from Oleander.
"Error parsing the license ID", "error", err	Sundew failed trying to parse the DDC license ID received from Oleander.
"Error while injecting license inside scan service", "error", err	Sundew failed trying to inject the generated license into ER2.
"Error stopping scan service", "error", err	Sundew failed trying to stop ER2.
"Error closing the response body", "error", err	Sundew failed trying to close the HTTP response body from a request.
"Error closing sundew", "error", err	Error when trying to close the Sundew service.

Reconfiguring Agents

In some situations, for example, if the hostname or IP address of the CipherTrust Manager appliance changes, Agents' connection with DDC must be reconfigured with the new hostname or IP address.

Reconfiguring DDC Agents on Windows

To reconfigure a DDC Agent:

1. Log on to the host machine as administrator.
2. Open Enterprise Recon Configuration Tool (**er2_config_cmd.exe**).
By default, the tool is available at C:\Program Files (x86)\Ground Labs\Enterprise Recon 2\.
3. In the **Master server IP address or host name** field, specify the new hostname or IP address of the CipherTrust Manager.
4. Click **Test Connection**. A message stating "Connectivity test is successful" confirms successful reconfiguration.
5. Click **Finish**.

Reconfiguring DDC Agents on Debian

To reconfigure a DDC Agent:

1. Log on to the host machine as a user with root privileges.
2. Reconfigure connection with DDC on the CipherTrust Manager appliance.

```
sudo er2-config -i <hostname|ip_address>
```


Here, <hostname|ip_address> represents the new IP address or hostname of the CipherTrust Manager appliance.
3. Restart the Agent service. Configuration settings will be effective after the Agent restarts.

```
sudo /etc/init.d/er2-agent -restart
```

Reconfiguring DDC Agents on RHEL

To reconfigure a DDC Agent:

1. Log on to the host machine as a user with root privileges.
2. Reconfigure connection with DDC on the CipherTrust Manager appliance.

```
er2-config -i <hostname|ip_address>
```


Here, <hostname|ip_address> represents the new IP address or hostname of the CipherTrust Manager appliance.
3. Restart the Agent service. Configuration settings will be effective after the Agent restarts.

```
/etc/init.d/er2-agent restart
```

Restarting DDC Agents

Restarting Agents on Windows

To restart a DDC Agent, run the following commands:

```
net stop "Enterprise Recon 2 Agent (<ARCH>)"
net start "Enterprise Recon 2 Agent (<ARCH>)"
```

Here, <ARCH> represents the Windows architecture - x32 or x64.

Restarting Agents on Debian

To restart a DDC Agent, run: `sudo /etc/init.d/er2-agent restart`

TIP Alternatively, restart the Agent service by stopping it and again starting it manually. Run the following commands:

```
sudo /etc/init.d/er2-agent stop
sudo /etc/init.d/er2-agent start
```

Restarting Agents on RHEL

To restart a DDC Agent, run: `#!/etc/init.d/er2-agent restart`

TIP Alternatively, restart the Agent service by stopping it and again starting it manually. Run the following commands:

```
#!/etc/init.d/er2-agent stop
#!/etc/init.d/er2-agent start
```

Mounting an NFS Share

To mount an NFS share on a Proxy agent, run this command as root:

```
# mount <nfs-server-hostname|nfs-server-ipaddress>:</target/directory/share-name>
```

REST API

You can use the REST interface from the *API playground*, or via any REST client such as curl. To use the REST interface from the API playground, acquire an authorization token and use it when making API calls.

Acquiring an Authorization Token

To acquire a token:

1. Open the CipherTrust Manager URL in a browser. The login page is displayed.
2. Click the **API & CLI Documentation** link. The API playground is displayed.
3. At the top right, click **Authenticate**.
4. Enter **username** and **password**.
5. Click **POST**.

NOTE This acquires an API token and prefills it in the playground examples. The token expires in 300 seconds (5 minutes). When it expires, use this process again to acquire a new token.

On successful token generation, the remaining token expiry time in seconds is displayed. Two new buttons, **Clear Credentials** and **Re Authenticate**, are also displayed at the top right.

Using the Token

The authorization token acquired above is used in examples in the API playground. The token expires in 5 minutes; if expired, generate a new token, as described in ["Acquiring an Authorization Token" above](#).

Making an API Call

To make an API call, in the left pane, find the API and click it. In the right pane, specify the values of the required parameters, and click appropriate button (for example, **POST**, **GET**, **DELETE**, and **Curl**, as appropriate) in the playground. For example, to create a branch location on the CipherTrust Manager:

1. In the left pane of the API playground, click **DDC**.
2. Under **ddc/system-settings/branch-locations**, click **Create**. The **Create** section of the API playground is displayed in the right pane.
3. In the **body** field, specify the parameters with their values, as shown below.

```
{
  "name": "mybranch",
  "city": "Paris",
  "countryId": "18faaf74-c511-4086-a5fb-8062ecf2d8f4",
  "stateId": ""
}
```

Expand **schema** under the **body** field for names and types of fields. Mouse over each field to view its description. The parameter names and casing in the **body** field must match with those shown in the schema. Also, ensure that parameters and their values are specified in double quotes.

4. Click **POST**.

Alternatively, to get an equivalent `curl` command, click the **Curl** button. The `curl` equivalent will be shown in the text field below. Use the `curl` tool to run the command to make the REST API call.

Similarly, all API calls can be made by referring to the schema shown in the playground.

CLI

The CipherTrust Manager includes a CLI tool, named **ksctl**, that can be downloaded and run locally to control a remote CipherTrust Manager appliance. **ksctl** exclusively uses the REST API to communicate with the CipherTrust Manager, so anything that you can do with the tool, you can also do directly with the REST API. Conversely, **ksctl** exposes most of the functionality of the REST API. It can perform management functions, such as managing registration tokens and clients. **ksctl** is designed to be run from a **remote system**, not on CipherTrust Manager itself.

To use the CLI:

1. Open the CipherTrust Manager URL in a browser.
2. Click the **API & CLI Documentation** link. The API playground is displayed.
3. At the top left, click **CLI Guide**. The CLI documentation is displayed.
4. At the top right, click the **CLI** download button. This downloads the **ksctl_images.zip** file.
5. Unzip the **ksctl_images.zip** file.
6. Set up the **ksctl-os** file for your system.
7. Run **ksctl ddc** to run Thales CipherTrust Data Discovery and Classification specific commands.

Refer to the CipherTrust Manager documentation for details. For details on commands related to Thales CipherTrust Data Discovery and Classification, refer to the online documentation of **ksctl ddc**.

Information Types

Infotype Name	Category	Region
American Express	Financial	Global
Australian Bank Account Number	Financial	Oceania
Australian Business Number	Financial	Oceania
Australian Company Number	Financial	Oceania
Australian Driver License Number	Personal Data	Oceania
Australian Healthcare Identifier - Organisation	Medical	Oceania
Australian Individual Healthcare Identifier	Medical	Oceania
Australian Mailing Address	Personal Data	Oceania
Australian Medicare Card	Medical	Oceania
Australian Medicare Provider	Medical	Oceania
Australian Passport Number	Personal Data	Oceania
Australian Tax File Number	National ID	Oceania
Australian Telephone Number	Personal Data	Oceania
Austrian Driver License Number	Personal Data	Europe
Austrian Mailing Address	Personal Data	Europe
Austrian Passport Number	Personal Data	Europe
Austrian Personalausweis	National ID	Europe
Austrian SSN	National ID	Europe
Austrian Telephone Number	Personal Data	Europe
Belgian Driver License Number	Personal Data	Europe
Belgian eID	National ID	Europe
Belgian National Number	National ID	Europe

Infotype Name	Category	Region
Belgian Passport Number	Personal Data	Europe
Belgian Telephone Number	Personal Data	Europe
Brazilian CPF	National ID	Americas
Brazilian Registro Geral	National ID	Americas
Bulgarian EGN	National ID	Europe
Canadian Bank Account Number	Financial	Americas
Canadian Health Service Number	Medical	Americas
Canadian Mailing Address	Personal Data	Americas
Canadian Passport Number	Personal Data	Americas
Canadian Personal Health Identification Number (PHIN)	Medical	Americas
Canadian Social Insurance Number	National ID	Americas
Canadian Telephone Number	Personal Data	Americas
Chilean RUN	National ID	Americas
China Union Pay	Financial	Global
Croatian OIB	National ID	Europe
Cypriot Passport Number	Personal Data	Europe
Czech Republic RC	National ID	Europe
Danish CPR	National ID	Europe
Danish Driver License Number	Personal Data	Europe
Danish Passport Number	Personal Data	Europe
Date Of Birth	Personal Data	Global
Date Of Birth (under 18)	Personal Data	Global
Diners Club	Financial	Global

Infotype Name	Category	Region
Discover	Financial	Global
Drug Enforcement Agency Number	Medical	Americas
Dutch Burgerservicenummer	National ID	Europe
Dutch Driver License Number	Personal Data	Europe
Dutch NIK	National ID	Europe
Dutch Passport Number	Personal Data	Europe
Dutch Telephone Number	Personal Data	Europe
Email addresses	Personal Data	Global
Ethnicity (English)	Personal Data	Global
European EHIC	Medical	Europe
Finnish HETU	National ID	Europe
French Carte Vitale	National ID	Europe
French CNI	National ID	Europe
French Driver License Number	Personal Data	Europe
French INSEE	National ID	Europe
French Mailing Address	Personal Data	Europe
French Passport Number	Personal Data	Europe
French Telephone Number	Personal Data	Europe
Gambian National Identification Number	National ID	Africa
Gender (English)	Personal Data	Global
Generic Bank Account Number	Financial	Global
German Driver License Number	Personal Data	Europe
German Mailing Address	Personal Data	Europe

Infotype Name	Category	Region
German Passport Number	Personal Data	Europe
German Personalausweis	National ID	Europe
German Telephone Number	Personal Data	Europe
Greek AFM	National ID	Europe
Greek AMKA	National ID	Europe
Hong Kong ID	National ID	Asia
Hungarian Personal ID	National ID	Europe
Icelandish Kennitala	National ID	Europe
International Bank Account Number (IBAN)	Financial	Global
IP Address	Personal Data	Global
Iranian National Identification Number	National ID	Asia
Irish Driver License Number	Personal Data	Europe
Irish Passport Card Number	Personal Data	Europe
Irish Passport Number	Personal Data	Europe
Irish Personal Public Service Number	National ID	Europe
Irish Telephone Number	Personal Data	Europe
ISO8583 message with PAN	Financial	Global
Israeli Bank Account Number	Financial	Asia
Israeli Identity Number	National ID	Asia
Italian CARTA D'IDENTITÀ	National ID	Europe
Italian Codice Fiscale	National ID	Europe
Italian Driver License Number	Personal Data	Europe
Italian Mailing Address	Personal Data	Europe

Infotype Name	Category	Region
Italian Passport	Personal Data	Europe
Italian Telephone Number	Personal Data	Europe
Japanese Bank Account Number	Financial	Asia
Japanese Driver License Number	Personal Data	Asia
Japanese Passport Number	Personal Data	Asia
Japanese Resident Registration Number	National ID	Asia
Japanese Social Insurance Number (SIN)	National ID	Asia
JCB	Financial	Global
Laser	Financial	Global
Latvian Personas Kods	National ID	Europe
License Number	Personal Data	Global
Login credentials	Personal Data	Global
Luxembourg Driver License Number	Personal Data	Europe
Luxembourg ID	National ID	Europe
Luxembourg Passport Number	Personal Data	Europe
Luxembourg Phone Number	Personal Data	Europe
MAC Address	Personal Data	Global
Macedonian UMCN	National ID	Europe
Maestro	Financial	Global
Malaysian NRIC	National ID	Asia
Maltese eID	National ID	Europe
Mastercard	Financial	Global
Mexican CURP	National ID	Americas

Infotype Name	Category	Region
New Zealand Inland Revenue Number	National ID	Oceania
New Zealand Mailing Address	Personal Data	Oceania
New Zealand Telephone Number	Personal Data	Oceania
Norwegian Birth Number	National ID	Europe
Norwegian Driver License Number	Personal Data	Europe
Norwegian Passport Number	Personal Data	Europe
Passport Number	Personal Data	Global
Peoples Republic of China ID	National ID	Asia
Personal Names (Austrian)	Personal Data	Europe
Personal Names (Belgian)	Personal Data	Europe
Personal Names (English)	Personal Data	Global
Personal Names (French)	Personal Data	Europe
Personal Names (German)	Personal Data	Europe
Personal Names (Italian)	Personal Data	Europe
Personal Names (Netherlands)	Personal Data	Europe
Personal Names (Polish)	Personal Data	Europe
Personal Names (Portuguese)	Personal Data	Europe
Polish Driver License Number	Personal Data	Europe
Polish Identity Card	National ID	Europe
Polish Mailing Address	Personal Data	Europe
Polish Passport Number	Personal Data	Europe
Polish PESEL	National ID	Europe
Polish Telephone Number	Personal Data	Europe

Infotype Name	Category	Region
Portuguese Citizen's Card	National ID	Europe
Portuguese Driver License Number	Personal Data	Europe
Portuguese Fiscal Number	National ID	Europe
Portuguese Identity Number	National ID	Europe
Portuguese Mailing Address	Personal Data	Europe
Portuguese Passport Number	Personal Data	Europe
Portuguese Phone Number	Personal Data	Europe
Private Label Card	Financial	Global
Religion (English)	Personal Data	Global
Romanian Identity Card	National ID	Europe
Romanian Numerical Personal Code	National ID	Europe
Saudi Arabia National ID	National ID	Asia
Serbian UMCN	National ID	Europe
Singaporean NRIC	National ID	Asia
Slovakian RC	National ID	Europe
Slovenian EMSO	National ID	Europe
South African Identity Number	National ID	Africa
South Korean Corporation Registration Number (법인등록번호)	Financial	Asia
South Korean Driver License Number	Personal Data	Asia
South Korean Foreigner Number	National ID	Asia
South Korean Gwangju Bank (광주은행) Account Number	Financial	Asia
South Korean Jeju Bank (제주은행) Account Number	Financial	Asia
South Korean Jeonbuk Bank (전북은행) Account Number	Financial	Asia

Infotype Name	Category	Region
South Korean KB Bank (국민은행) Account Number	Financial	Asia
South Korean KEB Hana Bank (KEB하나은행) Account Number	Financial	Asia
South Korean NH Bank (농협은행) Account Number	Financial	Asia
South Korean Passport	Personal Data	Asia
South Korean Phone Number	Personal Data	Asia
South Korean RRN	National ID	Asia
South Korean Shinhan Bank (신한은행) Account Number	Financial	Asia
South Korean Taxpayer Identification Number (사업자등록번호)	Financial	Asia
Spanish DNI	National ID	Europe
Spanish Driver License Number	Personal Data	Europe
Spanish NIE	National ID	Europe
Spanish Passport Number	Personal Data	Europe
Spanish Social Security Number	National ID	Europe
Spanish Telephone Number	Personal Data	Europe
Sri Lankan National Identity Card	National ID	Asia
Swedish Driver License Number	Personal Data	Europe
Swedish Nationellt ID-kort	National ID	Europe
Swedish Passport Number	Personal Data	Europe
Swedish Personnummer	National ID	Europe
SWIFT Code	Financial	Global
Swiss Social Security Number	National ID	Europe
Taiwanese ID	National ID	Asia
Thai Population Identification Code	National ID	Asia

Infotype Name	Category	Region
Troy	Financial	Global
Turkish Identification Number	National ID	Europe
Turkish Telephone Number	Personal Data	Europe
United Arab Emirates ID	National ID	Asia
United Kingdom Community Health Index	Medical	Europe
United Kingdom Driver License Number	Personal Data	Europe
United Kingdom Electoral Roll Number	Personal Data	Europe
United Kingdom Health and Care Number	Medical	Europe
United Kingdom Mailing Address	Personal Data	Europe
United Kingdom National Health Service Number	Medical	Europe
United Kingdom NI Number	National ID	Europe
United Kingdom Passport Number	Personal Data	Europe
United Kingdom Self Assessment UTR Number	National ID	Europe
United Kingdom Telephone Number	Personal Data	Europe
United Kingdom VAT Number	Financial	Europe
United States Bank Account Number	Financial	Americas
United States Driver License Number	Personal Data	Americas
United States Health Insurance Claim Number	Medical	Americas
United States Health Plan Identifier	Medical	Americas
United States Individual Taxpayer Identification Number (ITIN)	National ID	Americas
United States Mailing Address	Personal Data	Americas
United States National Provider Identifier	Medical	Americas
United States Routing Transit Number	Financial	Americas

Infotype Name	Category	Region
United States Social Security Number	National ID	Americas
United States Telephone Number	Personal Data	Americas
Visa	Financial	Global
Yugoslavia UMCN	National ID	Europe

Supported Formats

Files

Type	Format
Compressed	bzip2, Gzip (all types), TAR, Zip (all types)
Databases	Access, DBase, SQLite, MSSQL MDF & LDF
Images	BMP, FAX, GIF, JPG, PDF (embedded), PNG, TIF
Microsoft Backup Archive	Microsoft Binary / BKF
Microsoft Office	v5, 6, 95, 97, 2000, XP, 2003 onwards
Open Source	Star Office / Open Office / Libre Office
Open Standards	PDF, RTF, HTML, XML, CSV, TXT

Office files

WORD

- > **Legacy:** Legacy filename extensions denote binary Microsoft Word formatting that became outdated with the release of Microsoft Office 2007. Although the latest version of Microsoft Word can still open them, they are no longer developed. Legacy filename extensions include:
 - .doc – Legacy Word document; Microsoft Office refers to them as "Microsoft Word 97 – 2003 Document"
 - .dot – Legacy Word templates; officially designated "Microsoft Word 97 – 2003 Template"
 - .wbk – Legacy Word document backup; referred as "Microsoft Word Backup Document"
- > **OOXML:** Office Open XML (OOXML) format was introduced with Microsoft Office 2007 and became the default format of Microsoft Word ever since. Pertaining file extensions include:
 - .docx – Word document
 - .docm – Word macro-enabled document; same as docx, but may contain macros and scripts
 - .dotx – Word template
 - .dotm – Word macro-enabled template; same as dotx, but may contain macros and scripts
 - .docb – Word binary document introduced in Microsoft Office 2007

EXCEL

- > **Legacy:** Legacy filename extensions denote binary Microsoft Excel formats that became outdated with the release of Microsoft Office 2007. Although the latest version of Microsoft Excel can still open them, they are no longer developed. Legacy filename extensions include:

-
- .xls – Legacy Excel worksheets; officially designated "Microsoft Excel 97-2003 Worksheet"
 - .xlt – Legacy Excel templates; officially designated "Microsoft Excel 97-2003 Template"
 - .xlm – Legacy Excel macro
- > **OOXML:** Office Open XML (OOXML) format was introduced with Microsoft Office 2007 and became the default format of Microsoft Excel ever since. Excel-related file extensions of this format include:
- .xlsx – Excel workbook
 - .xlsm – Excel macro-enabled workbook; same as xlsx but may contain macros and scripts
 - .xltx – Excel template
 - .xltn – Excel macro-enabled template; same as xltx but may contain macros and scripts

POWERPOINT

> Legacy:

- .ppt – Legacy PowerPoint presentation
- .pot – Legacy PowerPoint template
- .pps – Legacy PowerPoint slideshow

> OOXML:

- .pptx – PowerPoint presentation
- .pptm – PowerPoint macro-enabled presentation
- .potx – PowerPoint template
- .potm – PowerPoint macro-enabled template
- .ppam – PowerPoint add-in
- .ppsx – PowerPoint slideshow
- .ppsm – PowerPoint macro-enabled slideshow
- .sldx – PowerPoint slide
- .sldm – PowerPoint macro-enabled slide

ACCESS

> Legacy:

- .ade – Protected Access Data Project (not supported in 2013)
- .adp - Access Data Project (not supported in 2013)
- .mdb - Access Database (2003 and earlier)
- .cdb - Access Database (Pocket Access for Windows CE)
- .mda - Access Database, used for addins (Access 2, 95, 97), previously used for workgroups (Access 2)
- .mdt - Access Add-in Data (2003 and earlier)
- .mdf - Access (SQL Server) detached database (2000)
- .mde - Protected Access Database, with compiled VBA and macros (2003 and earlier)

-
- .ldb - Access lock files (associated with .mdb)

> **Available formats since Access 2007:**

- .accdb – The file extension for the new Office Access 2007 file format. This takes the place of the MDB file extension
- .accde – The file extension for Office Access 2007 files that are in "execute only" mode. ACCDE files have all Visual Basic for Applications (VBA) source code hidden. A user of an ACCDE file can only execute VBA code, but not view or modify it. ACCDE takes the place of the MDE file extension
- .accdt – The file extension for Access Database Templates
- .accdr – is a new file extension that enables you to open a database in runtime mode. By simply changing a database's file extension from .accdb to .accdr, you can create a "locked-down" version of your Office Access database. You can change the file extension back to .accdb to restore full functionality

OTHER

- .pub – a Microsoft Publisher publication
- .xps – a XML-based document format used for printing (on Windows Vista and later) and preserving documents

Databases

- > Microsoft SQL
- > Oracle
- > DB2

Big Data

- > Hadoop

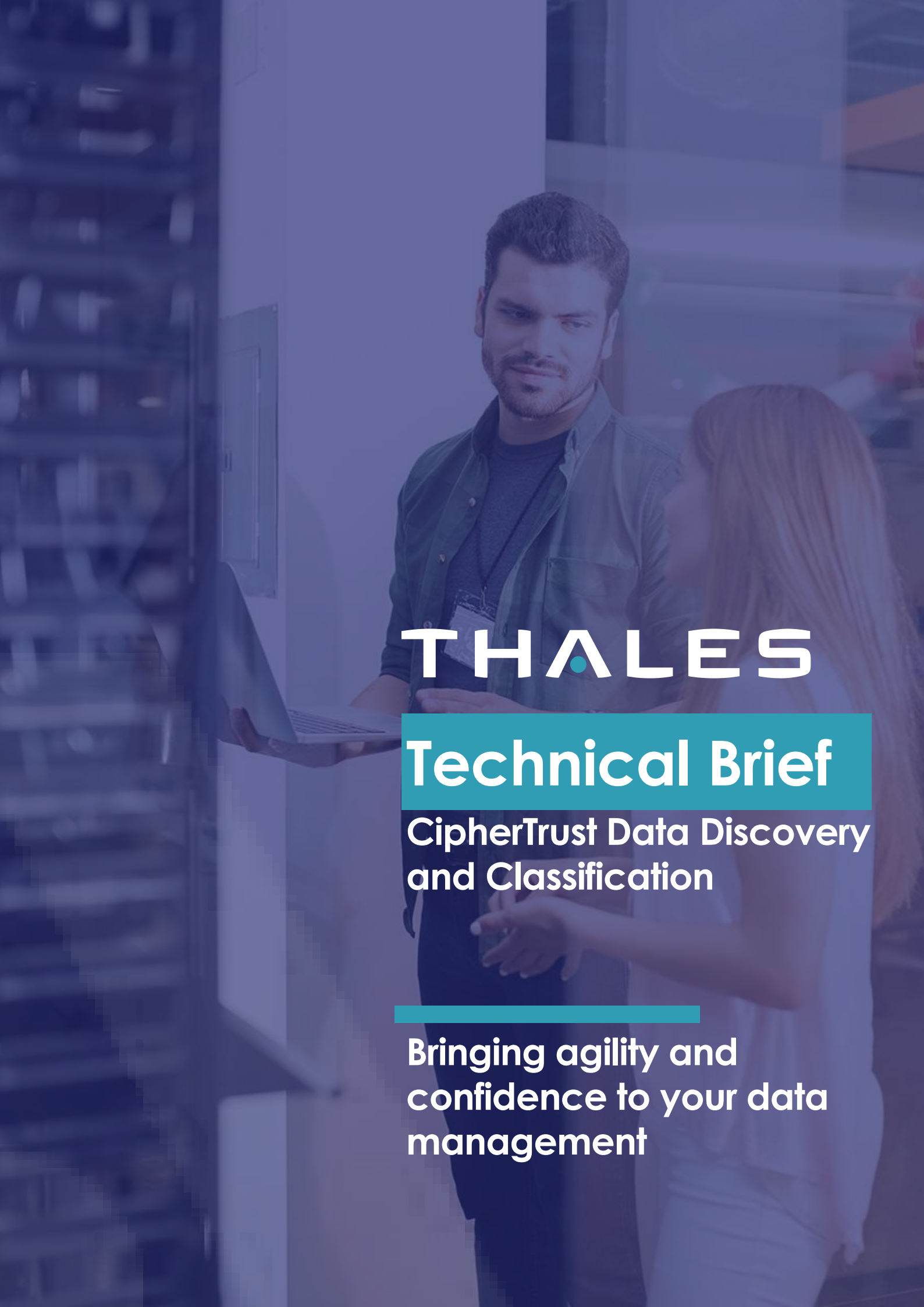
Security Audit Log Event Messages

The following table contains a complete list of security audit log event messages that DDC prints in the log file.

Message	Explanation
DDCScanClientInvalidCredentialsProbe	A probe with invalid credentials.
DDCScanClientUnexpectedErrorProbe	An unknown probe error.
DDCPhoenixBackgroundProcessAuthenticationError	A failed authentication against PQS in background processes.
DDCPhoenixUpdatePQSSettingsAuthenticationError	A failed authentication against PQS updating PQS settings.
DDCHDFSUpdateHDFSSettingsAuthenticationError	A failed authentication against HDFS updating HDFS settings.
DDCHDFSBackgroundProcessAuthenticationError	A failed authentication against HDFS in background processes.
DDCUnauthorizedCloneRequest	An unauthorized CLONE request.
DDCUnauthorizedGetRequest	An unauthorized GET request.
DDCUnauthorizedListRequest	An unauthorized LIST request.
DDCUnauthorizedListPaginatedRequestWithContext	An unauthorized LIST PAGINATED request with context.
DDCUnauthorizedCreateRequest	An unauthorized CREATE request.
DDCUnauthorizedUpdateRequest	An unauthorized UPDATE request.
DDCUnauthorizedListProvisionedRequest	An unauthorized LIST PROVISIONED request.
DDCUnauthorizedGetProvisionedRequest	An unauthorized GET PROVISIONED request.
DDCUnauthorizedGetActiveNodeRequest	An unauthorized GET ACTIVE NODE request.
DDCUnauthorizedTestConnectivityRequest	An unauthorized TEST CONNECTIVITY request.
DDCUnauthorizedGetLicenseRequest	An unauthorized GET LICENSE request.
DDCUnauthorizedDecryptRawDataFileRequest	An unauthorized DECRYPT RAW DATA FILE request.

Message	Explanation
DDCUnauthorizedGetDatastoreReportRequest	An unauthorized GET DATASTORE REPORT request.
DDCUnauthorizedFindScanRequest	An unauthorized FIND SCAN request.
DDCUnauthorizedScanActionRequest	An unauthorized SCAN ACTION request.
DDCPQSUnaccessibleGetSummaryReportError	An inaccessible PQS in GET SUMMARY REPORT request.
DDCPQSUnaccessibleGetDatastoreDetailReportError	An inaccessible PQS in GET DATASTORE DETAIL REPORT request.
DDCPQSUnaccessibleGetDataObjectsDetailsReportError	An inaccessible PQS in GET DATAOBJECTS DETAIL REPORT request.
DDCPQSUnaccessibleGetInfotypesSummaryReportError	An inaccessible PQS in GET INFOTYPES SUMMARY REPORT request.
DDCPQSUnaccessibleGetDataObjectsSummaryReportError	An inaccessible PQS in GET DATAOBJECTS SUMMARY REPORT request.
DDCPQSUnaccessibleGetScanDetailsReportError	An inaccessible PQS in GET SCAN DETAILS REPORT request.
DDCPQSUnaccessibleCreateReportTemplateError	An inaccessible PQS in CREATE REPORT TEMPLATE request.
DDCPQSUnaccessibleGetReportTemplateError	An inaccessible PQS in GET REPORT TEMPLATE request.
DDCPQSUnaccessibleFindReportTemplatesError	An inaccessible PQS in FIND REPORT TEMPLATE request.
DDCPQSUnaccessibleUpdateReportTemplateError	An inaccessible PQS in UPDATE REPORT TEMPLATE request.
DDCPQSUnaccessibleGetScanExecutionsError	An inaccessible PQS in GET SCAN EXECUTIONS request.
DDCResourceRetrievalGenericCloneError	A GENERIC CLONE request.
DDCResourceRetrievalGenericGetError	A GENERIC GET request.
DDCResourceRetrievalGenericListError	A GENERIC GET request.

Message	Explanation
DDCResourceRetrievalGenericListPaginatedRequestError	A GENERIC LIST PAGINATED request.
DDCResourceRetrievalGenericCreateError	A GENERIC CREATE request.
DDCResourceRetrievalGenericUpdateError	A GENERIC UPDATE request.
DDCResourceRetrievalGenericListProvisionError	A GENERIC LIST PROVISION request.
DDCDatastoreDecryptDataEncryptionKeyNotFoundError	A GET KEY request.
DDCDatastoreEncryptDataError	An ENCRYPT DATA request.
DDCSaveWatcherInterruptedTimeout	An INTERRUPTED TIMEOUT request.
DDCSaveClientRetrieveScanTimeout	A RETRIEVE SCAN TIMEOUT request.
DDCSaveActionRequest	A SCAN ACTION request.
DDCDatastoreUpdateRequest	A DATASTORE UPDATE request.
DDCDatastoreCreateRequest	A DATASTORE CREATE request.
DDCSaveDeleteRequest	A SCAN DELETE request.
DDCSummaryReportGetRequest	A GET SUMMARY REPORT request.
DDCDatastoreDetailReportGetRequest	A GET DATASTORE DETAILS REPORT request.
DDCDataObjectsDetailReportGetRequest	A GET DATASTORE DETAILS REPORT request.
DDCInfotypesSummaryReportGetRequest	A GET INFOTYPES SUMMARY REPORT request.
DDCDataObjectsSummaryReportGetRequest	A GET DATAOBJECTS SUMMARY REPORT request.
DDCSaveDetailsReportGetRequest	A GET SCAN DETAILS REPORT request.

A man and a woman are in a server room. The man, wearing a green jacket and a lanyard, is holding a laptop and looking at the woman. The woman, with long blonde hair, is looking at the laptop. The background shows server racks and a blue-tinted overlay.

THALES

Technical Brief

**CipherTrust Data Discovery
and Classification**

**Bringing agility and
confidence to your data
management**

Executive summary



The rapid, often exponential, growth of data year-on-year in organizations like yours makes effective data management an extremely challenging proposition. An increasing switch to remote working raises the bar even higher with many data storage areas or volumes coming into play which are outside your IT team's direct control.

As part of the broader group within your organization responsible for data privacy and security compliance, ensuring no vulnerable areas are overlooked when implementing your data protection strategy is of paramount importance. After all, a data breach will inevitably cause severe business disruption in addition to large fines incurred for non-compliance with the seemingly endless stream of new or enhanced data privacy laws and regulations. Failing to prepare properly is definitely not a viable option for you.

A significant weakness often seen in typical data management implementations is the lack of visibility into the precise types of data being held across various local servers, network drives and increasingly cloud storage locations. Many organizations have been breached already, some have encountered near misses and others are migrating large workloads to the cloud without fully understanding the fundamental nature of the data itself and the exposure risks involved – situations for you to avoid.

Historically staff knowledge was sufficient and simple 'off-the-shelf' encryption methods from database vendors would satisfy needs – no longer with the vast data footprint that is widely dispersed and growing by the second. You inevitably need assistance to take control and keep your organization safe.

Find sensitive data

Thales CipherTrust Data Discovery and Classification

Analytical outcomes

Local storage



Network storage



Database



Big Data



Cloud



Data protection



Risk analysis



Reporting & compliance



Thales CipherTrust Data Discovery and Classification is...

A tool that is significantly more effective and efficient than manual methods for data discovery. It helps classify all your data, while supporting every mainstream operating system, data storage type, structured and unstructured data you are likely to possess. Existing processes and tools are supplemented, rather than replaced, providing high levels of automation in your search for sensitive data, wherever it resides. Ultimately, it can help you become more agile and support better data management decisions:



It finds sensitive data that you may not realize even exists in your organization, helping you eliminate threats to your business continuity and reduce your data footprint



It enables you quickly to assess your risks of non-compliance and determine the appropriate protection actions to take



It provides confidence that you are protecting the right assets, including proprietary data or intellectual property, across all your data storage locations including the cloud



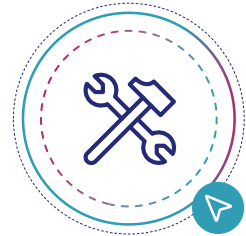
Contents



**Executive
summary**



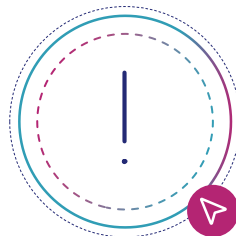
**The role of discovery
and classification**



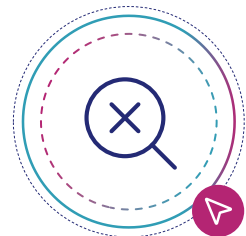
**How it all
fits together**



**Categorizing
sensitive data**



**Using agents for
discovery**



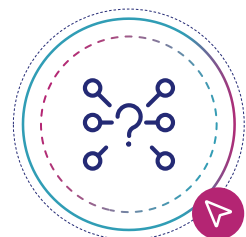
**Analyzing
scan results**



**Getting answers
to key questions**



**Discovery and
protection in
a single step**



**Reasons for
our choice
of architecture**



**Integrating with
other solutions**



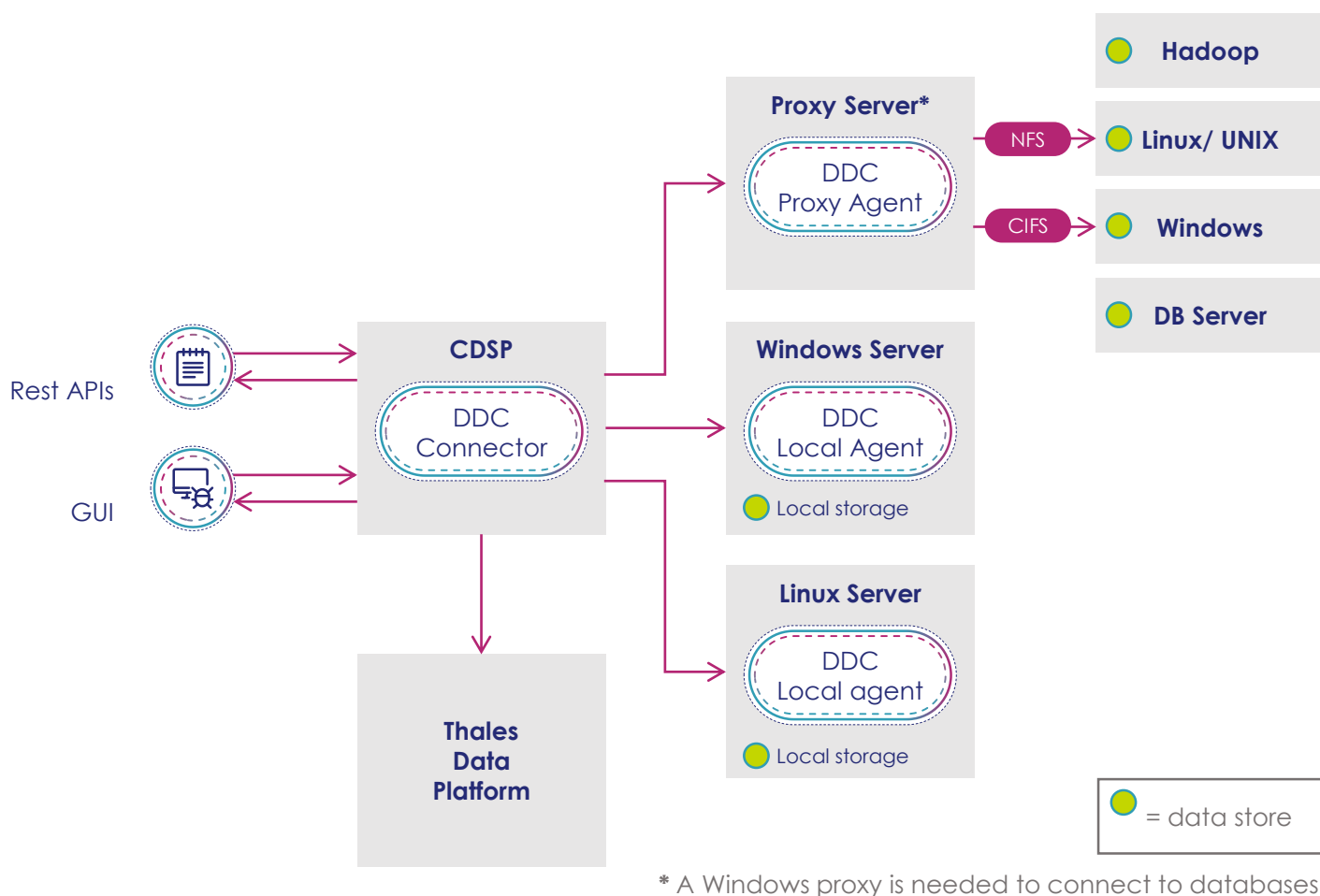
**Key
takeaways**



**Abbreviations and
glossary**

The role of CipherTrust Data Discovery and Classification (DDC)

DDC offers a more streamlined, accurate and unattended (or automated) approach in comparison to sensitive data searches using manual methods. If you are using a data discovery tool from another vendor you should still take a closer look at DDC - it likely covers additional data storage types, extending the range of structured and unstructured data that can be analyzed while offering higher levels of accuracy by finding sensitive data other tools often miss.



Sensitive items found by DDC are not just restricted to data privacy laws and regulations. Fine tuning using custom infotypes (which you define) facilitates visibility on where data sensitive to your organization is present. By enabling critical assets (such as internal financial data, trade secrets, intellectual property and confidential business plans for example) to be located, their current levels of protection and associated risks of unintended exposure can be better understood. Considerable flexibility comes from running multiple scans simultaneously, each looking for different types of information (if desired) to create a faster and more efficient way to analyze your complete data footprint across a diverse range of storage locations. It is the ability to create multiple classification profiles (customized where needed) that underpins such performance and efficiency benefits that never could be achieved in a comparable timeframe using manual methods or less capable tools.



Discovering the data is only part of the task in hand – **classification is equally important**, providing deeper understanding of the types of data and their footprint in your organization. DDC does not move or modify your data, merely summarizing what it uncovers during scans in a non-intrusive manner.

You can **see exactly how your data is split** between the personal, financial, medical and national ID categories – you may be surprised that some of this even exists. At least now you will have insight into precisely where it resides to manage the business risk appropriately.

The insights available after scans have been performed provide **important guidance** on what you may need to do next, especially related to protection. You can choose how many scans you wish to incorporate in any given report. Each report improves data visibility, providing detailed insight into:

The names of the infotypes found together with the number of sensitive data matches

The risk associated with each of the data objects analyzed during the scan in question

The protection status so that you can see what it protected and what is not

DDC provides options to run scans at any time, over and over again if desired. The tool is intended to be used regularly to keep your insight up to date rather than just being a one-off activity. Since data is always in a state of flux, DDC is ready to convey the latest true representation of what you have and what you might need to do to keep your business safe.

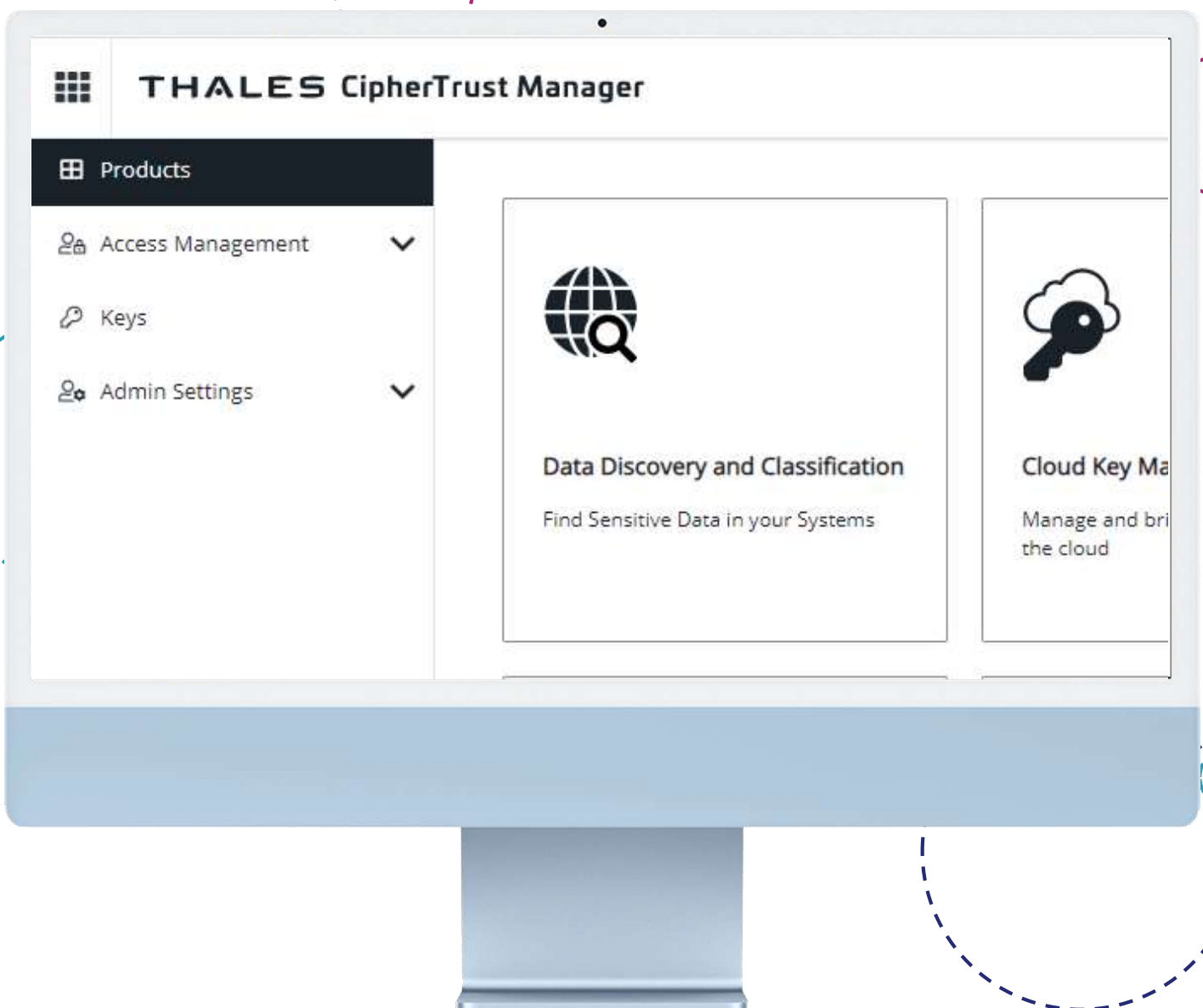




How it all fits together

DDC is configured using a graphical user interface (launched from within the CipherTrust Manager console) or via the REST API. It offers a simple and intuitive workflow, enabling you to define data locations for interrogation, specify the types of data you wish to discover and decide when and how often you wish to repeat the scanning process. Ultimately the rich views provide a deep dive into the sensitive data matches, the associated protection status and risk assessment of the data locations in question. The main features are summarized in the sections below, providing a high level insight into the power, flexibility and control available to you.

The workflow for DDC is easy to follow via the user interface. **Three main ingredients** help define your environment – **data stores**, **classification profiles** and **infotypes**.



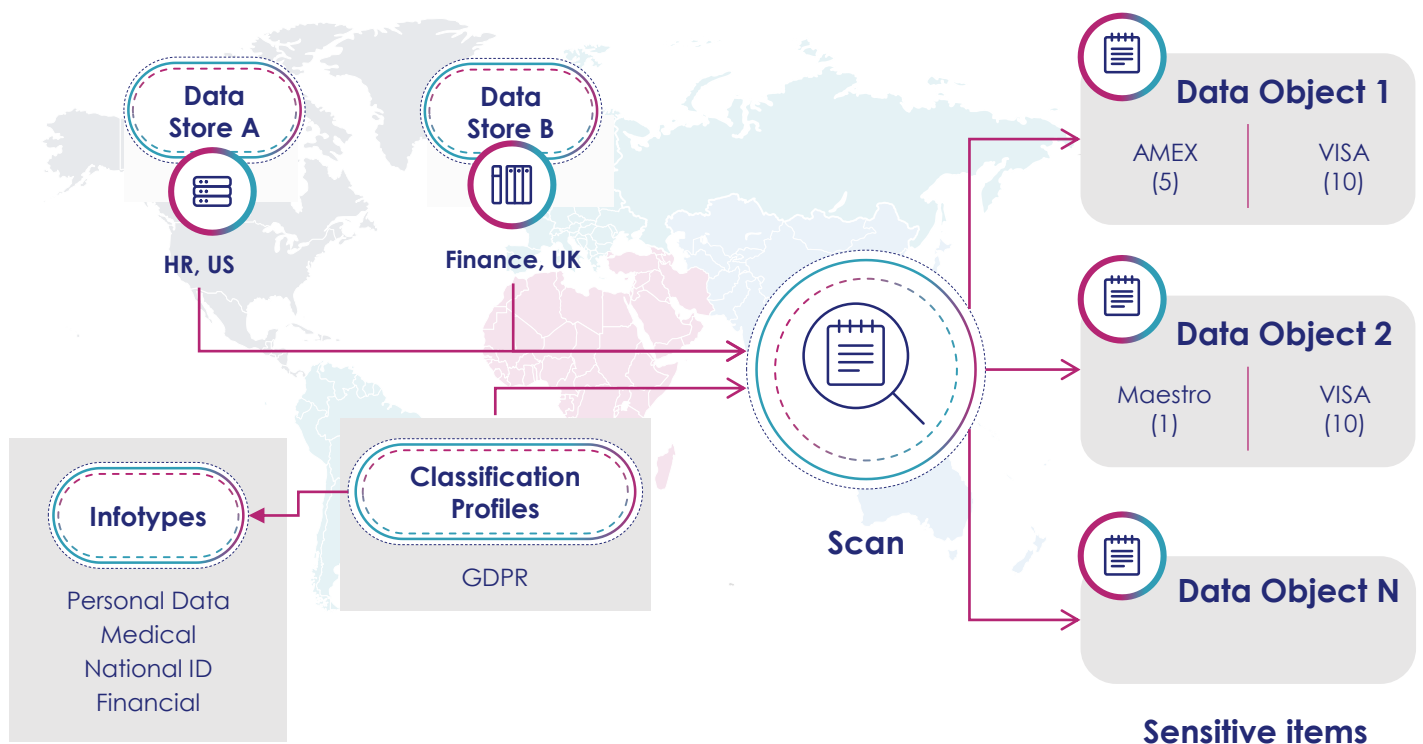


Typically, you might adopt the following sequence when using DDC for the first time:

Select and configure **data stores** of interest – we recommend concentrating on the most frequently accessed data stores first and a selected path rather than the whole store.

Choose one or more **classification profiles** – create custom classification profiles if necessary. During this process, select one or more **infotypes** of interest – use the custom infotype capability if the prebuilt list is insufficient.

Configure and start the **scan** – consider just selecting one data store for your first scan to test your configuration.





Data stores

Local storage	Local storage	
Network Storage	Windows share (CIS/SMB)	Unix file system (NFS)
Database	IBM DB2 Microsoft SQL MongoDB MySQL	Oracle DB PostgreSQL SAP HANA
Big Data	Hadoop clusters	Teradata
Cloud	AWS S3 Buckets Azure Blob Azure Table	Google (Gmail & Gdrive) Office 365 (Exchange, OneDrive & SharePoint) Salesforce
Server	Sharepoint Server	

Types of file supported

Databases	Access	SQLite	Microsoft Office	v5	XP
	Dbase	MSSQL MDF & LDF		6	2003 onwards
				95	Office Files: Word, Excel, PowerPoint.
				97	Access, Outlook, Other
				2000	(.pub & .xps)
Images	BMP FAX GIF JPG	PDF (embedded) PNG TIF	Open Source	Star Office / Open Office / Libre Office	
Compressed	Bzip2 Gzip (all types)	TAR ZIP (all types)	Open Standards	PDF RTF HTML	XML CSV TXT
Microsoft Backup Archive	Microsoft Binary / BKF				



Classification profile templates (prebuilt)

APA	Australia Privacy Amendment	HIPAA	Health Insurance Portability and Accountability Act
APPI	Act on Protection of Personal Information	KVKK	Turkish Personal Data Protection Law
CCPA	California Consumer Privacy Act	LGPD	General Data Protection Law (Brazil)
GDPR	Financial	NYDFS	New York State Department of Financial Services
GDPR	General Data Protection Regulation	PCI DSS	Payment Card Industry Data Security Standard
GDPR	Healthcare	PDP	Personal Data Protection Bill
GDPR	National ID	SHIELD	Privacy Shield Framework
GDPR	Personal Details	UK-GDPR	General Data Protection Regulation (UK)



The structured and unstructured data coverage provided by DDC is frequently being updated with new capabilities – check out the latest specifications [here](#).



Core classification profiles are prebuilt and aligned with major data laws and regulations, both regional and global. You can make changes to these or add totally new profiles if required. Various shortcuts are provided in terms of editing and copying so that you can quickly create and test your own profiles. There is full flexibility in specifying various parameters including sensitivity levels and infotypes.

Sensitivity levels

None

Public

Internal

Private

Restricted



Categorizing sensitive data

DDC is prebuilt with various infotypes, over 200 and counting, covering the vast majority of regional and global data privacy laws and regulations. When a prebuilt classification profile template is selected, the appropriate subset of infotypes is added automatically. You can modify or extend this list by creating your own custom infotypes, allowing rules to be defined which describe precisely how the scanning engine should look for the data strings in question. Full access to the groups and categories is available as part of the definition so that data matches can be displayed in the appropriate sections of the scan results.

Infotype categories (prebuilt)

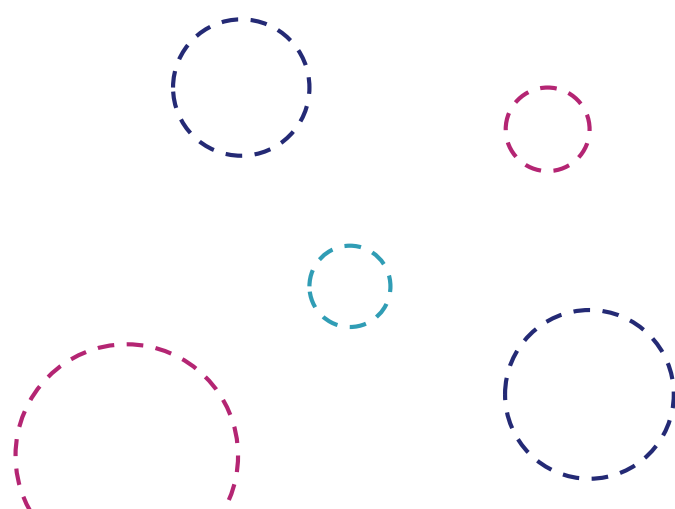
Financial	Credit/ debit cards	Bank account info
Personal data	Email addresses Login credentials Card number Ethnicity License number Roll number Passport number Date of birth MAC address	Mailing address Telephone number Gender Religion IP address Phone number Name Profanity
Medical	Patient health data	
National ID	Personal identification	



Using agents for **discovery**

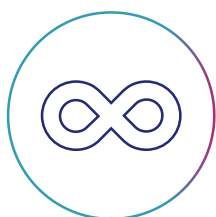
Detailed data analysis is carried out by discovery agents which perform the scanning and report the results back to the DDC Connector for analysis and processing. Two types of agents exist – local and proxy. You would normally install local agents on data store locations (for which you have the appropriate access rights) to ensure the data never leaves the server for security or performance reasons. In contrast, proxy agents are used for network or remote data stores where a separate proxy server hosts the agent.

Type of agent	Pros	Cons
Local	Faster scan Data remains local No need for credentials	Longer deployment time More complex management Only supports local data stores
Proxy	Faster to deploy and scale Ability to scan multiple data stores Supports multiple data store types No resources consumed on target host	Data sent over network to agent prior to scanning Increased network load and data footprint Ideally should be located on same virtual LAN (VLAN)





Common benefits apply to both types of agent:



An **unlimited number** of agents is available for deployment at no additional cost since licensing is based on the volume of new data scanned



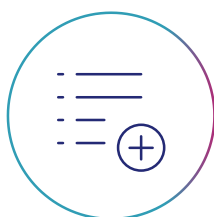
Multiple simultaneous sub-scans can be launched to reduce the time required for the overall scan



TLS encryption is available for all communications between data stores and agents to protect sensitive data from eavesdropping



Source data is untouched as agents only access a temporary data copy in internal memory during the scan



No sensitive data is stored during the process as only a summary of the scan results is sent to the DDC Connector



Scans can **still proceed** and record results even when connection to the DDC Connector is lost



With your agents installed, you can then start to think about what and how you wish to scan. We provide a great deal of flexibility in how scans are configured, managed and operated. Important parameters configurable for each scan independently include:



Name (up to 64 characters)



Description (up to 250 characters)



One or more data stores
(where to scan)



One or more classification
profiles (what to look for)



Manual or scheduled operation
(when to activate)

On completion of any given scan, it is added to a list of all scans that have been performed. Next, you would use the reporting tool to view detailed results from the scan. Depending on what is found you may make changes to the scan configuration and rerun the scan which is beneficial when working on different data stores or new types of compliance regulations for the first time.





Analyzing scan results

The results from historical and new scans are available for use with the built-in reporting tool. Three main views for scan reports are offered:

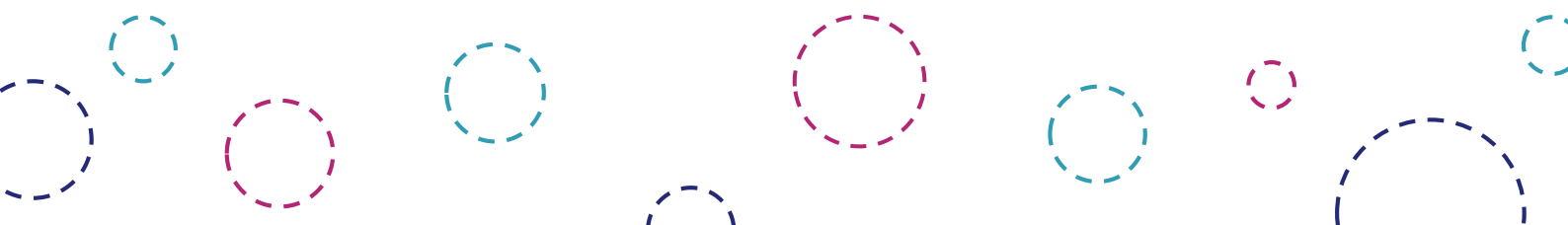
Scans

Data stores

Data objects

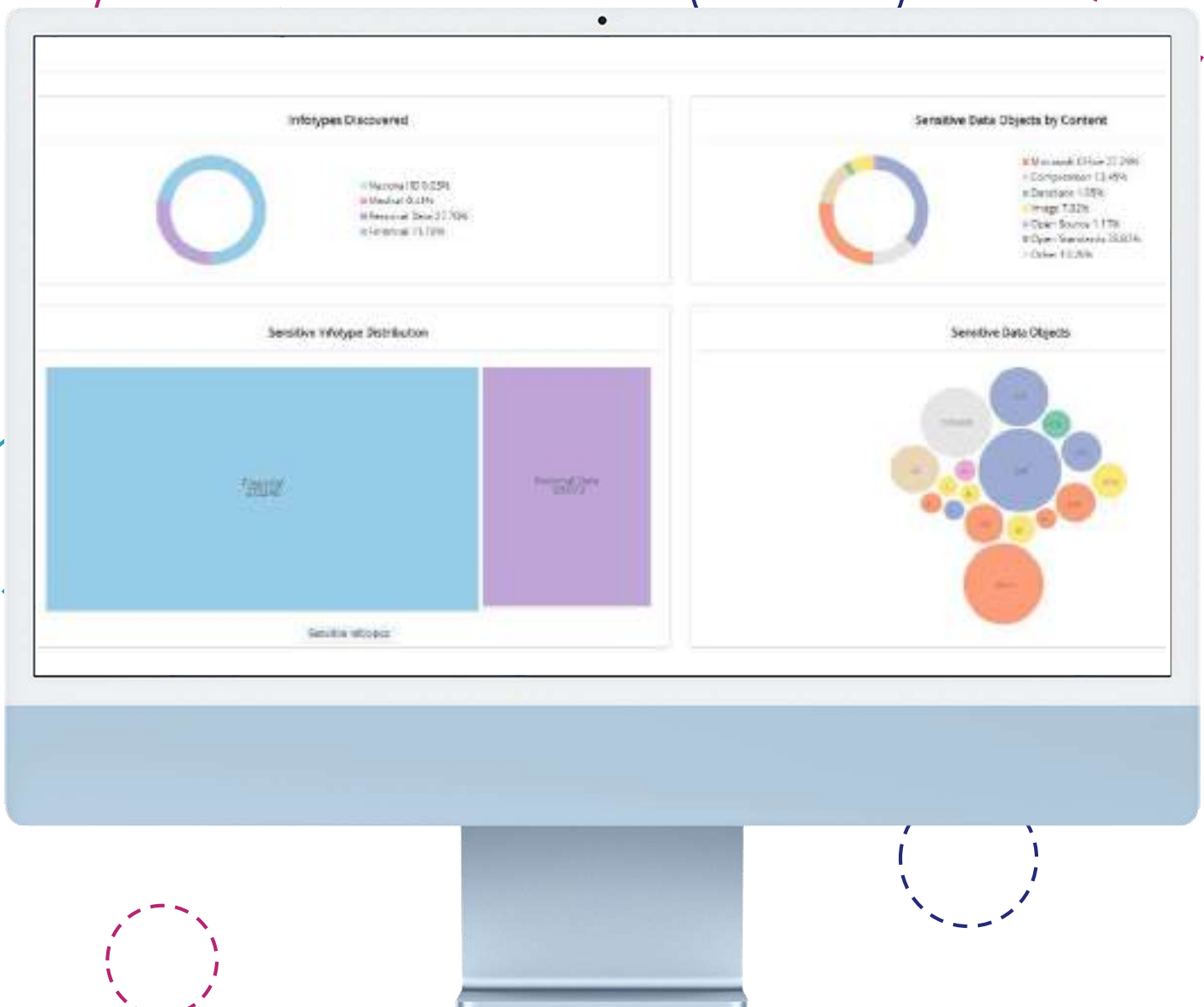
Before detailed information is available, a report is generated which aggregates the results from one or more successful scans. Often for logistical, security or performance reasons you may wish to use multiple scans (each looking for different types of regulated data) to analyze your entire data footprint. With our flexible approach you can still do this and fine-tune each scan as required using multiple iterations, before generating a final report which can represent the current status of all your data. It is also easy to understand your organizational posture from a single regulation point of view.

Each tab within the report module offers an insight into slightly different aspects of your data – all however have a common goal in helping you identify quickly any sensitive data at risk.



Scans view

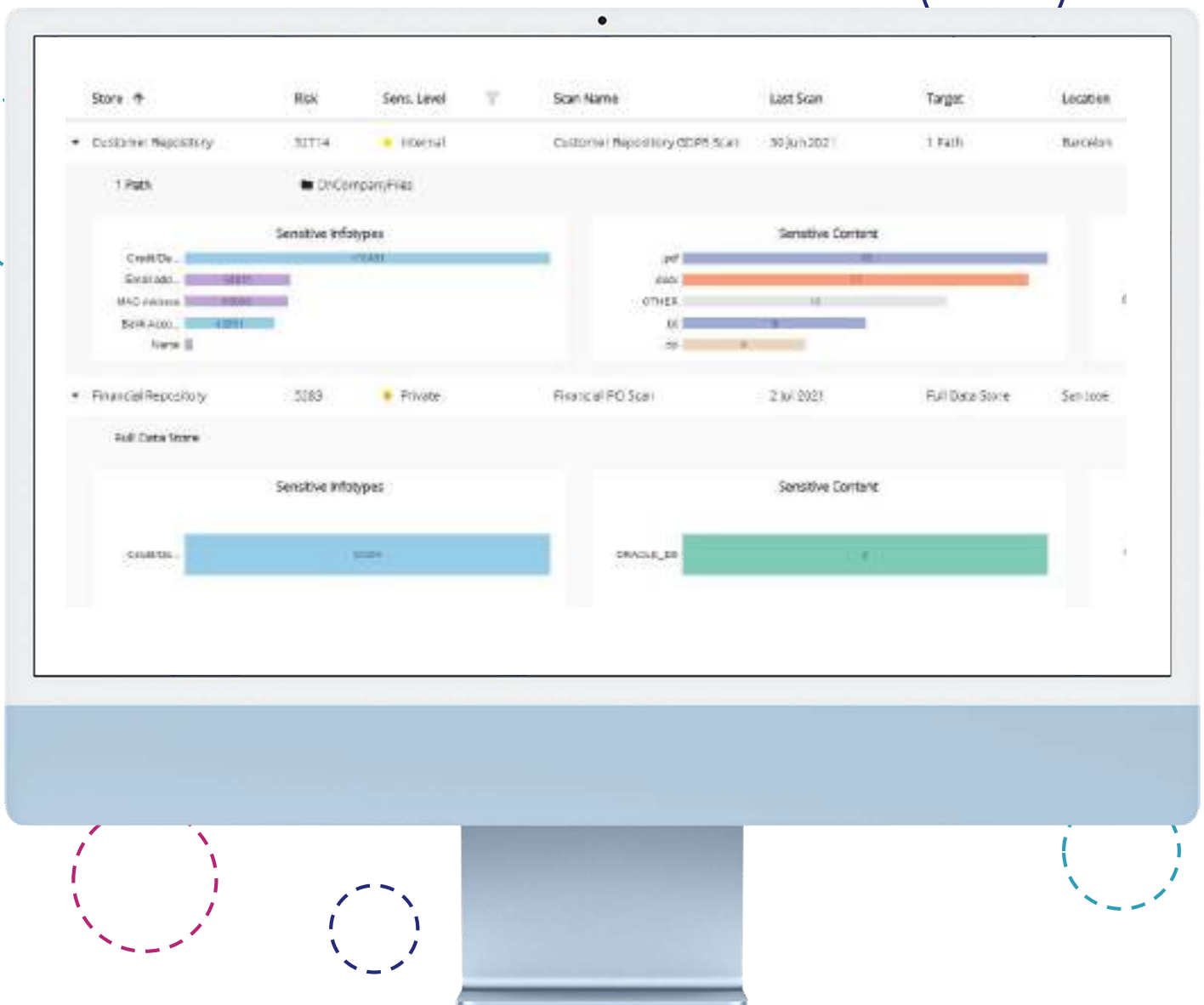
Provides a high level, aggregated scan view of the infotypes found (broken down by percentage in each category present) together with three other charts representing the presence of sensitive data objects in terms of category distribution, content types and file types in question.





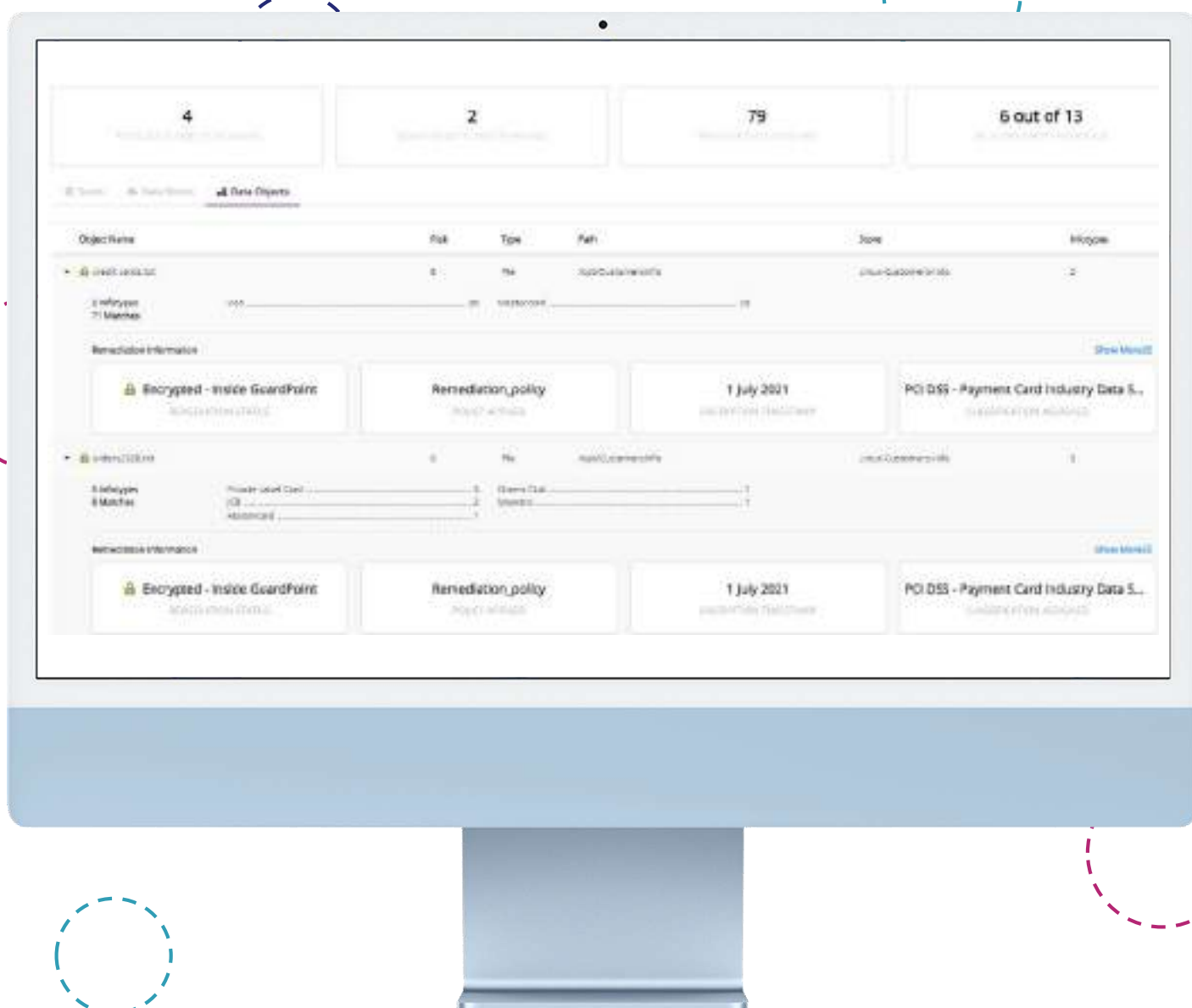
Data stores view

Provides a list of the data stores with details including the risk, sensitivity level, infotypes found and the number of sensitive data objects present for each data store – this provides a rapid means of assessing at a high level whether or not sensitive data is at risk and which data stores are involved.



Data objects view

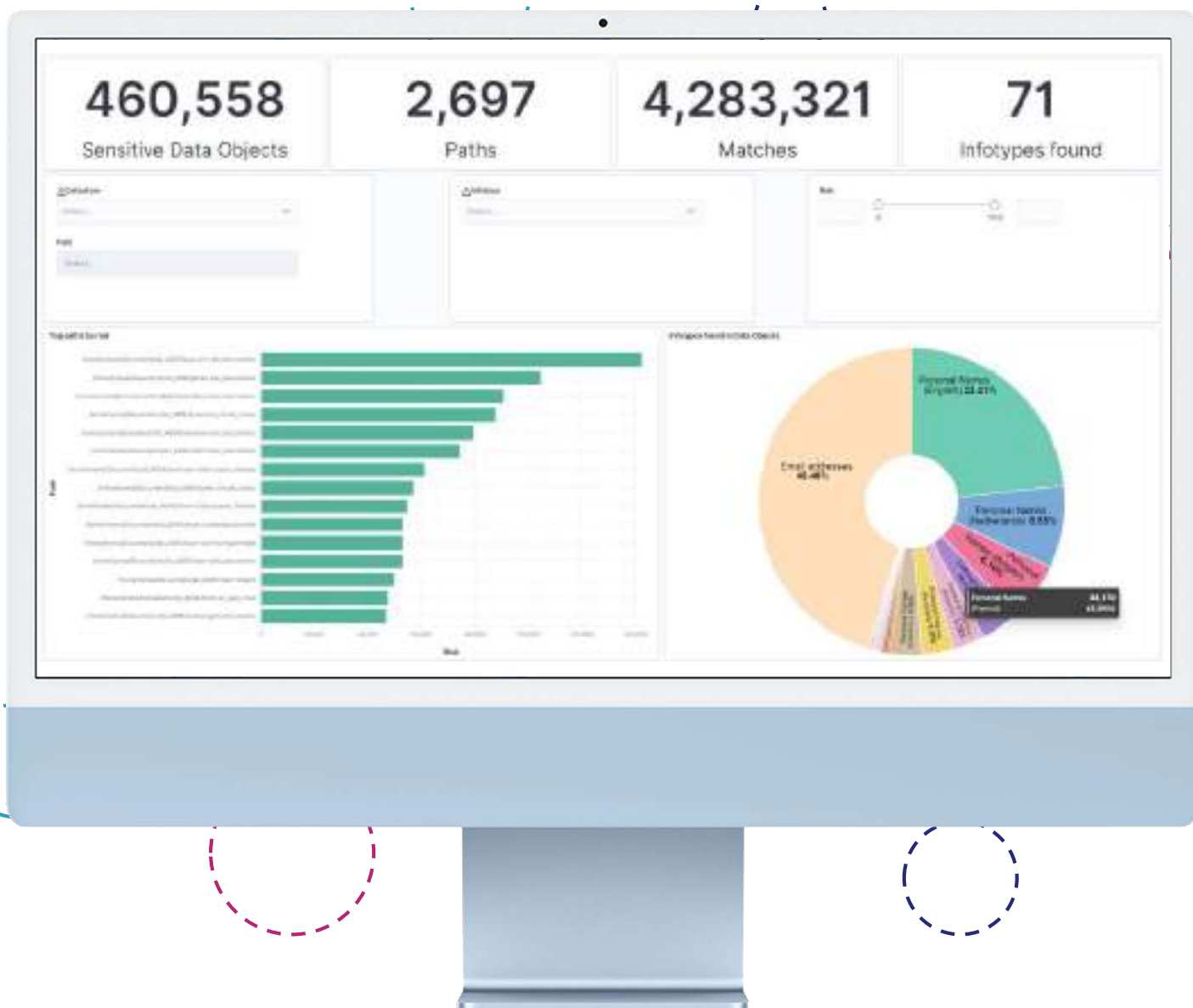
Provides a detailed list of all data objects scanned containing sensitive data, sorted in descending order of risk of exposure. For each sensitive data object in question, you can see where it is located, the data store in which it resides and the individual names and data matches for all of the associated infotypes found. Remediation status information is also present which provides a deeper insight into your sensitive data, confirming whether or not an encryption policy is active. No sensitive data is stored as part of the reporting process.





Exporting data

In addition to viewing objects using the built-in reporting tool, you can also export data in a NDJSON format for analysis by an external reporting tool such as one from [Elastic](#) used in Thales demonstrations.





Getting answers to **key** questions quickly

We make it easy to get up and running, so you can experience the wealth of insight obtainable from DDC

What types of regulated data exist in my organization?

Create a series of scans to search for different types of regulated data

Use one classification profile per scan for optimum performance

Narrow initial scanning to paths within data stores you use the most

Use the reporting tool to collate the scan results

The extent and categories of sensitive data found are presented clearly

Have I any sensitive data that is not protected?

Run one or more scans involving the data stores, classification profiles and infotypes of interest

Select the appropriate sensitivity levels

Consider using custom infotypes to help find data proprietary to you

Generate a report incorporating results from one or more previous scans

Use the subsequent report view to determine if you have any vulnerabilities

What are the priority items to address?

Use the 'Data Stores' tab view to sort data stores in descending order of risk

Click on the first data store to retrieve the next level of detail

The breakdown of the sensitive infotypes split by family is displayed, sorted by descending order of occurrence

Alternatively look at sensitive content by data object by using the 'Data Object' tab where the data object list is sorted by descending risk of exposure

Where is the vulnerable data located?

The 'Data Stores' tab provides information on the precise location of any target paths (if applicable) within the data store in question where sensitive data has been found

It always provides the name of the location where the data store is present





How can I use encryption to protect sensitive data?

Use the CipherTrust Intelligent Protection option to perform discovery and protection in a single step with no manual intervention

An alternative manual approach is to use DDC to find your sensitive data, then use CipherTrust Transparent Encryption (CTE) as a separate manual step to encrypt the data

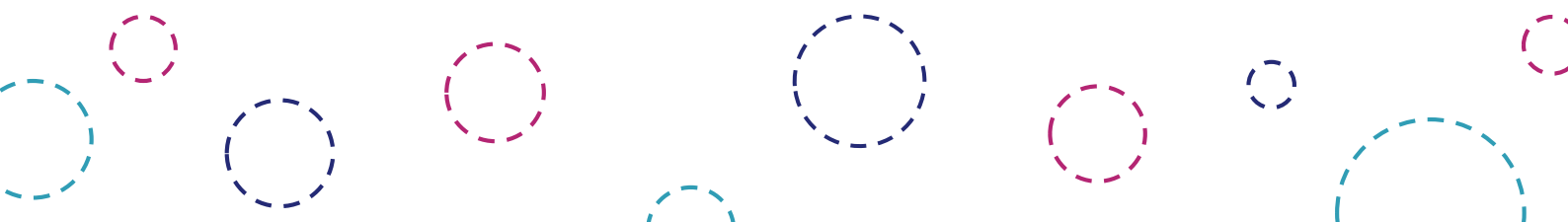
You may also consider other CipherTrust platform protection solutions such as tokenization, data masking or ACLs as an alternative to encryption

What happens if some of my data is already encrypted?

Data with no encryption applied can be scanned and analyzed providing the discovery agent has been granted access to the location or path in question

Data encrypted using CTE can be scanned and analyzed providing the discovery agent has been granted access to the location or path in question within the data store

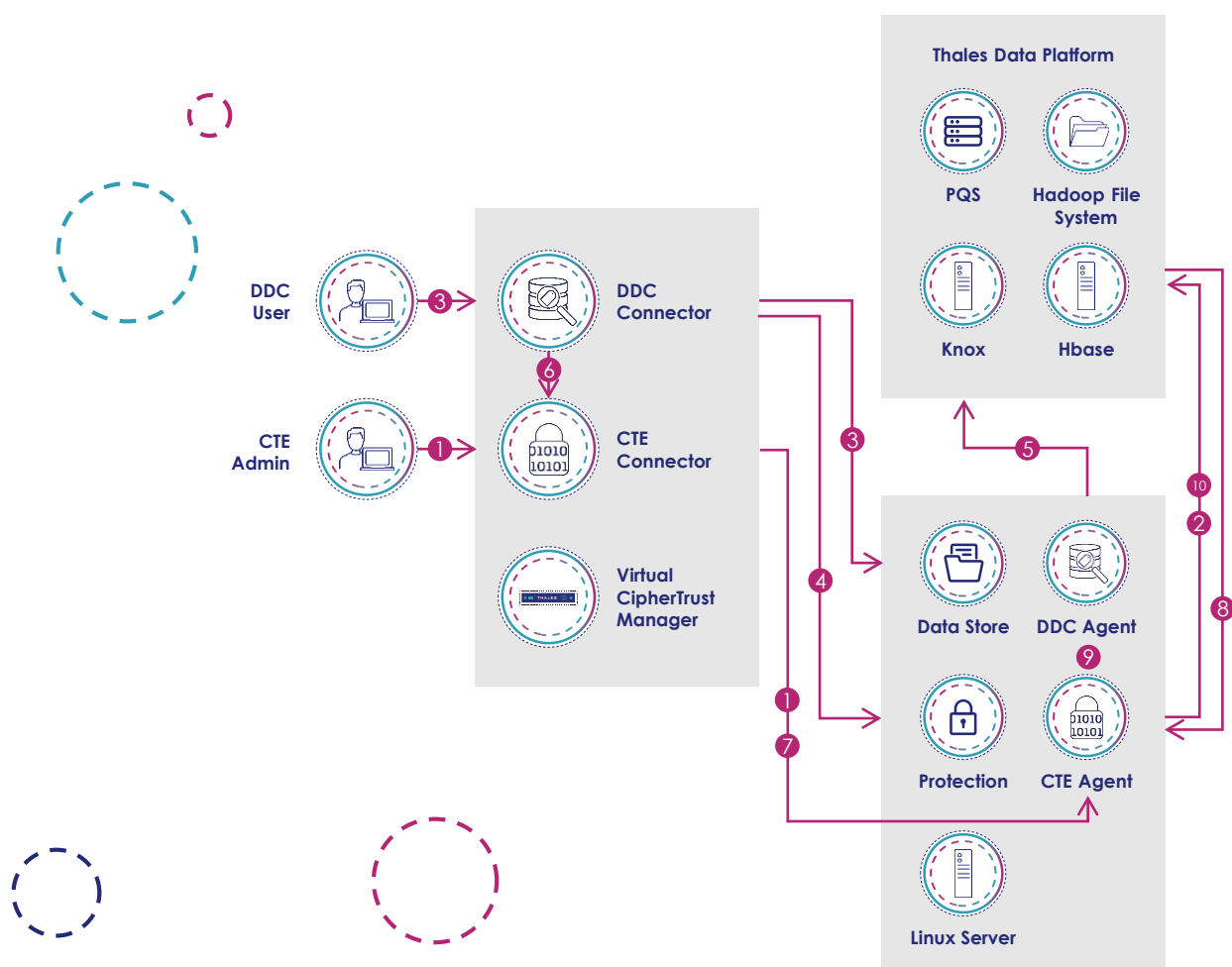
Data that has been encrypted using a third party tool can be scanned and analyzed properly only if DDC has access to the decrypted data – otherwise DDC will scan the encrypted data and not find any match



Discovery and protection in a **single** step



This feature creates an automatic workflow stage that links CipherTrust Data Discovery and Classification (DDC) with CipherTrust Transparent Encryption (CTE) – we call this CipherTrust Intelligent Protection. This is an optional feature that you can switch on or off as required for any given data store or file location which is linked to a GuardPoint accessible to both CTE and DDC.



- 1** CTE Admin creates a policy (using a classification-based resource set) on the CTE Connector and configures the GuardPoint on the CTE Agent
- 2** CTE Agent scans the GuardPoint and uploads the list of files to the Thales Data Platform
- 3** DDC User adds a data store located on the server where the CTE Agent resides and configures the classification profile
- 4** DDC User creates and initiates a scan on the data store linked to the GuardPoint path and enables protection
- 5** DDC Agent stores the scan results in the Thales Data Platform on completion of the scan, including the classification for each of the files linked to the GuardPoint
- 6** DDC Connector notifies the CTE Connector that new scan results are ready
- 7** CTE Connector notifies the CTE Agent that new classifications are ready
- 8** CTE Agent downloads the classification information from the Thales Data Platform
- 9** CTE Agent applies the security rules, to enforce access control, based on policy and classifications
- 10** CTE Agent encrypts each file according to the policy key rule, updating the file status information in the Thales Data Platform at the same time

Reasons for our choice of architecture



Single integrated platform

We decided to make DDC an integral part of the Thales CipherTrust Data Security Platform because, in our experience, data discovery is the foundation of any effective data security strategy. This integrated platform approach enables CipherTrust Manager to act as the central management console for various connectors including data discovery, transparent encryption and tokenization. If you are using CTE already, you will be familiar with the console and will benefit from consistency in how common management tasks such as user groups, data policies, access control and logging are implemented.



Accurate pattern matching

Thales performed an extensive investigation to determine the best data discovery technology in the market. After much research and bench testing, the team experienced a significant difference using GLASS Technology™ over regex. GLASS is a proprietary market-leading technology from [Ground Labs](#) (a Thales Technology Partner) for defining and matching simple and complex patterns of data at scale as opposed to a developer code language not built specifically for purpose. It was designed from the ground up with the express purpose of finding patterns of data in modern data sets across both structured and unstructured data storage scenarios.



High performance discovery

An important goal was to provide a highly efficient and accurate discovery engine. Most data discovery solutions available today use the regular expression (regex) language (originating from the 1950's) to search for patterns inside text strings. This approach requires extensive developer knowledge to implement and the syntax is not based on English grammar. The biggest enemy of regex is performance – as pattern concurrency and complexity increases, the slower the performance achieved. Regex has limits to the patterns it can match accurately, making it far from ideal in the modern world where there are hundreds of different privacy regulations and a broad range of data management issues to address.



Easy customization capabilities

Customization support was another important criteria to supplement the comprehensive prebuilt templates. To address proprietary needs, GLASS data patterns can be written by non-developers using English-style grammar to express how a piece of data or pattern exists. Detailed analysis shows it has overcome all important regex limitations for pattern matching techniques, especially its ability to operate across multiple platform and cloud environments, while being very frugal in terms of CPU overhead. This made GLASS an ideal core component for embedding inside DDC, working efficiently and transparently as part of the CipherTrust platform in identifying and securing critical data.



Flexible licensing model

A consumption model based on total data scanned is used to license DDC, providing close alignment with many cloud subscription models and offering an unlimited number of discovery agents without incurring any additional fees. This makes it much easier for all your teams to have access to the data discovery capabilities, with the only real decision to make is how much overall data discovery capacity is required for your organization. Re-scanning existing data does not subtract from your remaining allowance, only new data scanned is counted.

Security first

Naturally as a market leading security company we design and implement secure solutions. DDC is no exception with strong emphasis on ensuring no sensitive data is stored or vulnerable to exposure:



Ensuring that all data flowing between your data stores and the agents is protected using strong encryption



Providing granular access controls so that you can tailor the solution to your exact needs



Enabling you to secure any sensitive data discovered quickly using the automated workflow associated with CipherTrust Intelligent Protection



Integrating with other solutions



Thales CipherTrust Data Security Platform

Discovery

Encryption and Tokenization

Key Management



Discovery and Classification



Transparent Encryption



Database Protection



Application Data Protection



Tokenization



Enterprise Key Management



Cloud Key Management



CipherTrust Data Security Platform

DDC is optimized for use as part of the CipherTrust Data Security Platform, an integrated suite of data-centric security products and solutions that unify data discovery, protection and control in one platform. Consequently you can cover all data protection needs using a single platform from one vendor, in this case Thales. This provides you with a full suite of data protection capabilities including key management, user access control, file-level encryption, application-layer encryption, database encryption, format-preserving encryption, tokenization and data masking.



CipherTrust Manager

All components utilize CipherTrust Manager as their management console, delivering central management of encryption keys, granular access control and security policy configuration. One key advantage is that our solution is designed to protect your data wherever it resides, on-premises or in the cloud and whatever storage mechanism is in use, such as files, databases, big data or containers. Ultimately you will end up requiring fewer resources dedicated to data security operations, meet your compliance obligations with greater confidence and reduce the business risk to your organization.



It is also possible to deploy DDC alongside other third party data protection tools, such as those supporting data encryption or tokenization for example. It is non-intrusive, complementing tools you have already, rather than replacing them. However, CipherTrust Manager is still necessary since it provides the sole method to launch the data discovery capability and to manage the licenses required for its operation together with keys required for encrypting sensitive data.

Unlike deploying CipherTrust Intelligent Protection, if DDC is used to find the sensitive data and another non-CipherTrust tool from a third party is used to encrypt the data, two distinct steps would be necessary (potentially with a long lapse in-between).

Some of the main reasons Thales customers have been using DDC in conjunction with third party products include:

Trying to fill gaps in coverage since their existing discovery tool cannot scan all their structured and unstructured data

Seeking higher accuracy with data matches to eliminate the limitations they are experiencing with regex technology

Supporting a wider range of operating systems and platforms to interrogate more of their data footprint

We often get asked...

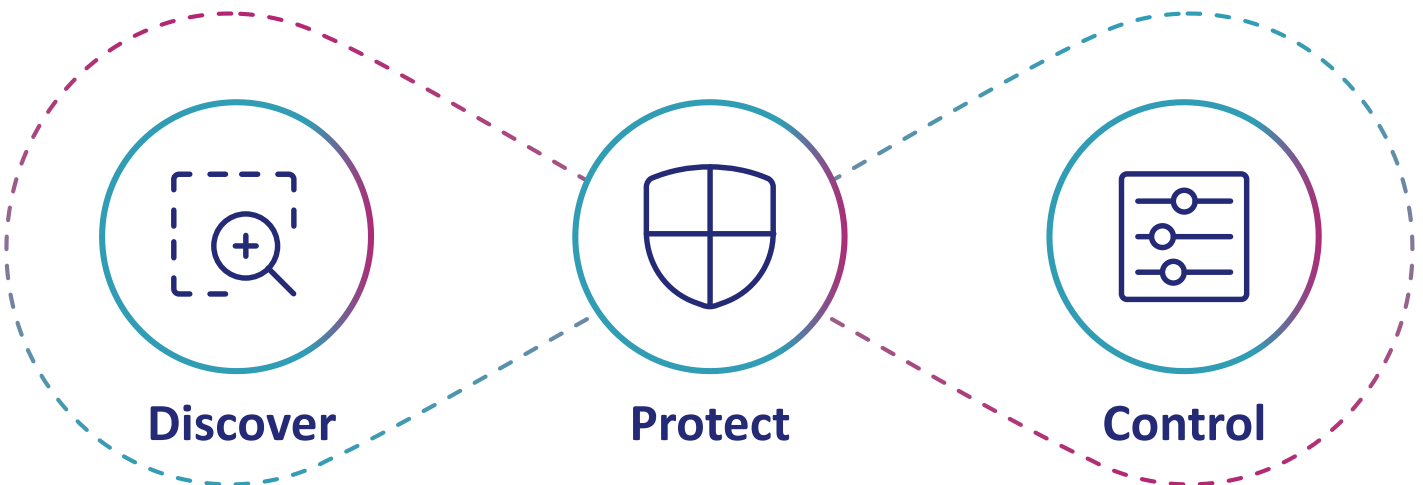
‘What is the difference between CipherTrust Data Discovery and Classification and other Data Loss Prevention (DLP) solutions?’

The DLP solutions focus on preventing sensitive data from leaving the organization's perimeter. DDC focuses on data privacy - identifying sensitive data, and getting a clear understanding of data and its risk. This enables appropriate steps to be taken to protect data and comply with data privacy and data security regulations.



Key takeaways

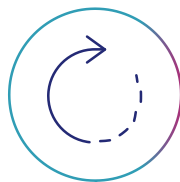
CipherTrust Data Security Platform



Your data management and protection strategy can ultimately be improved significantly using CipherTrust Data Discovery and Classification. In conjunction with CipherTrust Manager and CipherTrust Transparent Encryption it provides considerable visibility and insight into your sensitive data while offering quick and effective protection options. It enables you to:



Block unnecessary data access to reduce risk of data breaches



Keep up to date with the evolving regulatory landscape to remain in compliance



Address digital transformation concerns relating to cloud security to support business and technology goals



www.cpl.thalesgroup.com

Try our free 90 day evaluation to see for yourself how it can help your organization now

Abbreviations



1/3

ACL	Access Control List
CDSP	CipherTrust Data Security Platform
CIFS	Common Internet File System
CIP	CipherTrust Intelligent Protection
CM	CipherTrust Manager
CTE	CipherTrust Transparent Encryption
DB	Database
DDC	CipherTrust Data Discovery and Classification
GUI	Graphical User Interface
NAS	Network-attached Storage
REST API	Representational State Transfer Application Programming Interface
TDP	Thales Data Platform
NAS	Network-attached Storage



Classification profile	A classification profile uses a list of infotypes to define what kind of sensitive information to search for during a scan
Connector	A connector is a generic term which relates to the different licensable products or components (which includes DDC and CTE) that are managed using the CipherTrust Manager console
CTE agent	A CTE agent is a software component that is used to encrypt data associated with GuardPoints defined using CTE policies
Data match	A data match occurs during a scan when a qualified instance of an infotype actively included in the scope of the search is found
Data object	A file or database table located inside a data store is known as a data object
Data store	A data store is the entity where the data is actually stored, with DDC supporting various types - local, network, database, Big Data and cloud
DDC agent	A DDC agent is a software component that is used to scan a data store for specific types of data defined using infotypes in the associated classification profile
Discovery	Data discovery is the process of mapping an organization's data assets including their locations, types and sensitivity levels
False negatives	False negatives are where the discovery process fails to identify one or more valid instances of sensitive data
False positives	False positives are where the discovery process wrongly identifies sensitive data matches where none exist
GuardPoint	A GuardPoint specifies the list of folders or paths to be protected – access to files and encryption of files under the GuardPoint is controlled by security policies
Infotype	An infotype is used to categorize specific data (such as passport numbers or email addresses) to look for during a discovery scan, forming an integral component in the definition of a classification profile
Policy	A policy is a collection of rules that govern data access and encryption
Remediation	Remediation refers to the process of protecting (or securing) data that has been identified as vulnerable as the result of a scan – typical remediation methods include encryption, tokenization, data masking and access control
Resource set	A resource set refers to the files or directories to which the policy will apply together with the associated governing key rules

Risk	A risk is the presence of a sensitive data object in a data store and is directly related to the data matches found in the data object or data store
Scan	A scan is part of the discovery process that is used to search for sensitive data within data stores using criteria defined in classification profiles
Sensitive data object	A data object that contains any data match is known as a sensitive data object
Sensitivity level	The sensitivity level is a mandatory parameter (when defining data stores and classification profiles) relating to the degree of potential vulnerability of any given data object if exposed
Structured data	Structured data is highly organized and easily understood by machine language – examples include names, dates, addresses, credit card numbers, stock information, geolocation and more
Tag	A tag helps group data together and can be specified when creating data stores and classification profiles
Thales Data Platform	The Thales Data Platform (TDP) is a Big Data platform based on Hadoop technology which is used by DDC for various tasks including storage of scan results
Unstructured data	Unstructured data is information that either does not have a pre-defined data model (or schema) or is not organized in a pre-defined manner, making it unsuitable for storage in a relational database

CipherTrust Transparent Encryption for Kubernetes



Challenge: Securing Applications for Kubernetes Environments

Modern applications are increasingly built using containers, which are microservices packaged with their configurations and dependencies. Kubernetes is an open-source software for deploying and managing these containers. Containerized applications can be delivered, deployed and managed faster with Kubernetes to provide improved efficiency through re-usable modular components, cost savings through optimized resource utilization and reduced licensing expenses. However, there are several risks:

- **Privileged user abuse.** By default, Docker processes run with root privileges, administrators have full access to all tenant secrets. This level of untethered access poses multiple risks. Organizations could be subject to privilege escalation attacks, if administrators have unchecked access to container images and the data stored within them.
- **Cross container access.** Poor configuration of permissions can result in multiple containers having access to information that should remain private. Further, when containers are hosted in shared virtualized or cloud environments, critical information can be exposed to third parties.
- **Compliance risks.** Many compliance mandates require stringent access controls and auditing data access. However, many security teams have limited controls available to manage and track access to data held within containers and images. As a result, these teams find it difficult to comply with relevant security policies and regulatory mandates.

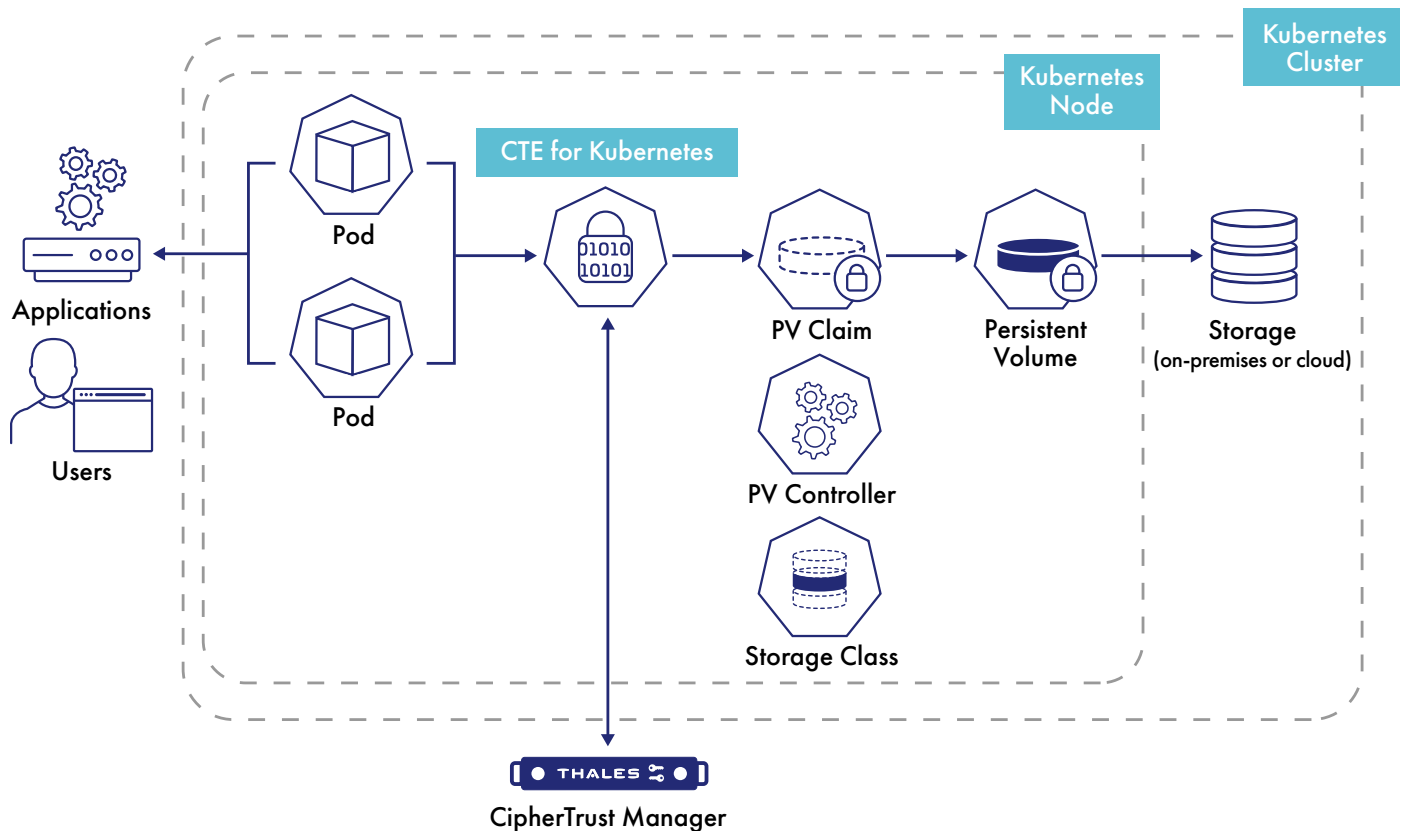
Solution: CipherTrust Transparent Encryption for Kubernetes

CipherTrust Transparent Encryption for Kubernetes delivers in-container capabilities for encryption, access controls, and data access logging, that enables organizations to establish strong safeguards around data in Kubernetes environments. With this extension for CipherTrust Transparent Encryption, data protection can be applied on a per-container basis, both to secure data inside of containers and in external storage accessible from containers, all centrally managed from the CipherTrust Manager..

Benefits

CipherTrust Transparent Encryption for Kubernetes provides:

- **Compliance.** This extension of CipherTrust Transparent Encryption addresses compliance requirements and regulatory mandates for protecting sensitive data such as— payment cards, healthcare records or other sensitive assets.
- **Protection from Privileged-User Threats.** This solution offers encryption with data access controls, enabling privileged users, such as Docker or OpenShift cluster administrators, to operate as regular users, without gaining unauthorized access to sensitive data.
- **Achieve Robust Security.** CipherTrust Transparent Encryption for Kubernetes enforces data security policies wherever the container is stored or used – data centers, virtualized environments, even in cloud implementations. Deploy and use containers where needed for cost effectiveness, control, or performance without having to make any changes to applications, containers, or infrastructure sets.



Features

- **Comprehensive Data Security Safeguards.** CipherTrust Transparent Encryption for Kubernetes extends CipherTrust Transparent Encryption, enabling security teams to establish data security controls inside of containers. With this extension, you can apply encryption, access controls, and data access logging on a per-container basis. Encryption can be applied to data generated and stored locally within the container and to data mounted in the container by network file systems.
- **Scalable Transparent Encryption.** Provides data security controls without having to make any changes to applications, containers or infrastructure sets. It enables a single policy to be applied to all containers within a Kubernetes cluster, or distinct policies applied to each container within a cluster. This solution can scale up or shrink down a Kubernetes environments as business needs change.
- **Granular Access Controls and Visibility.** CipherTrust Transparent Encryption for Kubernetes offers the detailed visibility and control you need to comply with the most stringent policies and mandates. With this Kubernetes security solution, enterprises can establish granular access policies based on specific users, processes, and resource sets within containers. Finally, this solution can establish isolation between containers, so only authorized containers can access sensitive information.

CipherTrust Manager

The CipherTrust Manager centralizes key, policy, and log management for the CipherTrust Data Security Platform including CipherTrust Transparent Encryption. It is available in both virtual and physical form-factors for securely storing master keys with an elevated root of trust. These appliances can be deployed on-premises as well as in private or public cloud infrastructures.

About Thales Cloud Protection & Licensing

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.



CTE for Kubernetes

Version: v.1.7.0 (latest)

30 January 2026

Contents

CTE for Kubernetes Administration	4
Getting Started with CTE for K8s	4
Container Image Fingerprint	4
Introduction to CipherTrust Transparent Encryption for Kubernetes	6
Configuration Concepts	9
CTE for Kubernetes Terminology	15
Installation Methods	16
Using Operator	17
Introduction to the Kubernetes Operator Resource	17
Installation, Deployment and Configuration	19
Installing and Managing CTE for Kubernetes in an Operator Environment	23
Kubernetes Operator Maintenance	30
Using Helm with CTE for Kubernetes	31
Installing and Deploying with Helm	31
Migrating a CTE install from Helm to Operator deployment	34
Deploying in an Air-Gapped Kubernetes Cluster using HELM	35
Upgrading CTE for Kubernetes with a Helm-based Installation	37
Deployment Methods	39
Deploying with a Custom Resource Definition	40
Deploying CTE for Kubernetes in an Air-Gapped OpenShift Container Platform Cluster using CTE for Kubernetes Operator	41
Preparing a CTE for Kubernetes Volume for Deployment	49
Preparing CipherTrust Manager	50
Create a CTE Policy for Kubernetes	50
Using Kubernetes service accounts	50
Prepare a file system persistent volume	54
Deploying Pods/Services in a Kubernetes Cluster	56
Deploying CTE for Kubernetes Storage Classes	58
Using Dynamic PVCs with CTE for Kubernetes	60
Creating a CTE for Kubernetes Claim Against the PVC	64
Individual Guard Policy Tuning	65
Create a CTE-CSI protected Persistent Storage Claim by deploying an Application/Staging pod	70
Using Taints and Tolerations	73
Allow only Trusted Pods to mount to CTE for Kubernetes volumes	77
Data Transformation	80
Using Initial Data Transformation for CTE for Kubernetes	81
Applying Data Transformation Key Rotation to CTE for Kubernetes	84
Decryption using Data transformation with CTE for Kubernetes	87
Volume Management	88
Volume Cloning Support	88
Volume Snapshot and Restoring	94

Volume Resizing and Expansion.	105
Advanced Configuration and Troubleshooting	107
Using External Certificates with CTE for Kubernetes	107
Troubleshooting.	111

CTE for Kubernetes Administration

- [Getting Started with CTE for K8s](#)
- [Volume Management](#)
- [Advanced Configuration and Troubleshooting](#)

Getting Started with CTE for K8s

Complete the following sections in order to ensure success with CTE for Kubernetes.

1. [Obtain the proper version of CTE for Kubernetes with the Container Image Fingerprint](#)
2. [Introduction to CipherTrust Transparent Encryption for Kubernetes](#)
3. [Installation Methods](#)
4. [Deployment Methods](#)
5. [Preparing a CTE for Kubernetes Volume for Deployment](#)
6. [Data Transformation for CTE for Kubernetes](#)

Container Image Fingerprint

Thales recommends that customers use container images from trusted repositories for their Kubernetes deployments to ensure that the containers have not been tampered with.

Verify that the CTE for Kubernetes Container Image Fingerprint matches the version and architecture that you are installing:

1.7.34

Release Date	Product	Version	Tag	Architecture	Image Digest
2026-01-07	CTE for Kubernetes Operator	1.7.0	1.7.8	Image Index (Arch independent)	sha256:d30ff765703db69846f0302cb06aa5315641796d56342a9be4549fb171414871
2026-01-07	CTE for Kubernetes Operator	1.7.0	1.7.8	amd64 (x86)	sha256:cb579438374811338b203c7363ce1318886903f47a2c443aca145eddd8071fb3

Release Date	Product	Version	Tag	Architecture	Image Digest
2026-01-07	CTE for Kubernetes Operator	1.7.0	1.7.8	arm64	sha256:28206597b1722a439ad16025ea755022f3db6dbda6cdd4d3d6d0b293a061eb8a
2026-01-07	CTE for Kubernetes	1.7.0	1.7.0.34	Image Index (Arch independent)	sha256:ce3214491cfe53b1309b2861c86584faeec76e14399f1502259713dabe54eec7
2026-01-07	CTE for Kubernetes	1.7.0	1.7.0.34	amd64 (x86)	sha256:cd612af9e2317965df91a4a48bf7a62bdb815efd88fe8379683738694c7ffac3
2026-01-07	CTE for Kubernetes	1.7.0	1.7.0.34	arm64	sha256:abebe85e6a8a1038e21d6d1cfe083d7b4e9edf244fcd68857da58d6bca3110ca

1.6.50

Release Date	Product	Version	Tag	Architecture	Image Digest
2025-06-06	CTE for Kubernetes Operator	1.6.0	1.6.5	Image Index (Arch independent)	sha256:d58cbb65d758fd0b9d4cfba4cb48a35bdaebe2e239fb25edb9b491cd8c2c3f73
2025-06-06	CTE for Kubernetes Operator	1.6.0	1.6.5	amd64 (x86)	sha256:e7240cb305b3afda5cd4af01d2b3213fc6156fefc30dc0c4f811b886d331bbd2
2025-06-06	CTE for Kubernetes Operator	1.6.0	1.6.5	arm64	sha256:a5fe74d1f5fabf571d81f9ec5b8f354a2447c294e1816653f4fda5cdd56bc4b4
2025-09-26	CTE for Kubernetes	1.6.0	1.6.0.50	amd64 (x86)	sha256:de6bb33666a04fbe44a481da8a4be4fa5365bcea28a5cd60

Release Date	Product	Version	Tag	Architecture	Image Digest
					762b9f400e7477ce
2025-09-26	CTE for Kubernetes	1.6.0	1.6.0.50	arm64	sha256:3117ae5c912cc9bd91f6b58f0a034a613116503bed6152b4dbbd9ee1529d3a1d

Introduction to CipherTrust Transparent Encryption for Kubernetes

Kubernetes is an open-source container-orchestration system that aims to provide a "platform for automating deployment, scaling, and operations of container workloads".

CTE for Kubernetes is an implementation of CipherTrust Transparent Encryption with native support for Kubernetes through the implementation of a CSI (Container Storage Interface) driver. Unlike traditional CTE, product installation and Guard Policy management is done through Kubernetes. This means that as the cluster scales up with more nodes, CTE for Kubernetes scales with it. CTE for Kubernetes is designed to protect Kubernetes Persistent Storage Claims that are backed by storage with filesystem semantics.

In order to support customers with diverse workloads, registration to CipherTrust Manager has been decentralized from the cluster nodes/ hosts operating system. Registration now happens through the use of Storage Classes which allows for a single cluster, and even a single node, to register different CTE for Kubernetes groups, each with a different set of policies and keys.

CTE Agent and CSI driver are deployed as container images. CTE devices are exposed as Persistent Storage volumes and Customer application containers do not need to be modified.

This document shows some examples on how a customer can protect their pod's persistent storage data through the use of CTE for Kubernetes.

Prerequisites

The CTE for Kubernetes solution requires the following:

- Kubernetes v1.22 or subsequent versions [Kubernetes IO releases](#)
- Available Persistent Storage for protecting with CTE for Kubernetes
- CipherTrust Manager v2.8 and subsequent versions
- Kubernetes nodes with access to the Internet to fetch CTE for Kubernetes images from Docker Hub
- Public Docker Hub credentials for pulling the images
- Working Kubernetes Cluster that can communicate with the cluster using `kubectl`

Required Privileges

Administrators can use security context constraints (SCCs) to control permissions for pods with an OpenShift Cluster. These permissions include actions that a pod can perform and what resources it can access. You can use SCCs to define a set of conditions that a pod must run with to be accepted into the system. You can also use the Security Context Constraints to provide additional privileges to the Service Accounts that are used by the CTE for Kubernetes driver in an OpenShift Cluster. A subset of these permissions are specified in the Pod specification for use in Kubernetes where SCC is not available.

For a container to exploit the System Administrator permissions, the container must run as a privileged container.

System Administrator Permissions	Value	Definition	Resources Used
<code>allowedCapabilities</code>	<code>SYS_ADMIN</code>	The CSI Controller component performs tasks like create, publish, unpublish, delete, snapshot, and restore volumes, on any node in the Kubernetes cluster. Without the <code>SYS_ADMIN</code> capability, the driver does not work and CTE functionality is lost.	
<code>allowPrivilegedContainer</code>	<code>true</code>	For a container to have <code>SYS_ADMIN</code> capability, the container must run as a privileged container.	
<code>allowHostDirVolumePlugin</code> <code>volumes</code>	<code>true</code> <code>emptyDir</code> <code>hostPath</code> <code>projected</code>	The CTE for Kubernetes driver mounts the directories from the host node into the containers. These two permissions allow for the CTE driver to mount these directories as volumes inside the container.	Directory examples: <code>/etc/kubernetes/</code> <code>/var/lib/kubelet/</code> <code>plugins_registry/</code> <code>/var/lib/kubelet/</code> <code>pods</code> It also mounts one of the following directories, depending on which CRI is configured: <code>/run/crio/</code> <code>crio.sock</code> <code>/run/cri-dockerd.sock</code> <code>/run/containerd/</code> <code>containerd.sock</code>
<code>allowHostPID</code>	<code>True</code>	The CTE for Kubernetes driver requires the process PID to track and authorize application processes accessing protected data.	
<code>runAsUser</code>	<code>type: RunAsAny</code>	Controls which user ID containers can use.	

System Administrator Permissions	Value	Definition	Resources Used
		RunAsAny allows processes inside the container to run with any user ID, including root, in that container's context.	
seLinuxContext	type: RunAsAny	Controls which SeLinux context the container can use. RunAsAny allows changes to the SELinux Context assigned to the namespace in which the container is running.	
fsGroup	type: RunAsAny	Controls which fsGroup owns the volumes created by the container. RunAsAny allows users to own any files/directories created by the container.	

Minimum System Requirements

Pod Type	Memory (RAM)	CPU
CTE CSI Controller	512Mb	1
CTE CSI Node Server	512Mb + 100Mb per Guard Policy	2
CTE CSI Staging pod	5Mb	0.1

Note

- Encryption CPU requirements on the node server are highly dependent on the workload that is accessing the CTE for Kubernetes encrypted volume. Additional CPU resources may be required depending on the expected number of IOPS for the application
- Applying hard memory limits on the CTE for Kubernetes node server pods can result in the pod being evicted. This will cause a service interruption to the CTE for Kubernetes volume. Hard memory limits are not recommended for the CTE node server.
- Staging pods do not consume any CPU resources after starting and very minimal memory resources after starting.

Audience

This document is intended for personnel responsible for maintaining your organization's security infrastructure. This includes security officers, key manager administrators, and network administrators.

All products manufactured and distributed by Thales Group are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.

Thales expects that the users of this product are proficient with security concepts and knowledgeable in all aspects of Kubernetes cluster administration and management.

Users must be able to:

- Install the CTE for Kubernetes CSI driver
- Create RBAC (role-based access control) rules
- Add Persistent Storage Claims to a namespace within the Kubernetes cluster

Limitations

- Killing CTE CSI pods can leave a pod with protected volumes in an unusable state. You should terminate all pods using a CTE protected volume before attempting to stop the CTE for Kubernetes service pods.

Support for Managed Clouds

CTE for Kubernetes can be deployed in the following Cloud environments:

- Amazon Elastic Kubernetes Service (AWS)
- Azure Kubernetes Service (Azure)
- Google Kubernetes Engine (GKE)
- Red Hat OpenShift

Kubernetes Terminology

For relevant Kubernetes definitions, see [CTE for Kubernetes Terminology](#) for more information.

Kubernetes Configuration Concepts

For explanations and examples of the configuration concepts for CTE for Kubernetes, see [Kubernetes Configuration Concepts](#) for more information.

Configuration Concepts

The following concepts are important to understand for success with CTE for Kubernetes.

For more information on Kubernetes, consult the [Kubernetes documentation](#).

CipherTrust Manager CTE for Kubernetes Storage Groups

CTE for Kubernetes configuration is accomplished with CipherTrust Managers(CM) through **K8s Storage Groups**. Security administrators define which combination of Kubernetes Storage Classes and Namespaces are allowed to register to CM. Once a K8s Storage Group is defined, the user can create a **GuardPolicy** that can be used by the CTE-CSI volume to create, view, edit, and delete Kubernetes (K8s) storage groups on the K8s Storage Groups page of the CipherTrust Manager GUI.

Note

A single node in a Kubernetes cluster can register itself multiple times to different K8s Storage Groups based on different Storage Class and Namespace usage patterns. This allows for separation of security context based on the security needs of the Kubernetes namespace that the application pod will be running in.

- See [Managing Kubernetes Storage Groups and Clients](#) for more information about creating storage groups in CipherTrust Manager.

Using CipherTrust Manager with CTE for Kubernetes

On the CipherTrust Manager, to setup your Kubernetes environment:

1. Create a [Kubernetes Client](#).
2. Create a Kubernetes storage group, which includes a client profile which contains a Guard Policy. The Guard Policy must contain:
 - Supported access controls
 - Resource sets
 - Process sets
 - Signature sets
 - User sets with UIDs
3. Register the Kubernetes Client

Registration is the process of configuring a Kubernetes (Kubernetes) client with CipherTrust Manager. This process creates SSL certificates for further communication between the CipherTrust Manager and the Kubernetes client.

After registration, the Kubernetes client can communicate with the CipherTrust Manager. All of the Guard Policies applied to the Kubernetes storage group are automatically added to the Kubernetes client. The client configuration is then built for the Kubernetes client (exactly like a CTE client) and sent to the CipherTrust Manager.

After successful registration, the Kubernetes client appears on the Kubernetes Clients page of the CipherTrust Manager GUI. The client status becomes Healthy.

CTE for Kubernetes CSI driver

Protecting Persistent volumes in a Kubernetes Cluster

Unlike traditional CSI drivers, the CTE for Kubernetes CSI driver (CTE-CSI) does not store persistent data. CTE-CSI relies on other Persistent Volumes(PV) in the cluster for actual data storage. The CTE-CSI driver provides a layer on top of those PVs that adds encryption and data protection. This encryption is transparent to the Pod that wants to use a CTE-CSI volume. Each CTE Persistent Volume Claim (PVC) must be backed by a filesystem-capable PV that is claimed by a PVC in the same namespace.

Each CTE PVC must be configured to use a GuardPolicy, configured in CM, that will provide the required access control and key rules.

When protecting a PV, you must:

1. Choose a PVC in the cluster to protect and to act as your data source.
2. Create a CTE-CSI Storage Class with a parameter to connect to CM.
3. Create a CTE-CSI PVC with a GuardPolicy and a data source for PVC parameters.
4. Add the CTE-CSI PVC to a Pod as a regular volume.

Supported Persistent volumes

CTE Kubernetes is expected to work well with most persistent volumes types with file system semantics. Raw volumes with no filesystem are not supported at this time.

For more information, see [Types of Persistent Volumes](#)

The following table lists the **unsupported** PVC volumes:

Storage Type	Supported	Full Name
configMap	No	ConfigMap
downwardAPI	No	Downward API
emptyDir	No	EmptyDir
fc	No	Fibre Channel (FC) storage
flocker	No	Flocker storage
gitRepo	No	Git repository
hostPath	No	HostPath volume
projected	No	Projected volume
quobyte	No	Quobyte volume
secret	No	Secrets

Storage Type	Supported	Full Name
storageOS	No	StorageOS volume
subPath	No	Subpath volume

CTE service pods

CTE for Kubernetes is implemented as a Kubernetes CSI driver. A typical CTE-CSI driver deployment will contain one **cte-csi-node-xxxx** for every worker node in the cluster. These pods are the main workers for CTE for Kubernetes and perform tasks such as volume attachment, CTE agent registration and volume encryption within the cluster node. The cluster will also contain at least one **cte-csi-controller-xxxx** which is in charge of dynamic provisioning of CTE-CSI volumes. These two types of service pods run in the **kube-system** namespace and are able to serve volumes across different namespaces in the cluster.

Warning

It is important to *never* restart a node's **cte-csi-node-xxx** pod if this node still has CTE-CSI volumes attached to it from other application pods. Killing a busy CTE-CSI pod will result in application Pods losing their connection to the encryption service. This will render the volume on those pods inaccessible, and can result in data loss. When this happens, the application Pod must be restarted to restore proper functionality.

CTE for Kubernetes registration

CTE for Kubernetes dynamically registers the agent when a CTE-CSI volume attachment request is performed on a node. CTE will use the configuration parameters of the StorageClass, as well as the Namespace of the Pod, requesting the volume to attempt registration. Registration only succeeds if the Storage Class name and the Pod's Namespace matches that of the Storage Group defined in CipherTrust Manager. The CTE-CSI driver will automatically deregister itself for a given StorageClass/Namespace combination once all Pods that were consuming a particular registration in the node have been evicted from the node. The amount of time that the CTE-CSI driver waits from last eviction to deregistration can be configured through the StorageClass **registration_period** parameter

Note

Registration and deregistration to CM is dynamic, based on pod usage within the nodes in the cluster. This can quickly exhaust registration token client capacity. Therefore, it is recommended that the user define a CipherTrust Manager registration token with a large client capacity.

Warning

Kubernetes will always retry and mount a Pod's volume in the event of a failure. If that failure is due to a registration failure (for example, non-matching StorageClass/Namespace parameters in the K8s Storage Group), retry attempts by Kubernetes will quickly exhaust registration token client capacity.

CTE-CSI Storage Class

For CTE for K8s, as soon as you create a pod to protect a cte-csi volume, cte-csi is using a storage class. Once this storage class is created, it automatically creates a K8s client and registers it with CipherTrust Manager.

When first creating Storage Class objects, Administrators set the name and other parameters for a class in a yaml file. Note that objects cannot be updated once they are created.

Important standard Kubernetes specification definitions

The following list defines the parts of the yaml file:

- **metadata.name:** Name of the StorageClass. This name **MUST** match the StorageClass name specified in the CM K8s Storage Group.
- **provisioner:** Determines what volume plugin is used for provisioning Persistent Volumes. This field must be specified. For CTE for Kubernetes, it is always defined as **csi.cte.cpl.thalesgroup.com**
- **reclaimPolicy:** Determines what to do with the PersistentVolume once its claim has been deleted. CTE PVs do not hold any data. They act solely as configuration placeholders for real PVs. The policy **MUST** be set to **Delete** as they are not usable outside of the context of its PVC that created it.
- **allowVolumeExpansion:** CTE for Kubernetes does not store data so these volumes are not expandable. This does not prevent the source unprotected volume from being expandable. This **MUST** be set to **false**

CTE-CSI Storage Class parameters

- **key_manager_addr:** Domain name or IP address of the CipherTrust Manager. This is a required field.
- **k8_storage_group:** Name of the CipherTrust Manager K8s Storage Group. This is a required field.
- **registration_token_secret:** Kubernetes Secret with CM registration token. The CM registration token must be saved into the Kubernetes secret with as a base64 encoded string. This is a required field.
- **registration_period:** Time in minutes to wait before deregistering from the CipherTrust Manager once all volumes on a node have been unguarded. Parameter must be added as a string integer value. Default "10" minute. This field is optional.

Example

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: <CHANGE ME to K8s Storage Class>
provisioner: csi.cte.cpl.thalesgroup.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: false
parameters:
```

```
#Domain name or IP address of the CipherTrust Manager (Required)
key_manager_addr: <CHANGEME to your CM ADDR>

#Name of the CipherTrust Manager K8s Storage Group. (Required)
k8_storage_group: <CHANGEME to your K8s Storage Group name>

#Kubernetes Secret with CM registration token (Required)
registration_token_secret: <CHANGEME to K8s secret>

#Time in minutes to wait before unregistering from the CipherTrust Manager
#once all volumes have been unguarded. Parameter must be added as a string
#integer value. Default "10" minute. (Optional)
registration_period: "10"
```

CTE Persistent Volume Claim

CTE-PVC is the protected Persistent Volume Claim. You can use it to configure CTE-K8s features and functionalities.

CTE-PVC is used by the protected pods. Configuration information like CipherTrust Manager Guard Policy information, and underlying source PVC information can be provided with this CTE-PVC.

You can configure many features with a CTE-PVC, such as:

- Dataxform
- Taints & Tolerations
- Guard Policy Tuning
- Volume management (volume cloning, volume snapshot, etc)

Important standard Kubernetes spec definitions

The following list defines the parts of the yaml file:

- **metadata.name:** Name of the CTE-CSI PVC. Use this PVC name in your Pod definition instead of the unprotected source PVC.
- **metadata.annotations.csi.cte.cpl.thalesgroup.com/policy:** CTE for Kubernetes GuardPolicy name. This GuardPolicy is located on the CipherTrust Manager and should match a policy name available on the storage class for this PVC. This parameter is required.
- **metadata.annotations.csi.cte.cpl.thalesgroup.com/source_pvc:** Name of the unprotected source PVC that will be protected by this CTE-PVC. This parameter is required.
- **spec.storageClassName:** CTE-CSI StorageClass name. This parameter is required
- **spec.resources.requests.storage:** Kubernetes PV capacity request. This parameter is required by Kubernetes but ignored by CTE-CSI as capacity is dictated by the source PVC.

Example

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cte-claim
  annotations:
    # CTE for Kubernetes GuardPolicy name. This GuardPolicy is located on the
    # CipherTrust Manager and should match a policy name available on the
    # storage class for this PVC. (Required)
    csi.cte.cpl.thalesgroup.com/policy: <CHANGEME CTE for Kubernetes policy>

    # Name of the unprotected source PVC that will be protected by this CTE-P
VC.
    # (Required)
    csi.cte.cpl.thalesgroup.com/source_pvc: <CHANGEME to a regular PVC to pro
tect>
spec:
  storageClassName: <CHANGEME to the CTE-CSI StorageClass>
  accessModes:
    - ReadWriteMany
  resources:
  requests:
    # This parameter is required by Kubernetes but ignored by CTE-CSI.
    storage: 1Ki

```

CTE for Kubernetes Terminology

The following Kubernetes terms are important to understand when working with CTE for Kubernetes containers.

Container Storage Interface

Container Storage Interface (CSI) is a plugin for Kubernetes that allows the implementation of third-party persistent storage solutions. This provides Kubernetes users more options for storage. CTE for Kubernetes is implemented as a CSI plugin which allows Thales to provide encryption and data protection while using Kubernetes volume configuration semantics.

Persistent Volumes

A PersistentVolume (PV) is a piece of storage in the cluster that has been provisioned by an administrator, or dynamically provisioned using [Storage Classes](#). It is a resource in the cluster. PVs are volume plugins like volumes, but have a lifecycle independent of any individual Pod that uses the PV. This API object captures the details of the implementation of the storage.

Unlike other storage plugins, CTE PVs are virtual in nature and do not hold store actual data. They functions as an overlay on top of other Kubernetes PVs. It then encrypts the contents of that PV.

Persistent Volume Claims

A Persistent Volume Claim (PVC) is a request to bind a PV to a namespace. Once claimed, a PV can be attached to any Pod within the same namespace. A PV that has not been claimed is not used by any Pod.

Storage Class

Provides a method for administrators to describe the classes of storage they offer. They are the foundation for defining parameters for dynamic provisioning.

Namespaces

In Kubernetes, namespaces provide a mechanism for isolating groups of resources within a single cluster. Namespaces are a way to divide cluster resources between multiple users.

Note

- Namespaces **cannot** be nested inside one another. Each Kubernetes resource can only reside in one namespace.
- Avoid creating namespaces with the prefix `kube-`. It is reserved for Kubernetes system namespaces.
- Names of the resources need to be unique within a namespace, but not across namespaces.

Pods

A Pod is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers. A Pod's contents are always co-located and co-scheduled, and run in a shared context. A Pod models an application-specific "logical host": it contains one or more application containers which are relatively tightly coupled. A pod consumes the nodes resources. In non-cloud contexts, applications executed on the same physical or virtual machine are analogous to cloud applications executed on the same logical host. Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.

CTE Staging Pod

A CTE staging pod is a special pod that mounts on a Persistent Volume available to a Kubernetes cluster, so that CTE can attach the protection layer on top of the PV. CTE for Kubernetes automatically generates a staging pod on the same node where a Pod with a CTE PVC volume is deployed.

Storage Group

Similar, conceptually, to a Client Group. A Kubernetes Storage Group is created in CipherTrust Manager and consists of Kubernetes clients, similar to how a Client group consists of CTE Clients.

Installation Methods

CTE for Kubernetes supports two installation methods: Helm Charts and Operator. Beginning with version 1.5.0, the Operator is the default installation method. The desired installation method can be specified using command-line arguments in the `deploy.sh` script.

Operator Installation

The Operator-based installation deploys a dedicated operator pod that monitors the usage of CTE PVCs across the cluster. It automates the upgrade of CTE for Kubernetes by detecting when nodes are no longer using CTE and proceeds with the upgrade process accordingly.

- [Install Operator](#)

Helm Charts Installation

Helm offers a manual method to deploy CTE for Kubernetes. It places the responsibility on the user to manage node upgrades. Before upgrading, users must ensure that no applications on the node are actively using a CTE Persistent Volume Claim (PVC). Manual draining of the node is required to complete the upgrade process.

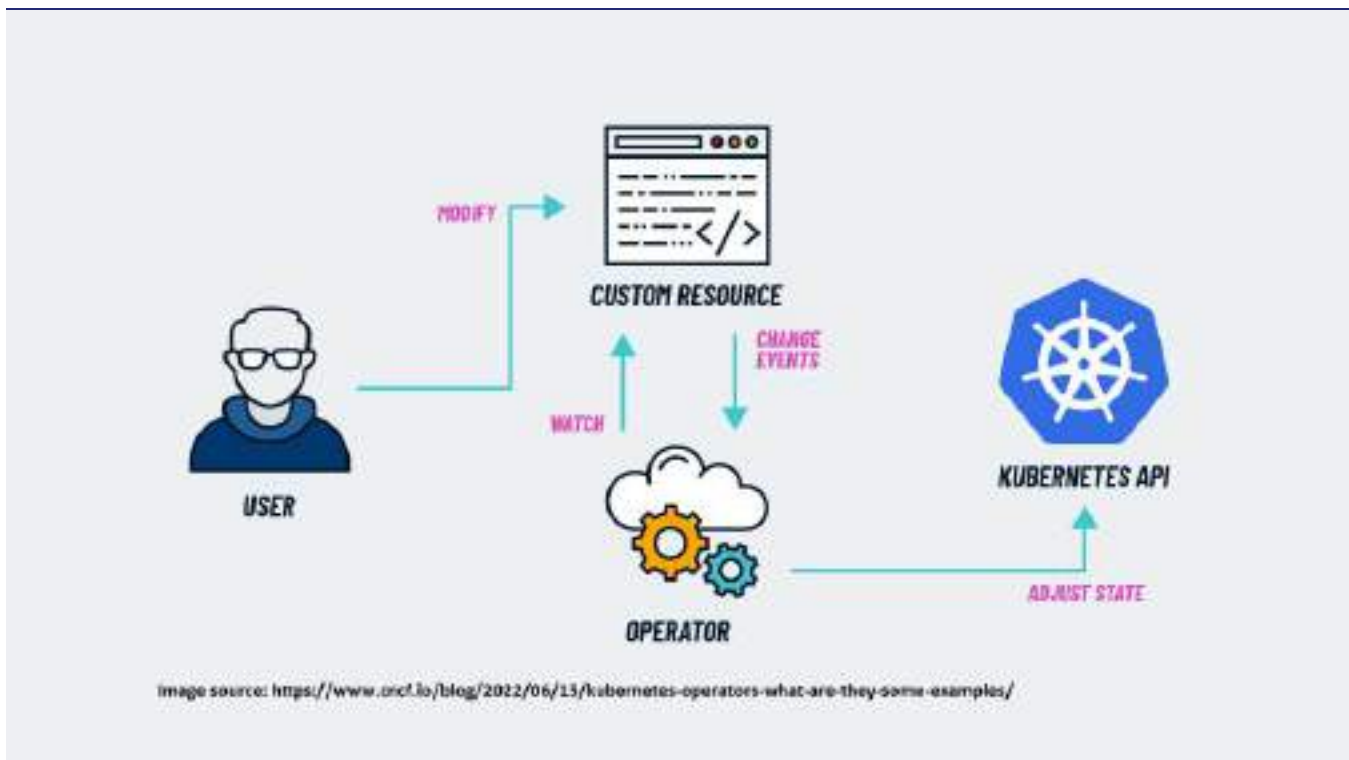
- [Installing with Helm](#)

Using Operator

- [Introduction to the Kubernetes Operator Resource](#)
- [Installation, Deployment and Configuration for CTE for Kubernetes](#)
- [Installing and Managing CTE for Kubernetes in an Operator Environment](#)
- [Kubernetes Operator Maintenance](#)

Introduction to the Kubernetes Operator Resource

The CTE for Kubernetes Operator is an OpenShift operator that Thales created for CTE for Kubernetes. The Kubernetes Operator can deploy, monitor, upgrade and delete CTE for Kubernetes. When the Operator is deployed, its controller deploys the CTE for Kubernetes driver in the OpenShift cluster. The manifests required to deploy the CTE-K8s driver are bundled with the operator.



Kubernetes has various default resources like Pod, Deployment, DaemonSet, etc. When you define a manifest for instantiating one of those resources, you must specify the **Kind** as Pod/Deployment/DaemonSet etc. Kubernetes provides a default set of controllers that understand the resource definitions and know how to manage their life cycle. For instance, the Deployment Controller manages the Create/Update/Delete of the Deployment resource.

Kubernetes architecture allows users to extend the API server in a way that users can create their own custom resource (CR) and write their own controller to manage the custom resource. An operator bundles a CR and the Controller that manages the CR. An operator can watch resources across the cluster and take actions when required. The operator contains a CR, an API for managing a CR, and a custom controller that manages this resource. When you install the CTE for Kubernetes Operator on an OpenShift cluster, the operator registers the new CR and the controller with the Kubernetes API server. Whenever the API server receives a request to create a resource, where `Kind=CTEK8sOperator`, it passes on the request to CTE for Kubernetes Operator's Custom Controller. The controller contains all of the logic needed to complete tasks before, during or after the deployment of the driver.

For more information on how to use operators to manage other applications, refer to the [Kubernetes Operator pattern](#).

Note

- The CTE for Kubernetes feature is supported with CTE for Kubernetes v1.2.0 and subsequent versions.
- For CTE for Kubernetes v1.5.0 and subsequent versions, Operator is the default deployment method.
- CTE-Kubernetes Operator version v1.5.9 has been certified with Redhat and published on the Redhat portal: [RedHat Portal](#).
- For Kubernetes cluster, including managed Kubernetes clusters in the cloud, namely Google GKE

cluster, Amazon EKS cluster and Microsoft's AKS cluster, the operator is also certified and available at: [Kubernetes Catalog](#).

- CTE for Kubernetes Operator supports both x86_64 and arm64 deployment

Installation, Deployment and Configuration

Getting CTE for Kubernetes deployment files

Obtain CTE for Kubernetes through the `yaml` files available in the `cte-csi-deploy` Git repository at:

```
git clone https://github.com/thalescpl-io/ciphertrust-transparent-encryption-kubernetes.git
```

Prerequisites

Installation Dependencies for Kubernetes Operator

CTE for Kubernetes v1.5.0 and subsequent versions install with the CTE for Kubernetes Operator by default. Therefore, you must install the Operator Lifecycle Manager (OLM) if you choose to use Kubernetes Operator. If OLM is not properly installed, the operator installation fails.

OLM is a framework for managing the lifecycle of Kubernetes Operators. It simplifies the process of installing, managing, and upgrading operators and their associated applications within a Kubernetes cluster.

When installing CTE for Kubernetes, it requires the following dependencies:

Dependency	Deployment Method
kubectl	Helm / Kubernetes Operator
Operator LifeCycle Manager (OLM)	Kubernetes Operator

Creating the Required Kubernetes Secret

For CTE for Kubernetes Operator v1.5.9, the Operator Controller Manager has a change that requires an additional step to be carried out by the cluster administrator before deployment. **This prerequisite is mandatory.**

There has been a change in how the Kubernetes Operator Controller Manager downloads images. The `kube-rbac-proxy` used by the `operator-controller-manager` pod performs RBAC authorizations with the Kubernetes API Server. The CTE for Kubernetes Operator previously pulled this image from `gcr.io`. However, this image has been deprecated and will be removed from GCR. The CTE for Kubernetes Operator has shifted to pulling the image hosted by `registry.redhat.io`. While `gcr.io` is a public registry and does not require authentication for pulling images, `registry.redhat.io` requires authentication.

Due to this change, it is now a requirement to create a specific Kubernetes secret named `rh-kube-proxy-secret`. The cluster administrator must create this secret in the namespace in which they are deploying the operator. Without this secret, attempts to create new pods will result in errors due to the authentication

requirement of `registry.redhat.io`.

OpenShift Environment

1. Use a login account created on redhat to create the secret using the following command:

```
oc create secret docker-registry rh-kube-proxy-secret --docker-username="<username on Redhat portal>" --docker-password="<Redhat portal password>" --docker-server="registry.redhat.io" --namespace="<namespace in which CTE-K8s Operator is to be deployed>"
```

If you miss this step, you will encounter the following in an `ErrImagePull/ImagePullBackOff` in the Events section of the `cte-k8s-operator-controller-manager-XXXXXXXXXX-YYYYYY` pod:

```
Events:
Type      Reason      Age          From          Message
----      -
Normal    Pulling      18m (x3 over 19m)    kubelet        Pulling image "registry.redhat.io/openshift4/ose-kube-rbac-proxy-rhel9:v4.17.0-202503121206.p0.g7718265.assembly.stream.el9"
Warning   Failed       18m (x3 over 19m)    kubelet        Failed to pull image "registry.redhat.io/openshift4/ose-kube-rbac-proxy-rhel9:v4.17.0-202503121206.p0.g7718265.assembly.stream.el9": failed to pull and unpack image "registry.redhat.io/openshift4/ose-kube-rbac-proxy-rhel9:v4.17.0-202503121206.p0.g7718265.assembly.stream.el9": failed to resolve reference "registry.redhat.io/openshift4/ose-kube-rbac-proxy-rhel9:v4.17.0-202503121206.p0.g7718265.assembly.stream.el9": failed to authorize: failed to fetch anonymous token: unexpected status from GET request to https://registry.redhat.io/auth/realms/rhcc/protocol/redhat-docker-v2/auth?scope=repository%3Aopenshift4%2Fose-kube-rbac-proxy-rhel9%3Apull&service=docker-registry: 401 Unauthorized
```

2. If you experience this error:
 - a. Create a secret, `rh-kube-proxy-secret`, in the `kube-system` namespace in which the operator is being installed:

```
kubectl create secret docker-registry rh-kube-proxy-secret --docker-username="<your userid on Redhat portal>" --docker-password="<your redhat portal password>" --docker-server="registry.redhat.io" -n <namespace of operator deploy>
```

- b. Verify the secret has been created successfully:

```
kubectl get secret rh-kube-proxy-secret -n <namespace>
```

c. Once the secret is created, delete the pod that is showing `ImagePullErr`: `kubectl delete pod -n cte-k8s-operator-controller-manager-c6997c9fd-lc9mq`

This will cause the pod to restart, pick up the secret that you just created, and pull the image successfully.

Note

The previous steps do not have any effect on the CTE-K8s pods that have already been deployed by an earlier version of the Operator.

Standard Kubernetes Environment

- In a standard Kubernetes environment, create the secret using the following command:

```
kubectl create secret docker-registry rh-kube-proxy-secret \
--docker-server=registry.redhat.io \
--docker-username=<your_redhat_username> \
--docker-password=<your_redhat_password> \
--docker-email=<your_email> \
-n <your_namespace>
```

Installing the CTE for Kubernetes Operator

Warning

- Do not uninstall CTE for Kubernetes and the operator from the CLI, if they were installed using the GUI.
- Similarly, do not uninstall CTE for Kubernetes and the operator from the GUI, if it was installed using the CLI.

Note

You can install CTE for Kubernetes and the Operator **only** in the OpenShift-operators namespace. If you want to install in other namespaces, use the CLI option.

The CTE for Kubernetes Operator can be installed using one of two methods:

- CLI
- Cluster Console GUI

CLI Method Prerequisites

1. Install the [Operator Lifecycle Manager \(OLM\)](#) on the cluster. Refer to [Installing OLM](#) for instructions on how to install OLM on the cluster.

Note

The latest version of OpenShift is installed by default.

2. Install `oc` (OpenShift CLI command for OpenShift cluster) on the cluster.

Note

The user installing the Operator must have Cluster Admin permissions.

3. Download the [Deploy Scripts](#).
4. Execute the `deploy.sh` script from the deploy directory, type:

```
./deploy.sh --operator --operator-ns=<namespace-in-which-to-deploy-the-operator> --cte-ns=<namespace-in-which-to-deploy-cte-4-k8s>
```

Example

```
./deploy.sh --operator-ns=my-ns1 --cte-ns=my-ns2
```

If either of the namespace options is not specified, the script sets `kube-system` as the default namespace for deployment.

Installing a specific image

1. Change to the CTE for Kubernetes directory, type:

```
cd ciphertrust-transparent-encryption-kubernetes
```

2. Deploy the specific image:

```
./deploy.sh -t 1.6.0.31
```

Namespace Deployments

Ensure that the namespace passed to the deployment script exists before initiating deployment. This prevents the script from prompting for creation of namespace during deployment. For example, if the deployment script is invoked as:

```
./deploy.sh --operator --operator-ns=my-ns1 --cte-ns=my-ns2
```

where both namespaces `my-ns1` and `my-ns2`, do not exist, the script would prompt with the following response:

```
Starting the cte-csi containers
NAMESPACE my-ns1 not found!!!!!!
Namespace my-ns1 not found. Do you want to create it now [N/y]
```

Once the namespace information is available, the script proceeds to create the objects required and installs the operator. After installing the operator, the script deploys `ctek8soperator` CRD. This deploys CTE for Kubernetes on the cluster in the namespace specified.

Using the Cluster Console Web GUI

The CTE for Kubernetes Operator is certified with Red Hat for the OpenShift platform. It is integrated with OperatorHub. The operator can be discovered on the OperatorHub page.

1. Open a browser and navigate to the **Operators > OperatorHub** link, in the left navigation panel on the console GUI. Type `CTE` in the search field under **All Items** to find the CTE for Kubernetes Operator.
2. Click on the tile to access the install option.
3. Ensure that all prerequisites are met before installing the operator.
4. Click **Install** to install the operator. Do not change the default values on the install page.

Installing and Managing CTE for Kubernetes in an Operator Environment

Installing CTE for Kubernetes

To install CTE for Kubernetes (when CTE for Kubernetes Operator is already installed):

1. Navigate to the deploy directory that was downloaded for operator install.
2. Ensure the file `ctek8soperator-crd.yaml` has correct values.
3. Type:

```
oc apply -f <path to>/ctek8soperator-crd.yaml
```

Updating CTE for Kubernetes

To update CTE for Kubernetes:

1. Stop any application that is using CTE for Kubernetes volumes.
2. Navigate to the deploy directory that was downloaded for operator install.
3. Edit the `ctek8soperator-crd.yaml` file.
4. Update the `version` field in the spec section to the latest version of [CTE for Kubernetes](#).
5. Apply the change with the command:

```
oc apply -f <path to>/ctek8soperator-crd.yaml
```

Using user-defined Configuration Parameters with RedHat OpenShift

You can use user-defined configuration parameters for CTE-K8s daemonset so user can control on which nodes CTE-K8s is deployed as an operator

cte-csi-node manifest

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  annotations:
    deprecated.daemonset.template.generation: "1"
  creationTimestamp: "2024-06-20T10:08:55Z"
  generation: 1
  name: cte-csi-node
  namespace: cte-o
  ownerReferences:
  - apiVersion: cte-k8s-operator.csi.cte.cpl.thalesgroup.com/v1
    blockOwnerDeletion: true
    controller: true
    kind: CteK8sOperator
    name: ctek8soperator
    uid: 292a9b6d-716d-4b35-b965-916ae58b709c
  resourceVersion: "678877896"
  uid: aa2160c0-030f-49e6-b02f-0249cb78d033
spec:
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app.kubernetes.io/component: CteK8sOperator-Deployment
      app.kubernetes.io/instance: ctek8soperator
```

```

app.kubernetes.io/managed-by: CteK8sOperator-Operator
app.kubernetes.io/name: CteK8sOperator
app.kubernetes.io/version: 1.2.0-latest
template:
  metadata:
    creationTimestamp: null
  labels:
    app.kubernetes.io/component: CteK8sOperator-Deployment
    app.kubernetes.io/instance: ctek8soperator
    app.kubernetes.io/managed-by: CteK8sOperator-Operator
    app.kubernetes.io/name: CteK8sOperator
    app.kubernetes.io/version: 1.2.0-latest
  spec:
    affinity:
      nodeAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
          nodeSelectorTerms:
            - matchExpressions:
                - key: type
                  operator: NotIn
                  values:
                    - virtual-kubelet
            - key: kubernetes.io/arch
              operator: In
              values:
                - amd64
                - arm64
    containers:
      - args:
          - --endpoint=$(CSI_ENDPOINT)
          - --nodeid=$(KUBE_NODE_NAME)
          - --namespace=$(KUBE_NAMESPACE)
          - --v=5
          - --apiburst=300
          - --apiqps=200
          - --registration-cleanup-interval=10
          - --pauseimage=registry.k8s.io/pause@sha256:7031c1b283388d2c2e09b57badb80
            3c05ebed362dc88d84b480cc47f72a21097
        env:
          - name: CSI_ENDPOINT
            value: unix:///csi/csi.sock
          - name: KUBE_NODE_NAME
            valueFrom:
              fieldRef:
                apiVersion: v1
                fieldPath: spec.nodeName
          - name: KUBE_NAMESPACE
            valueFrom:
              fieldRef:
                apiVersion: v1
                fieldPath: metadata.namespace
        image: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kub
          ernetes:1.2.0-latest
        imagePullPolicy: Always
        lifecycle:
          preStop:
            exec:
              command:
                - /bin/sh

```

```

        - -c
        - rm -rf /csi/csi.sock
name: cte-csi
resources: {}
securityContext:
capabilities:
  add:
    - SYS_ADMIN
privileged: true
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
  - mountPath: /csi
name: plugin-dir
  - mountPath: /var/lib/kubelet/pods
mountPropagation: Bidirectional
name: mountpoint-dir
  - mountPath: /etc/kubernetes/
name: kube-cred
  - mountPath: /dev/shm
name: dshm
  - mountPath: /var/log
name: varlog
  - mountPath: /chroot
name: chroot
  - mountPath: /var/run/cri.sock
name: cri-sock
- args:
  - --agentlogs
  image: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kub
ernetes:1.2.0-latest
  imagePullPolicy: IfNotPresent
  name: cte-agent-logs
  resources: {}
  securityContext:
  privileged: true
  terminationMessagePath: /dev/termination-log
  terminationMessagePolicy: File
  volumeMounts:
  - mountPath: /var/log
  name: varlog
  - mountPath: /chroot
  name: chroot
- args:
  - --csi-address=/csi/csi.sock
  - --kubelet-registration-path=/var/lib/kubelet/plugins/csi.cte.cpl.thales
group.com/csi.sock
  - --v=5
  image: k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.0.1
  imagePullPolicy: IfNotPresent
  name: csi-sidecar-registrar
  resources: {}
  securityContext:
  privileged: true
  terminationMessagePath: /dev/termination-log
  terminationMessagePolicy: File
  volumeMounts:
  - mountPath: /csi
  name: plugin-dir

```

```

    - mountPath: /registration
    name: registration-dir
- args:
  - --csi-address=/csi/csi.sock
  - --v=5
  image: k8s.gcr.io/sig-storage/csi-attacher:v3.0.0
  imagePullPolicy: IfNotPresent
  name: csi-sidecar-attacher
  resources: {}
  securityContext:
    privileged: true
  terminationMessagePath: /dev/termination-log
  terminationMessagePolicy: File
  volumeMounts:
    - mountPath: /csi
      name: plugin-dir
dnsPolicy: ClusterFirst
hostPID: true
imagePullSecrets:
- name: cte-csi-secret
nodeSelector:
  kubernetes.io/os: linux
priorityClassName: system-node-critical
restartPolicy: Always
schedulerName: default-scheduler
securityContext: {}
serviceAccount: cte-csi-node
serviceAccountName: cte-csi-node
terminationGracePeriodSeconds: 600
tolerations:
- operator: Exists
volumes:
- hostPath:
    path: /var/lib/kubelet/plugins_registry/
    type: Directory
    name: registration-dir
- hostPath:
    path: /var/lib/kubelet/plugins/csi.cte.cpl.thalesgroup.com/
    type: DirectoryOrCreate
    name: plugin-dir
- hostPath:
    path: /var/lib/kubelet/pods
    type: DirectoryOrCreate
    name: mountpoint-dir
- hostPath:
    path: /etc/kubernetes/
    type: Directory
    name: kube-cred
- emptyDir:
    medium: Memory
    name: dshm
- emptyDir: {}
    name: varlog
- emptyDir: {}
    name: chroot
- hostPath:
    path: /run/crio/crio.sock
    type: ""
    name: cri-sock

```



```

updateStrategy:
  rollingUpdate:
    maxSurge: 0
    maxUnavailable: 1
    type: RollingUpdate
status:
  currentNumberScheduled: 15
  desiredNumberScheduled: 15
  numberAvailable: 12
  numberMisscheduled: 0
  numberReady: 12
  numberUnavailable: 3
  observedGeneration: 1
  updatedNumberScheduled: 15

```

Using ConfigMaps to customize deployment of CTE for Kubernetes

If you are deploying CTE for Kubernetes using Helm, you can customize this deployment by modifying parameters in *values.yaml* file or manifest files (*cte-csi-controller.yaml*, *cte-csi-driver*, and *cte-csi-nodeserver.yaml*) directly.

However, if you are deploying CTE for Kubernetes using operator, you can customize the deployment by modifying only small subset of parameters that are available in the *ctek8soperator-crd.yaml* CRD. The operator uses these values to update the values in the bundled CTE for Kubernetes manifest files before deploying it. To modify other parameters (such as, parameters in the manifest files), you require a new build of the operator, which supports that parameter in its CRD. This is because the manifest files required for deploying CTE for Kubernetes are bundled into an operator image at the time operator is built.

From version 1.5.0 and subsequent versions, this limitation has been addressed. Now, you can either use the CRD to customize small set of parameters, or modify the manifest files and upload them as a ConfigMap. The Kubernetes cluster administrator can customize the deployment by modifying the manifest files (*cte-csi-controller.yaml* and *cte-csi-nodeserver.yaml*) and uploading them as a ConfigMap.

Note

If ConfigMap is available then changes in *ctek8soperator-crd.yaml* CRD are ignored.

Once the operator finds the ConfigMap, it extracts the manifests from this ConfigMap and uses them to deploy CTE for Kubernetes. If the ConfigMap is not found or the operator is not able to successfully extract the manifests, then it falls back to the manifests packaged at the time of build.

To create a custom ConfigMap follow the steps below.

1. On the cluster where you want to deploy CTE for Kubernetes, clone the `ciphertrust-transparent-encryption-kubernetes.git` repository and navigate to this directory.

```

git clone https://github.com/thalescpl-io/ciphertrust-transparent-encryption-kubernetes.git

```

```
cd ciphertrust-transparent-encryption-kubernetes
```

2. Execute the `detemplatize_manifest.sh` command. To execute this command Helm must be installed.

```
./detemplatize_manifest.sh
```

This script creates a `custom_manifests/<CTE-K8s version>` directory tree.

3. Edit the manifest files in the `custom_manifests` directory corresponding to the version of CTE for Kubernetes being deployed. For instance, if you are deploying version 1.6.0 (1.6.0-latest), then edit the files in the `custom_manifests/1.6.0` directory.
4. Execute the `kubectl` command to create the ConfigMap.

Note

It is mandatory to name the ConfigMap as `ctecustomconfig` and create it in the same namespace where the CTE for Kubernetes operator is being deployed.

Example: To create `ctecustomconfig` in the `kube-system` namespace for deployment of version 1.6.0, execute the below command:

```
kubectl create configmap ctecustomconfig -n kube-system --from-file=./custom_manifests/1.6.0/
```

5. Verify that the manifest was created.

```
kubectl get configmap ctecustomconfig -n kube-system
```

After completing above steps, deploy the operator and CTE for Kubernetes using either GUI or the deployment scripts.

Upgrading CTE for Kubernetes

1. Check the [Docker site](#) for a new version of CTE for Kubernetes:
2. Stop any application that is using CTE for Kubernetes volumes.
3. Expand **Operators** section and click **Installed Operators** page from the left hand bar on the page.
4. Click on the Name of the operator installed.

5. On the next page, select the **YAML** tab. This displays the manifest of CTE for Kubernetes instances installed on the cluster.
6. Update the version parameter in the CRD spec with the updated CTE-K8s version.
7. Click **Save**.

The upgrade process terminates all of the CTE for Kubernetes pods running the previous version, while activating new pods running the new version. Once the `cte-csi-node-XXXX` pods are running on all of the nodes of the cluster, the application that uses the CTE for Kubernetes volume can be re-deployed.

8. If any of the other parameters in `CTE-K8s-Operator.yaml`, like `logLevel`, `apiburst`, `apqps`, etc. need to be updated, follow the same process as above to update the relevant parameter.

Deleting CTE for Kubernetes

To delete CTE for Kubernetes:

1. Stop any application that is using CTE for Kubernetes volumes.
2. Navigate to the deploy directory that was downloaded for operator install.
3. Type:

```
oc delete -f <path to>/ctek8soperator-crd.yaml
```

Kubernetes Operator Maintenance

Upgrading CTE for Kubernetes Operator

The operator is designed to upgrade to a newer version automatically as soon as a new version is published by Thales. The Operator Lifecycle Manager, or OLM, constantly polls for updates and upgrades the operator whenever an update is available.

Uninstalling CTE for Kubernetes Operator

To uninstall CTE for Kubernetes Operator from the User Interface:

1. Stop any application that is using CTE for Kubernetes volumes.
2. Expand “Operators” section and click **Installed Operators** page from the left hand bar on the page.
3. Click on the Name of the operator installed.
4. On the next page, select the **Actions** tab.
5. Click **Uninstall Operator** in the list of actions displayed.

6. Click **Uninstall** on the confirmation pop-up to uninstall CTE for Kubernetes.

To uninstall the CTE for Kubernetes Operator using the CLI:

1. Stop any application that is using CTE for Kubernetes volumes.
2. Navigate to the deploy directory that was downloaded for operator install, type:

```
./deploy.sh --operator --operator-ns=<namespace-in-which-ctek8soperator-is-dep
loyed> --cte-ns=<namespace-in-which-ctek8s-is-deployed> --remove
```

This removes both CTE for Kubernetes and the operator.

Using Helm with CTE for Kubernetes

- Installing and Deploying with Helm
- Deploying in an Air-Gapped Kubernetes Cluster Environment using HELM
- Migrating a CTE install from Helm to Operator deployment
- Upgrading CTE for Kubernetes with a Helm-based Installation

Installing and Deploying with Helm

How to Deploy pods/services in Kubernetes Cluster

The CTE for Kubernetes images are distributed through the [Thales Docker Hub](#).

- All of the pods/services are deployed using `yaml` files. The `yaml` files are executed using the provided `deploy.sh` scripts.
- CTE for Kubernetes **only** supports homogeneous Kubernetes clusters, i.e. all nodes of the Kubernetes clusters must be x86_64 or all nodes must be arm64.

Options for Deploy Scripts

Deploy all of the pods by using the following command and needed arguments from the following table:

Option	Function	Description
-r	remove	Remove all the running pods, services and secrets.
-t	tag	Tag of the image on the server. Default is the latest.
	--helm	Deploy CTE for Kubernetes using Helm.

Option	Function	Description
	--operator-ns=	The namespace in which to deploy the Operator.
	--cte-ns=	The namespace in which to deploy the CSI driver.
	--cri-sock=	Container Runtime Interface socket path.

For CTE for Kubernetes v1.3.0 and subsequent versions, the deployment script uses the repository image index as the image name, instead of the individual platform image names. The manifest (`cte_csi : <tag>`) points to an image.

Note

The **default** image name is: `cte_csi`. You do not need to specify the image name if you use the default name.

Install Helm

Helm is a package manager for Kubernetes.

1. Download Helm CLI, type:

```
curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 \ |
bash
```

2. Check for the version, type:

```
helm version --client
```

3. Validate the connection, type:

```
helm ls --all-namespaces
```

Installing using Helm

1. Change to the CTE for Kubernetes directory, type:

```
cd ciphertrust-transparent-encryption-kubernetes
```

2. Deploy with Helm:

```
./deploy.sh --helm
```

Terminating Pods

To terminate all of the pods and delete all of the services and secrets:

1. Change to the CTE for Kubernetes directory, type:

```
cd ciphertrust-transparent-encryption-kubernetes
```

2. For Helm deployment, type:

```
./deploy.sh --remove --helm
```

Verify CTE for Kubernetes

Verify that CTE for Kubernetes is running, type:

```
kubect1 get pods --namespace=kube-system -o wide | grep cte-csi
```

RESPONSE

NAME	READY	ST
ATUS kube-system cte-csi-controller-5db88d6cb-tn6lr	4/4	Ru
nning 0 6m59s 10.244.1.5 ubuntu20-02-kubcluster-worker		
<none> <none>		
kube-system cte-csi-node-lz7t9	4/4	Ru
nning 0 6m59s 10.244.0.26 ubuntu20-02-kubcluster-master		
<none> <none>		
kube-system cte-csi-node-pzvwb	4/4	Ru
nning 0 6m59s 10.244.2.2 ubuntu20-02-kubcluster-worker2		
<none> <none>		
kube-system cte-csi-node-wmhl	4/4	Ru
nning 0 6m59s 10.244.1.4 ubuntu20-02-kubcluster-worker		
<none> <none>		

Helm Charts Deployment

Helm offers a straightforward method for deploying CTE for Kubernetes. However, it places the responsibility on the user to manage node upgrades. Before upgrading, users must ensure that no applications on the node are actively using a CTE Persistent Volume Claim (PVC). Manual draining of the node is required to complete the upgrade process.

Specify the desired installation method using command-line arguments in the `deploy.sh` script. When using the Operator method, you must install the Operator Lifecycle Manager (OLM) prior to deployment.

- See [Deploying CTE for Kubernetes in an Air-Gapped Kubernetes Cluster Environment using HELM](#)

Migrating Helm to Operator

You can migrate Helm to Operator if you choose to move from manual to automatic management:

- [Migrating a CTE for Kubernetes install from Helm to Operator deployment](#)

Upgrading Helm

Helm must be upgraded manually by the administrator:

- [Upgrading CTE for Kubernetes with HELM](#)

Migrating a CTE install from Helm to Operator deployment

For CTE for Kubernetes v1.5.0 and subsequent versions, CTE for Kubernetes Operator is automatically installed by default.

Prior to v1.5.0, users had to manually migrate from a Helm-based deployment to Operator-based deployment. This is an optional feature for prior versions.

Upgrading to v1.5.0 from a prior version

If you are using a version prior to v1.5.0, for example, CTE for Kubernetes v1.3.0 deployed using Helm, then if you attempt to install CTE for Kubernetes v1.5.0, CTE for Kubernetes displays an error message stating that the existing deployment was completed using Helm. For that scenario, you have the following two options:

- Pass the `--helm` parameter to the `deploy.sh` script to upgrade to CTE for Kubernetes v1.5.0 using Helm charts. See [\[Upgrading\]](#)
- Migrate to operator based deployment using the following section:

Manually migrating to CTE for Kubernetes Operator

To switch to a Kubernetes operator deployment, you can manually migrate from Helm to Operator so that you can upgrade to CTE for Kubernetes v1.5.0 and subsequent versions:

1. Stop all of your applications that use CTE Volumes. Ensure that you stop all application pods using CTE-K8s and those actively accessing CTE volumes.
2. Revert the CTE for Kubernetes deployment, type:

For CTE for Kubernetes v1.5 and subsequent versions

```
./deploy.sh --helm --remove
```

For CTE for Kubernetes v1.4 and previous versions

```
./deploy.sh --remove
```

3. Deploy Kubernetes operator, type:

```
./deploy.sh --operator -t <image tag>
```

4. Redeploy the application pods that use CTE volumes.

Deploying in an Air-Gapped Kubernetes Cluster using HELM

Prerequisites

On a system that has access to the internet and to an internal container registry:

1. **Install skopeo.** Skopeo is a command-line utility (CLI) used to interact with local and remote container images and container image registries.
2. **Install jq.** JQ is a lightweight and flexible command-line JSON processor.
3. If the internal registry is private, and requires authorization to add/list images, then, **before** executing the script, create an authorization file to allow **skopeo** access to the internal registry:

```
skopeo login <url-for-internal-registry[:port number]> --authfile ./docker_auth -u <username>
```

Example

```
skopeo login <k8s.gcr.io[:22]> --authfile ./docker_auth -u penderyn
```

Note

The name of the authfile **must** be `docker_auth`. If this is changed, then the name must be updated in the following script.

4. Enter the password for the user when prompted.
5. Copy and paste the following into a script and execute the script:

```
#!/bin/bash

DOCKER_AUTH_JSON=./docker_auth
SRC_REGISTRY="docker.io/thalesciphertrust"
SRC_CTEK8S_NAME="${SRC_REGISTRY}/ciphertrust-transparent-encryption-kubernetes"
DEST_REGISTRY=<path to your registry such as example.com[:port]/my-internal-re
```



```

gistry>

skopeo login ${DEST_REGISTRY} --authfile ${DOCKER_AUTH_JSON} -u <username>

DEST_REGISTRY=`echo ${ DEST_REGISTRY} | sed -e "s/\//g"`
for VER in `skopeo list-tags docker://${SRC_CTEK8S_NAME} | jq '.Tags' | grep
-vE "[\\|]" | sed -e s/[\\",,]/g`
do

skopeo copy --all docker://${SRC_CTEK8S_NAME}:${VER} \
                docker://${DEST_CTEK8S_NAME}:${VER} --authfil
e ${DOCKER_AUTH_JSON}
done

skopeo copy --all docker://k8s.gcr.io/sig-storage/csi-node-driver-registrar:v
2.0.1 \
                docker://${DEST_REGISTRY}/csi-node-driver-registrar:v2.0.1 --authf
ile ${DOCKER_AUTH_JSON}

skopeo copy --all docker://k8s.gcr.io/sig-storage/csi-attacher:v3.3.0 \
                docker://${DEST_REGISTRY}/csi-attacher:v3.3.0 --authfile ${DOCKE
R_AUTH_JSON}

skopeo copy --all docker://k8s.gcr.io/sig-storage/csi-provisioner:v4.0.0 \
                docker://${DEST_REGISTRY}/csi-provisioner:v4.0.0 --authfile ${DOCK
ER_AUTH_JSON}

skopeo copy --all docker://k8s.gcr.io/pause:3.9 \
                docker://${DEST_REGISTRY}/pause:3.9 --authfile ${DOCKER_AUTH_JSON}

```

Deploying in an Air-Gapped Kubernetes Cluster

1. Download CTE for Kubernetes deployment files, type:

```
git clone https://github.com/thalescpl-io/ciphertrust-transparent-encryption-k
ubernetes.git
```

2. Change to the CTE for Kubernetes directory, type:

```
cd ciphertrust-transparent-encryption-kubernetes
```

3. Edit the script variables to reflect your setup, type:

```
vi deploy/kubernetes/1.6.0/values.yaml
```

a. Replace `docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernetes` with appropriate values for your environment: `<example.com/my-internal-registry>/ciphertrust-transparent-encryption-kubernetes`

b. Replace `registry.k8s.io/pause:3.9` with appropriate values for your environment: `<example.com/my-internal-registry>/pause:3.9`

c. Replace `registry.k8s.io/sig-storage/csi-provisioner:v4.0.0` with appropriate values for your environment: `<example.com/my-internal-registry>/csi-provisioner:v4.0.0`

d. Replace `k8s.gcr.io/sig-storage/csi-attacher:v3.3.0` with appropriate values for your environment: `<example.com/my-internal-registry>/csi-attacher:v3.3.0`

e. Replace `k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.0.1` with appropriate values for your environment: `<example.com/my-internal-registry>/csi-node-driver-registrar:v2.0.1`

4. Deploy the script, type:

```
./deploy.sh -t 1.6.0-latest --helm
```

Upgrading CTE for Kubernetes with a Helm-based Installation

Introduction

Upgrading CTE for Kubernetes to the latest release is accomplished by invoking the `deploy.sh` script with the desired release tag. The CTE for Kubernetes node server cannot be automatically upgraded. Doing this will result in application failures for existing pods using CTE volumes on the nodes. You must use a manual procedure to stop the pods running on the nodes and delete the CTE for Kubernetes node server pods on that node.

Redeploy CTE for Kubernetes

To redeploy CTE for Kubernetes, type:

```
./deploy.sh -t <version#>.<build#>
```

For example:

```
./deploy.sh -t 1.6.0.31
```

Drain and update a node

To upgrade the nodes:

1. Get a list of nodes currently running CTE for Kubernetes, type:

```
kubectl get pods -n kube-system -o wide -l app=cte-csi-node --no-headers -o custom-columns=":spec.nodeName"
```

Response Example:

```
control.example.com  
node1.example.com
```

2. Drain a node from the list, type:

```
kubectl drain --delete-emptydir-data --ignore-daemonsets <nodeName>
```

Response example:

```
node/node1.example.com cordoned  
node/node1.example.com drained
```

Note

The drain command pauses while it waits for the node to be drained.

3. Use a compound kubectl command to delete the CTE for Kubernetes node server on this node:

```
kubectl delete -n kube-system pod $(kubectl get pods -n kube-system --field-selector spec.nodeName=<nodeName> -l app=cte-csi-node --no-headers -o custom-columns=":metadata.name")
```

- The `kubectl get pods` needs to be bounded to only output the CTE for Kubernetes-node label for this node
- Use `--field-selector spec.nodeName=<nodeName>` to specify the CTE for Kubernetes node to drain.
- Use `-l app=cte-csi-node` to select only the CTE for Kubernetes node server from that node.
- Use `--no-headers -o custom-columns=":metadata.name"` to tell kubectl to only output the pod names.
- Pass the output of that command to the `kubectl delete pod` which triggers a new CTE for Kubernetes

node server pod to be created with the desired image.

4. Verify that the pod has started successfully, type:

```
kubectl get pods -n kube-system --field-selector spec.nodeName=<nodeName> -l app=cte-csi-node -o wide
```

Response example:

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
NOMINATED NODE	READINESS GATES					
cte-csi-node-kgwkj	4/4	Running	0	14m	192.168.77.163	node1.example.com
	<none>	<none>				

5. Return node to working status, type:

```
kubectl uncordon <nodeName>
```

Response example:

```
node/node1.example.com uncordoned
```

Deployment Methods

Deployment Methods for Clusters Connected to the Internet

CTE for Kubernetes supports two deployment methods: Kubernetes Operator and Helm Charts. Beginning with version 1.5.0, the Operator is the default deployment method.

Operator Deployment

The Operator-based installation deploys a dedicated operator pod that monitors the usage of CTE PVCs across the cluster. It automates the upgrade of CTE for Kubernetes by detecting when nodes are no longer using CTE and proceeding with the upgrade process accordingly.

- See [Using the CTE for Kubernetes Operator Resource](#) for more information

Once the operator is installed, the Kubernetes API servers becomes aware of the Kubernetes customer resource (CR).

- See [Deploying with a Custom Resource Definition](#) for more information

Helm Deployment

Helm is a package manager for Kubernetes.

- Deploy a CTE for Kubernetes Volume with [HELM](#).

Deployment Methods for Clusters NOT Connected to the Internet

Operator Deployment

- Deploying CTE for Kubernetes in an Air-Gapped OpenShift Container Platform Cluster using CTE for Kubernetes Operator

Helm Deployment

- Deploying CTE for Kubernetes in an Air-Gapped Kubernetes Cluster Environment using HELM

Deploying with a Custom Resource Definition

Once the operator is installed, the Kubernetes API servers becomes aware of the Kubernetes customer resource (CR). The installation process registers the:

- CR
- API for managing the CR
- Controller that handles the requests, for the CR, from the API Server

To instantiate the CR:

1. Click **View Operator** on the page displayed immediately after the operator is installed.
2. Alternatively:
 - a. Expand “Operators” section and click **Installed Operators** page from the left hand bar on the page.
 - b. Click on the Name of the operator installed.
 - c. Click **Create Instance** link on the page displayed.
 - d. Click **Create** to deploy CTE for Kubernetes.

Deploying with a Custom Resource Definition

You can deploy CTE for Kubernetes using a Custom Resource Definition (CRD). The following displays a sample CRD used to create an instance of the CTE for Kubernetes:

CTE-K8S-Operator-crd.yaml

```
apiVersion: cte-k8s-operator.csi.cte.cpl.thalesgroup.com/v1
kind: CteK8sOperator
metadata:
  labels:
```

```

    app.kubernetes.io/name: ctek8soperator
    app.kubernetes.io/instance: ctek8soperator
    app.kubernetes.io/part-of: cte-k8s-operator
    app.kubernetes.io/managed-by: kustomize
    app.kubernetes.io/created-by: cte-k8s-operator
  name: ctek8soperator
  spec:
    replicas: 1
    image: "docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernetes"
    version: "1.2.0-latest"
    imagePullPolicy: Always
    logLevel: 5
    apiBurst: 300
    apiQPS: 200
    imagePullSecrets:
      - name: cte-csi-secret
    registrationCleanupInterval: 10
    pauseImage: "k8s.gcr.io/pause:latest"
    volumes:
      - name: cri-sock
        hostPath:
          path: "/run/crio/crio.sock"
    # The following parameters are optional. If values are not specified, CT
    E-K8s uses the default values
    csiProvisionerImage: registry.k8s.io/sig-storage/csi-provisioner:v4.0.0
    csiNodeDriverRegistrarImage: k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.0.1
    csiAttacherImage: k8s.gcr.io/sig-storage/csi-attacher:v3.3.0
    csiSnapshotterImage: registry.k8s.io/sig-storage/csi-snapshotter:v6.3.3
    snapImagePullPolicy: IfNotPresent

```

Applying a CRD

To apply the CRD, type:

```
kubectl apply -f <path to\cte-k8s-operator.yaml>
```

Deploying CTE for Kubernetes in an Air-Gapped OpenShift Container Platform Cluster using CTE for Kubernetes Operator

Overview

This section documents the mandatory steps to deploy CTE for Kubernetes Operator and CTE for Kubernetes on an OCP cluster that is not connected to the internet. Once your OCP cluster is setup and configured properly, you can then mirror the CTE for Kubernetes Operator Catalog, CTE for Kubernetes images, and other images on which CTE for Kubernetes is dependent.

Note

The following information is valid for CTE for Kubernetes Operator v1.5.9 and subsequent versions. It is **not valid** with any previous version.

Prerequisites

You must have a docker-v2 compliant image registry that is set up properly, running, and is accessible from the internal network used by your OCP cluster. The illustrative commands in this document use the registry setup internally at `ocp-discon.lab.ocp.lan:5000`. This registry is accessible on the OCP private network.

Assumptions for this Document

The Bastion/Service/Helper node of the OCP cluster has access to the internet and the OCP cluster.

If this is not true, then you must manually mirror the CTE for Kubernetes Operator Catalog onto a system that has access to internet. The catalog may include other related images in the form of a file archive from docker.io (or their respective locations). This archive must then be copied to the Bastion/Service/Helper node.

Mirror the CTE for Kubernetes Catalog

"

1. Configure `imageset-configuration-ctek8soperator.yaml`. Refer to [Appendix: Sample Script](#) for a sample `ImageSetConfiguration` file for CTE for Kubernetes Operator and related images.
2. Mirror the CTE for Kubernetes Operator Catalog, and other related images, to a file on a disk.
3. Run the `oc-mirror` command, type:

```
oc-mirror --config imageset-configs/imageset-config-ctek8soperator.yaml file:///root/ocp-mirror/ocp-release/ctek8soperator-images 2>&1 | tee ctek8soperator-mirror.log
```

Expected output:

```
---> info: Mirroring completed in 3m58.31s (11.52MB/s)
---> Creating archive /root/ocp-mirror/ocp-release/ctek8soperator-images/mirror_seq1_000000.tar
```

4. Disable internet access

Note

Not applicable if creating the mirror archive through another server. If so, manually copy the archive.

5. Upload the images, that were just mirrored to the file archive, to your internal docker-v2 compliant registry:

```
oc-mirror --from=/root/ocp-mirror/ocp-release/ctek8soperator-images/mirror_seq
1_000000.tar docker://ocp-discon.lab.ocp.lan:5000
```

Expected output:

```
---> Rendering catalog image "ocp-discon.lab.ocp.lan:5000/shetland/ciphertrus
t-transparent-encryption-kubernetes-operator-catalog:v1.2.5" with file-based c
atalog
---> Writing image mapping to oc-mirror-workspace/results-1737970204/mapping.t
xt
---> Writing CatalogSource manifests to oc-mirror-workspace/results-1737970204
---> Writing ICSP manifests to oc-mirror-workspace/results-1737970204
```

Edit the deploy scripts -- Mandatory

1. Edit the file `oc-mirror-workspace/results-1737970204/imageContentSourcePolicy.yaml` and change the `Spec.Metadata.Name`:

```
metadata:
  name: cte-operator-generic-0
  :
  :
metadata:
  labels:
    operators.openshift.org/catalog: "true"
  name: cte-operator-0
```

2. Apply `imageContentSourcePolicy.yaml`.

```
cat oc-mirror-workspace/results-1737970204/imageContentSourcePolicy.yaml
```

Response:


```
imagecontentsourcepolicy.operator.openshift.io/cte-operator-generic-0 created
imagecontentsourcepolicy.operator.openshift.io/cte-operator-0 created
```

3. Open the following file for editing:

```
oc-mirror-workspace/results-1737970204/catalogSource-cs-ciphertrust-transparent-encryption-kubernetes-operator-catal.yaml
```

4. Append the following 5 lines to the file:

```
displayName: CTE-K8s Operator Catalog
publisher: Thales Group
updateStrategy:
  registryPoll:
    interval: 24h
```

Response:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ctek8soperator-catalog
  namespace: openshift-marketplace
spec:
  image: ocp-discon.lab.ocp.lan:5000/public-repos/ciphertrust-transparent-encryption-kubernetes-operator-catalog:v1.2.5
  sourceType: grpc
  displayName: CTE-K8s Operator Catalog
  publisher: Thales Group
  updateStrategy:
    registryPoll:
      interval: 24h
```

5. Apply the CatalogSource manifest:

```
oc apply -f oc-mirror-workspace/results-1741799472/catalogSource-cs-ciphertrust-transparent-encryption-kubernetes-operator-catal.yaml
```

Response:

```
catalogsource.operators.coreos.com/ctek8soperator-catalog created
```

6. Verify the catalog resource:

```
oc get catsrc -n openshift-marketplace
```

Response:

NAME	DISPLAY	TYPE	PUBLISHER	AGE
ctek8soperator-catalog	CTE-K8s Operator Catalog	grpc	Thales Group	5s

7. Verify the pods:

```
oc get pods -n openshift-marketplace
```

Response:

NAME	READY	STATUS	RESTARTS	AGE
ctek8soperator-catalog-9wv58	1/1	Running	0	27s

Download and configure deploy scripts -- Mandatory

Before Deployment, you must follow the steps in this section. It is mandatory.

1. Clone the deploy scripts from <https://github.com/thalescpl-io/ciphertrust-transparent-encryption-kubernetes.git>
2. Edit this file:

```
vi deploy/kubernetes/x.y.z/values.yaml
```

3. Replace the registry name for each of the images with the corresponding URL in the file `oc-mirror-workspace/results-nnnnnnnn/mapping.txt`.
4. Execute the `detemplatize_manifest.sh` script to create a set of files for use in creating a configMap.
Type:

```
./detemplatize_manifest.sh
```

This script:

- Picks up manifest files from `deploy/kubernetes/*/templates`
- Uses the `values.yaml` file from the respective chart version to get values for fields templated in the manifests.

- Creates a set of detemplatized manifest files for creating a custom ConfigMap for CTE-K8s deployment

In case further changes are required, beyond what can be customized via `values.yaml` (`logLevel`, `Image URL`, `APIServerThrottle.burst`, `APIServerThrottle.qps`), like change tolerations, change nodeAffinity, for controller pod or nodeserver pod, do it now directly on the custom manifest files generated.

5. Create a ConfigMap with the customized manifests, type:

```
oc create configmap ctecustomconfig -n <operator namespace> --from-file=./custom_manifests/X.Y.Z/
```

6. Verify that the ConfigMap was created, type:

```
oc get configmap ctecustomconfig -n <operator namespace>
```

7. Edit this file:

```
vi deploy/kubernetes/x.y.z/operator-deploy/ctek8soperator-subscription.yaml
```

Change the `Spec.source` and `Spec.sourceNamespace` to match the values in `oc-mirror-workspace/results-1741799472/catalogSources-ciphertrust-transparent-encryption-kubernetes-operatorcatal.yaml` that you changed in the section above.

8. Edit the `deploy/kubernetes/X.Y.Z/operator-deploy/deploy.sh`
9. Comment out the line containing `certified-operators`. The difference should look like the following:

```
diff --git a/deploy/kubernetes/1.6.0/operator-deploy/deploy.sh b/deploy/kubernetes/1.6.0/operator-deploy/deploy.sh
```

Response:

```
diff --git a/deploy/kubernetes/1.6.0/operator-deploy/deploy.sh b/deploy/kubernetes/1.6.0/operator-deploy/deploy.sh
index b556bc3..3462943 100755
--- a/deploy/kubernetes/1.6.0/operator-deploy/deploy.sh
+++ b/deploy/kubernetes/1.6.0/operator-deploy/deploy.sh
@@ -211,7 +211,7 @@ deploy_cte_csi()
{
    if [ ${IS_OCP} -eq 1 ]; then
        VALIDATE=""
        # sed -i s/"^  source: .*"/"  source: certified-operators"/g ${DEPLOY_SCRIPT_PATH}/ctek8soperator-subscription.yaml
```

```

        # sed -i s/"^  sourceNamespace: .*/"  sourceNamespace: openshift-mark
etplace"/g ${DEPLOY_SCRIPT_PATH}/ctek8soperator-subscription.yaml
    else
        # some fields in the manifests for Openshift are not yet supported on
Kubernetes.
    }

```

Deploy CTE for Kubernetes Operator

- Deploy CTE for Kubernetes Operator with the `deploy.sh` script.

```
./deploy.sh
```

Response

```

Starting the cte-csi containers.
Automatic detection of CRI socket is enabled
Using default CRI socket path /run/crio/crio.sock for container runtime cri-o
Using CRISocket path: /run/crio/crio.sock
-----
CipherTrust Transparent Encryption for Kubernetes Operator will be deployed in
namespace: kube-system
CipherTrust Transparent Encryption for Kubernetes will be deployed in namespac
e: kube-system
-----
serviceaccount/cte-csi-controller created
clusterrole.rbac.authorization.k8s.io/cte-csi-controller-ac created
clusterrolebinding.rbac.authorization.k8s.io/cte-csi-controller-binding created
serviceaccount/cte-csi-node created
clusterrole.rbac.authorization.k8s.io/cte-csi-node-ac created
clusterrolebinding.rbac.authorization.k8s.io/cte-csi-node-binding created
operatorgroup.operators.coreos.com/ctek8soperator-og created
subscription.operators.coreos.com/ctek8soperator-sub created
Waiting for InstallPlan to be instantiated and approved.....
Successfully installed CipherTrust Transparent Encryption for Kubernetes Operat
or
Deploying CipherTrust Transparent Encryption for Kubernetes
Waiting for the ctek8soperator CRD to get registered with K8s..
Successfully registered ctek8soperator CRD with K8s
Waiting for ctek8soperator controller manager to get ready.....
ctek8soperator controller manager is ready
ctek8soperator.cte-k8s-operator.csi.cte.cpl.thalesgroup.com/ctek8soperator crea
ted

```

```

Waiting for CipherTrust Transparent Encryption for Kubernetes.....
=====
CipherTrust Transparent Encryption for Kubernetes Operator deployed in namespace kube-system
CipherTrust Transparent Encryption for Kubernetes in namespace kube-system
cte-csi-controller-6dd49c67f4-gf4x9          0/5      ContainerCreatin
g    0          1s
cte-csi-node-2wj4s                          0/4      ContainerCreatin
g    0          1s
cte-csi-node-bjntf                          0/4      ContainerCreatin
g    0          1s
cte-csi-node-mp4xv                          0/4      ContainerCreatin
g    0          1s
cte-csi-node-r9ldw                          0/4      ContainerCreatin
g    0          1s
cte-csi-node-w7gzf                          0/4      ContainerCreatin
g    0          1s
=====

```

Appendix: Sample Script

```

kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: ocp-discon.lab.ocp.lan:5000/cteoperator/operator-metadata
    skipTLS: false
  mirror:
    operators:
      - catalog : docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernetes-operator-catalog:v1.2.5
    packages:
      - name: cte-k8s-operator
        defaultChannel: stable
        channels:
          - name: stable
    additionalImages:
      - name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernetes:<current_version.build>
      - name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernetes:<current_version>-latest
      - name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernetes:<current_version.build>
      - name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernetes:<current_version>-latest
      - name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernetes:<current_version.build>
      - name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernetes:<current_version>-latest
      - name: registry.k8s.io/pause:3.9
      - name: registry.k8s.io/sig-storage/csi-provisioner:v4.0.0

```

```
- name: k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.0.1
- name: k8s.gcr.io/sig-storage/csi-attacher:v3.3.0
- name: registry.k8s.io/sig-storage/csi-snapshotter:v6.3.3
- name: registry.k8s.io/sig-storage/csi-resizer:v1.8.0
```

Example of Additional Images

```
- name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernet
es:1.6.0.27
- name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernet
es:1.6.0-latest
- name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernet
es:1.5.0.37
- name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernet
es:1.5.0-latest
- name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernet
es:1.4.0.33
- name: docker.io/thalesciphertrust/ciphertrust-transparent-encryption-kubernet
es:1.4.0-latest
```

Preparing a CTE for Kubernetes Volume for Deployment

The following sections must be completed in order to ensure success with deploying CTE for Kubernetes.

1. [Preparing CipherTrust Manager](#)
2. [Create a CTE Policy for Kubernetes](#)
3. [Using Kubernetes service accounts](#)
4. [Prepare a file system persistent volume](#)
5. [Deploying Pods/Services in a Kubernetes Cluster](#)
6. [Deploying CTE for Kubernetes Storage Classes](#)
7. [Using Dynamic PVCs with CTE for Kubernetes](#)
8. [Creating a CTE for Kubernetes Claim Against the PVC](#)
9. [Individual Guard Policy Tuning](#)
10. [Create a CTE-CSI protected Persistent Storage Claim by deploying an application/staging pod](#)
11. [Using Taints and Tolerations](#)
12. [Allow only Trusted Pods to mount to CTE for Kubernetes volumes](#)

Preparing CipherTrust Manager

Note

Make sure that all of the names are written in lower case letters or else the yaml file will generate errors.

Managing Kubernetes Storage Groups and Clients

For information on how to create and manage Kubernetes (K8s) storage groups, K8s clients, apply GuardPolicies to storage groups, and protect K8s clients, see the CipherTrust Manager K8s documentation:

- [Managing Kubernetes Storage Groups](#)
- [Managing Kubernetes Clients](#)
- [Protecting Kubernetes Clients](#)

Create a CTE Policy for Kubernetes

For the generic instructions on how to create a CTE for Kubernetes policy, see [Protecting Kubernetes Clients](#).

Specifically, when creating a policy for protecting Persistent Volumes in CTE for Kubernetes: CTE for Kubernetes

1. Create a policy with type : **CTE for Kubernetes**
2. For the CTE CSI policy name, use the name listed for the policy parameter:
`csi.cte.cpl.thalesgroup.com/policy` in the `cte-csi-claim.yaml` file.
3. Make the GuardPolicy name the same name as the CSI policy.
4. Attach this GuardPolicy to the K8s Storage Group.

Note

CTE for Kubernetes only supports **User Sets** with a UID and GID. User names and Group names are **not** supported.

Using Kubernetes service accounts

A service account is a type of Kubernetes account that provides a distinct identity, in a Kubernetes cluster, so that you can allow or restrict access based on this service account. Application Pods, system components, and entities inside and outside the cluster can use a specific Service Account's credentials to identify as that Service Account. This is needed because the UID inside a K8s pod is not persistent, and changes frequently in OpenShift deployments. Therefore, access control based on user IDs is not feasible for such workloads.

Some service accounts may have administrator privileges to use any Kubernetes API resource on the cluster, while others may only have the privilege to start pods without any other access to the Kubernetes API.

Each service account is bound to a Kubernetes namespace. Every namespace gets a default ServiceAccount upon creation.

Configuring on CipherTrust Manager

Configure this feature on CipherTrust Manager. The feature uses the existing User Set. It leverages the unused `UNAME` value from the User Set, remapping it for use as a Service Account name. In your policy, you can add rules that allow or deny access to a CTE volume-based on the service account pod status.

- See [Creating User Sets](#) for more information.

Creating Namespace & ServiceAccounts in Kubernetes

1. Create a namespace, type:

```
NS=<service-account-name>
kubectl create ns "$NS"
```

Example

```
NS=<demo-sa>
kubectl create ns "$NS"
```

2. Create service accounts, type:

```
kubectl -n "$NS" create serviceaccount <service-account-role>
```

Example

```
kubectl -n "$NS" create serviceaccount sa-restricted
kubectl -n "$NS" create serviceaccount sa-admin
```

Note

An `sa-restricted` service account **cannot execute** pods. Only an `sa-admin` service account **can execute** pods.

Restricted Role

The restricted role allows only the following functions on pods:

- get
- list
- watch
- create

Role-Based Access Control (RBAC) Examples

The following simple examples are provided so that you can use them to evaluate the feature. Use the service account feature in the appropriate method for solving your use case. Refer to the following Kubernetes documentation: [Service Accounts](#), for more information.

Role: Restricted

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: restricted
  namespace: demo-sa
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "list", "watch", "create"]
```

Role: Administrator

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: admin-lite
  namespace: demo-sa
rules:
- apiGroups: [""]
  resources: ["pods", "pods/exec", "configmaps"]
  verbs: ["*"]
- apiGroups: ["apps"]
  resources: ["deployments"]
  verbs: ["*"]
```

Role: K8s Restricted --> Service account restricted

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
```

```

name: restricted-to-sa-restricted
namespace: demo-sa
subjects:
- kind: ServiceAccount
  name: sa-restricted
  namespace: demo-sa
roleRef:
kind: Role
name: restricted
apiGroup: rbac.authorization.k8s.io

```

K8s Administrator --> Service Account Administrator

```

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
name: admin-lite-to-sa-admin
namespace: demo-sa
subjects:
- kind: ServiceAccount
  name: sa-admin
  namespace: demo-sa
roleRef:
kind: Role
name: admin-lite
apiGroup: rbac.authorization.k8s.io

```

- To apply it, type:

```
kubectl apply -f rbac.yaml
```

Pod manifests

Demo of Service Account restricted yaml file

```

apiVersion: v1
kind: Pod
metadata:
name: demo
namespace: demo-sa
spec:
serviceAccountName: sa-restricted
volumes:
- name: data
  persistentVolumeClaim:
    claimName: cte-clain
containers:
- name: app
  image: busybox:1.36
  command: ["sh", "-c", "sleep 365d"]
  volumeMounts:

```

```
- name: data
  mountPath: /data
```

- To apply it, type:

```
kubectl apply -f demo-pod-sa-restricted.yaml
```

Demo as Service Account Administrator yaml file

```
apiVersion: v1
kind: Pod
metadata:
  name: demo
  namespace: demo-sa
spec:
  serviceAccountName: sa-admin
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: cte-clain
  containers:
    - name: app
      image: busybox:1.36
      command: ["sh", "-c", "sleep 365d"]
      volumeMounts:
        - name: data
          mountPath: /data
```

- To apply it, type:

```
kubectl apply -f demo-pod-sa-admin.yaml
```

Prepare a file system persistent volume

Prerequisite

1. Create an NFS Server.
2. Export the path.
3. In the example below, the `yaml` file uses the NFS server `10.10.10.10` and the exported path is: `/my/nfs/volume/path`.

Limitation

CTE for Kubernetes **does not support** mounting conventionally-protected PVCs and CTE-protected PVCs simultaneously. This could lead to unexpected behavior.

Create a Persistent Storage (PV) YAML file description

The following example explains how to create a persistent storage for CTE for Kubernetes to guard. Find more information on creating Persistent Volumes at:

- [Persistent Volumes](#)

Note

Replace the dummy entries for path and server with your own working NFS configuration.

nfs-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs-test-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteMany
  storageClassName: nfs
  persistentVolumeReclaimPolicy: Retain
  mountOptions:
    - hard
  nfs:
    path: /my/nfs/volume/path
    server: 10.10.10.10
```

Create a Persistent Storage Claim (PVC) YAML file description

nfs-claim.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs-test-claim
spec:
  storageClassName: nfs
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

Note

The PV name is not referenced in the PVC. This particular claim will claim any PV that matches the requirements.

Apply Persistent Storage YAML files

Type:

```
kubectl apply -f nfs-pv.yaml
kubectl apply -f nfs-claim.yaml
```

Deploying Pods/Services in a Kubernetes Cluster

The CTE for Kubernetes images are distributed through the [Thales Docker Hub](#).

- All of the pods/services are deployed using `yaml` files. The `yaml` files are executed using the provided `deploy.sh` scripts.
- CTE for Kubernetes **only** supports homogeneous Kubernetes clusters, i.e., all nodes of the Kubernetes clusters must be x86_64, or all nodes must be arm64.

Pod Types Created

The installation of CTE for Kubernetes creates two types of pods:

- `cte-csi-controller-XXXXXXXXXX-XXXXX`
 - Kind: **Pod Deployment Type**

A Deployment manages a set of Pods to run an application workload. After you describe a desired state in a Deployment, and the Deployment Controller changes to that actual state at a controlled rate, you can define Deployments to create new ReplicaSets, or to remove existing Deployments, and adopt all of their resources with new Deployments. See [Kubernetes Deployments](#) for more information.

- One Pod, started by default, may be scaled up post installation
- `cte-csi-node-XXXXX`
 - Kind: **Pod DaemonSet Type**

A DaemonSet defines Pods that provide node-local facilities. It ensures that all, or a subset of, nodes run a copy of a specified Pod. As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected. Deleting a DaemonSet will clean up the Pods it created. See [DaemonSet](#) for more information.

- One Pod for each Kubernetes node in the cluster

Options for Deploy Scripts for Pods

Deploy all of the pods by using the following command and needed arguments from the following table:

Short Option	Long Option	Description
-r	--remove	Remove all the running pods, services and secrets.
-t	--tag	Tag of the image on the server. Default is the latest.
-o	--operator	Deploy the CTE for Kubernetes Operator and CSI driver.
	--operator-ns=	The namespace in which to deploy the Operator.
	--cte-ns=	The namespace in which to deploy the CSI driver.
	--cri-sock=	Container Runtime Interface socket path.

For CTE for Kubernetes v1.3.0 and subsequent versions, the deployment script uses the repository image index as the image name, instead of the individual platform image names. The manifest (`cte_csi : <tag>`) points to an image.

Note

The **default** image name is: `cte_csi`. You do not need to specify the image name if you use the default name.

Terminating Pods

To terminate all of the pods and delete all of the services and secrets:

1. Change to the CTE for Kubernetes directory, type:

```
cd ciphertrust-transparent-encryption-kubernetes
```

2. For Operator deployment, type:

```
./deploy.sh --remove --operator-ns=my-ns1 --cte-ns=my-ns2
```

3. For Helm deployment, type:

```
./deploy.sh --remove --helm
```

Verify CTE for Kubernetes

Verify that CTE for Kubernetes is running, type:

```
kubect1 get pods --namespace=kube-system -o wide | grep cte-csi
```

RESPONSE

NAME	READY	STATUS	RESTART
S AGE IP Node			
kube-system cte-csi-controller-5db888d6cb-tn6lr	4/4	Running	0
6m59s 10.244.1.5 ubuntu20-02-kubcluster-worker	<none>	<none>	
kube-system cte-csi-node-lz7t9	4/4	Running	0
6m59s 10.244.0.26 ubuntu20-02-kubcluster-master	<none>	<none>	
kube-system cte-csi-node-pzvwb	4/4	Running	0
6m59s 10.244.2.2 ubuntu20-02-kubcluster-worker2	<none>	<none>	
kube-system cte-csi-node-wmhh1	4/4	Running	0
6m59s 10.244.1.4 ubuntu20-02-kubcluster-worker	<none>	<none>	

Deploying CTE for Kubernetes Storage Classes

For information on K8 storage classes, see [Storage Classes](#) for more information.

Note

All of the Kubernetes clients that you want to attach to a storage group must have the same Kubernetes Namespace and Kubernetes structureless pods.

To deploy a storage class for CTE for Kubernetes:

1. Create and save a registration token in CipherTrust Manager. See [Tokens](#) for more information.

Note

Select **Base64** format for the registration token, if using CipherTrust Manager v2.10 and subsequent versions.

2. Create a K8s Storage Group in CipherTrust Manager. See [Creating a K8s Storage Group](#) for more information.
3. If using CipherTrust Manager v2.9 or previous versions, encode the token in base64 format, type:

```
echo -n <CM REGISTRATION TOKEN STRING> | base64 -w 0
```

4. Copy the base64 encoding to create a Kubernetes secret YAML file, **cte-csi-cmtoken.yaml**:

```
apiVersion: v1
kind: Secret
metadata:
  name: <CHANGE to name of the K8s secret. For example: cm-reg-token>
type: Opaque
data:
  # This is a base64 encoded registration token. To generate:
  # echo <CM REGISTRATION TOKEN STRING> | base64 -w 0
  registration_token: bWlEaUJlZ08xNkNsbnRqZmc4a1dvcU1SUG9uaVpnNkVtUjVYSGFLUV
  ZVTHRhbGRrb0M5T1ZwTEpvTXp4UldmSQ==
```

5. The name of the K8s secret must be embedded in the `registration_token_secret` parameter in the storage class YAML file. Use the YAML file, **cte-storageclass.yaml** and fill in the appropriate values.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: <CHANGE to name of the Kubernetes Storage Class. For example: csi-test-sc>
provisioner: csi.cte.cpl.thalesgroup.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
parameters:

  # Domain name or IP address of the CipherTrust Manager (Required)
  key_manager_addr: <CHANGE_ME to your CM IP ADDR>

  # Name of the CipherTrust Manager K8s Storage Group. (Required)
  k8_storage_group: <CHANGE to the name of the Kubernetes Storage Group. For example: test-group>

  # Kubernetes Secret with CM registration token (Required)
  registration_token_secret: <CHANGE to the K8s secret. For example: cm-reg-token>

  # Time in minutes to wait before unregistering from the CipherTrust Manager
  # once all of the volumes have been unguarded. Parameter must be added as a string
```



```
# integer value. Default is 10 minutes. (Required)
registration_period: "10"
```

- Record the storage class name, for further use. Deploy by typing:

```
kubectl apply -f cte-csi-cmtoken.yaml
kubectl apply -f cte-storageclass.yaml
```

Registering to Multiple CipherTrust Manager Servers for failover in an HA Cluster

For a cluster for CipherTrust Manager servers, you can configure multiple IP addresses with `cte-storageclass.yaml` to allow for seamless failover to the next available CipherTrust Manager if a CipherTrust Manager server fails.

- CipherTrust Manager IP addresses must be comma separated. For example:

```
key_manager_addr: <CM_1_IP_ADDR>, <CM_2_IP_ADDR>, <CM_3_IP_ADDR>
```

- You can provide one CM IP address, or multiple CM IP addresses. Thales recommends using more than one IP address.

Note

When providing multiple IP addresses, the IP addresses **must** be part of same CipherTrust Manager cluster.

Using Dynamic PVCs with CTE for Kubernetes

Static provisioning for Persistent Volumes (PV) requires the administrator to make assumptions, in order to create PVs that applications may need. As your Kubernetes environment expands, this can become a bottleneck.

Dynamic provisioning solves this issue. Instead of the Kubernetes administrator creating specific PVs, the administrator defines Storage Classes. Each Storage Class has a specific storage pool from which PVs can be provisioned automatically to meet an application's requirements.

Kubernetes provides a variety of internal provisioners. With dynamic provisioning, a developer can use a PVC to request a specific storage type and have a new PV provisioned automatically. The PVC must request a StorageClass that has already been created and configured on the target cluster, by an administrator, for dynamic provisioning to work.

CTE for Kubernetes now allows you to deploy HELM charts that use a StorageClass as input for creating volumes. Helm Charts help you define, install, and upgrade Kubernetes applications, thereby helping you manage Kubernetes clusters.

Specifying a Storage Class on a HELM chart means that the cluster will select volumes from the pool of volumes for that storage class. If no volumes exist, but the CTE-CSI driver has the capabilities for automatic provisioning, a new volume will be created and added to the cluster. Traditional CTE for Kubernetes PVCs must be specified with a source PVC in the definition, which makes it incompatible with this type of deployment. By using HELM charts, this will allow CTE for Kubernetes volumes to pass enough information to a source Storage Class so that dynamic provisioning in a CTE for Kubernetes PVC can dynamically provision and attach a data source PVC based on the PVC specifications.

Specifically, with dynamic PVCs now being supported, it changes the creation method for PVs. The CTE-K8s StorageClass is created with the new parameters and the CTE-K8s PVC is created without sourcePVC annotations, (policy annotation is optional). After the controller checks for the existence of policy and sourcePVC parameters, if it doesn't find sourcePVC parameters, it creates a new unprotected PVC (source PVC) using parameters from the CTE-K8s PVC and CTE-K8s StorageClass. CTE-K8s then binds the unprotected PVC to a PV if any qualifying PVs are available on the cluster. If not, it requests for a new PV to be provisioned. Then the CTE-K8s PVC provisions the PVC as normal. After that, the Application Pod can use the CTE-K8s PVC claim.

CTE-K8s Storage Class definition

To support dynamic PVCs, three new parameters must be added to the CTE-K8s Storage Class definition:

- `source_storage_class`
- `default_policy`
- `allow_source_pvc_delete`

Example

Following is an example of an entire Persistent Volume Claim set with the new parameters:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-test-sc
provisioner: csi.cte.cpl.thalesgroup.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
parameters:

  # Domain name or IP address of the CipherTrust Manager (Required)
  key_manager_addr: 192.168.70.1

  # Name of the CipherTrust Manager K8s Storage Group. (Required)
  k8_storage_group: test-group

  # Kubernetes Secret with CipherTrust Manager registration token (Required)
  registration_token_secret: cm-reg-token

  # Kubernetes secret with External CA signed client certificate (Optional)
  kubectrl create secret generic <secret-name> --from-file=<clientName>.crt=<client_cert.pem> --from-file=<clientName>.key=<client_key.pem> --from-file=<clientName>.passphrase=<passphrase>

  # Multiple client cert details can be added with same secret as well with abo
```

```

ve command
  external_ca_client_secret: <secret-name>

  # Small registration description to be displayed in the CipherTrust Manager
  (Optional)
  client_description: "Describe your K8s client"

  # Time in minutes to wait before unregistering from the CipherTrust Manager
  # once all volumes have been unguarded. Parameter must be added as a string
  # integer value. Default "10" minute. (Optional)
  registration_period: "10"

  # When specified, this parameter automatically adds the
  # csi.cte.cpl.thalesgroup.com/source_pvc parameter to the CTE-K8s PVC based
  # on the requested parameters (Optional)
  source_storage_class: some_sc_name

  # When specified, this parameter automatically adds the
  # csi.cte.cpl.thalesgroup.com/policy parameter to the CTE-K8s PVC based on
  # the requested parameters. (Optional) (Required if source_storage_class is s
  et)
  default_policy: <default_policy_name>

  # When specified and set to "true", this parameter automatically deletes the
  dynamic sourcePVC,
  # and possibly, the actual data volume, depending on the provisioner driver i
  mplementation.
  # If set to "false," you must manually delete the created source PVC. (Option
  al)
  allow_source_pvc_delete: "false"

```

New Persistent Volume Claim Usage

The added support for dynamic PVCs changes some of the rules for PVCs:

- The `source_pvc` and `policy` annotations in CTE-PVC are now optional if you use the new Storage Class parameters
- If you configured the `policy` annotation in the CTE-PVC, then CTE-K8s uses this policy regardless if `default_policy` is specified or not.
- If the `policy` annotation is not configured in the CTE-PVC, then CTE-K8s uses the `default_policy` from CTE-StorageClass.
- If both policy annotations are missing, then the deployment fails.
- If the `source_pvc` is configured in the CTE-PVC, then this source PVC is used, regardless of the `source_storage_class` parameter specified.
- If the `source_pvc` is not configured in the CTE-PVC, then CTE-K8s uses the `source_storage_class` parameter specified in CTE-StorageClass to provision dynamically. The `cte-csi` driver creates a source PVC (i.e. `cte-claim-unprotected-cprfnzjxjg3gpyabrqh5d`) using `source_storage_class` and the storage size parameter in the CTE-PVC request.

Example: Persistent Volume Claim

Using these new parameters, you could configure CTE for Kubernetes as follows:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cte-claim1
  annotations:
    csi.cte.cpl.thalesgroup.com/policy: policy_1
    csi.cte.cpl.thalesgroup.com/source_pvc: nfs-test-claim
spec:
  storageClassName: csi-test-sc
  accessModes:
    - ReadWriteMany
resources:
  requests:
    storage: 1Gi No longer ignored by the driver
```

Applying the CTE-PVC

To apply the CTE-PVC manifest, type:

```
kubectl apply -f cte-csi-claim.yaml
```

Based on the CTE Storage Class `volumeBindingMode` and the underlying source Storage Class `volumeBindingMode`, a new volume is provisioned at the time of the CTE-PVC deployment or App Pod deployment.

PVC status Example:

NAME	STATUS	VOL
UME	STORAGECLASS	
VOLUMEATTRIBUTESCLASS	AGE	
persistentvolumeclaim/cte-claim-restore		Boun
d pvc-b168e8d9-e26b-4515-aab2-91dc241ce74a	2Gi	RWX
t-sc <unset>	13m	cte-tes
persistentvolumeclaim/cte-claim-restore-unprotected-cprfnzjxjxg3gpyabrqh5d		Boun
d pvc-4183880c-6480-43e8-9564-b9f06fe557bf	2Gi	RWX
i-imm <unset>	13m	nfs-cs

- **CTE-PVC**: `cte-claim-restore`
- **Source PVC**: `cte-claim-restore-unprotected-<UID>`

CTE-CSI protected Persistent Storage Claim

Creating a CTE-CSI protected Persistent Storage Claim by deploying an application pod:

```
apiVersion: v1
```

```

kind: Pod
metadata:
  name: cte-csi-demo
spec:
  volumes:
    - name: test-vol
      persistentVolumeClaim:
        claimName: cte-claim1
  containers:
    - name: ubuntu
      image: ubuntu
      volumeMounts:
        - mountPath: "/data"
          name: test-vol
      command:
        - "sleep"
        - "604800"
      imagePullPolicy: IfNotPresent
      restartPolicy: Always

```

External Provisioner support with CTE-K8s for dynamically provision volume

If your driver is not capable of dynamically provisioning the persistent volume for a storage type, you can use an external provisioner for that specific storage type to create the volume dynamically and deploy with CTE for Kubernetes. Following are some external provisioners which provision data volumes dynamically for a storage class. You can use this storage class as the source StorageClass in cte-storageclass.yaml parameter (source_storage_class).

- [NFS Subdirectory External Provisioner](#)
- [NFS CSI driver for Kubernetes](#)
- [AWS-EFS CSI dynamic provisioning](#)
- [Drivers list for different type of storage that may be used for Dynamic Volume Provisioning](#)

After deploying a cte-claim1.yaml, the data persistent volume is created dynamically.

Creating a CTE for Kubernetes Claim Against the PVC

Note

For the next step, you must have your PV and PVC already deployed.

1. Create the following YAML file:

cte-csi-claim.yaml

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cte-claim
  annotations:
    # The following must match your CTE Kubernetes Policy name.
    csi.cte.cpl.thalesgroup.com/policy: policy_1
    # NFS source persistent volume claim
    csi.cte.cpl.thalesgroup.com/source_pvc: nfs-test-claim
spec:
  storageClassName: <CHANGE to the storageclass name that you deployed. For
example: e.g. csi-test-sc>
  accessModes:
    - ReadWriteMany
resources:
  requests:
    storage: 1Ki

```

2. To deploy the `cte-csi-claim.yaml` script, type:

```
kubectl apply -f cte-csi-claim.yaml
```

Note

- AccessMode of CTE-PVC **must** be the same as the sourcePVC accessMode.
- Multiple CTE-PVC **cannot** use same sourcePVC (`csi.cte.cpl.thalesgroup.com/source_pvc: nfs-test-claim`) at the same time.

Individual Guard Policy Tuning

Note

In the code commands and responses, the term GuardPoint means Guard Policy.

Configuration

There are two scenarios for CTE for Kubernetes Guard Policy configuration tuning.

On the Guard

This is for creating a Guard Policy with the required configuration settings, other than the default values. Guard Policy creation occurs when you deploy an application pod using a CTE-PVC. Each CTE-PVC deployment results in the creation of a unique Guard Policy.

Note

- If the same CTE-PVC is used by multiple application pods deployed on the same node, then the pod will contain a single Guard Policy.
- If the same CTE-PVC is used by multiple application pods deployed on different nodes, then on each node there will be a unique Guard Policy and the same tuning configuration changes will be set on those Policies.

On the Run

Configuration settings while Guard Policy is active. Guard Policy active means application pod is already running with CTE-PVC and user want to configure a Guard Policy. Only debug log level configuration is supported on the run for now.

Tunable Parameters

Configurable Parameter	Default Value	Description	Status	On the Guard Tuning Support	On the Run Tuning Support
debug_all	4	Sets the sensitivity of the logs, can be adjusted later using `secfsd -log_level [4-8]	Change as needed	yes	yes
debug_extra	0	Debug level	Change as needed	yes	no
enable_xattr	0	Enables use of xattrs in the file system, default is off for best performance	Set to 1 if performing a restore from CTE with xattrs. Set to 0 for normal use.	yes	no
fileinfo_cache_timeout	100	Amount of milliseconds to keep file attribute data in cache	Change as needed	yes	no
loginuid	1	Enforce loginuid. Without this set,	Change as needed	yes	no

Configurable Parameter	Default Value	Description	Status	On the Guard Tuning Support	On the Run Tuning Support
		su can bypass security			
max_worker_threads	10	Maximum parallel threads allowed	Change as needed	yes	no
mixed_policy	1	If mixed modes needed (ex: apply key on read, no apply key on write), set this value; it causes an extra access check	Change as needed	yes	no
nfs_user	0	UID of specific NFS user to use	Change as needed	yes	no
parallel_writes	1	Allows non-overlapping writes to run in parallel	Change as needed	yes	no
splice	0	Allows use of splice call from FUSE	Change as needed	yes	no
writeback_cache_local	1	Uses writeback cache for local file systems (extx, xfs, btrfs)	Change as needed	yes	no
writeback_cache_nfs	0	Uses writeback cache for NFS	Change as needed	yes	no

Preparing CTE for Kubernetes deployment files

Creating a ConfigMap with tuning parameters

- ConfigMap **must** be in the same namespace as the CTE-PVC namespace.
- **On the Guard** ConfigMap settings require the prefix `guard` with the tuning parameter name. You can use multiple tuning parameters at same time to configure a Guard Policy.
- **On the Run** ConfigMap settings require the prefix `run` with the tuning parameter name.
- Create a ConfigMap with tuning parameters and key-value pairs.
- Following is a ConfigMap example:

gp-tuning-config-map.yaml


```

apiVersion: v1
kind: ConfigMap
  metadata:
    name: cte-claim-gp-tuning-cm
    #namespace: <ns1>           # If CTE-PVC will be deployed in ns1 namespace

  data:
    #guard.<config-parameter-name>: "<config-value>"           # On the Guard tuning example
    #run.<config-parameter-name>: "<config-value>"           # On the Run tuning example (This field can be added/updated on the Run)
    guard.debug_all: "8"
    guard.max_worker_threads: "20"
    guard.writeback_cache_nfs: "1"
    #run.debug_all: "8"

```

Where

- `<config-parameter-name>`: Supported tuning parameter name (See above table)
- `<config-value>`: value corresponding to the parameter. Make sure values are in double quotation marks ("").
- Apply the ConfigMap, type:

```
kubectl apply -f gp-tuning-config-map.yaml
```

Creating a CTE-PVC with tuning enabled for "On the Guard"

New annotation will require for this feature:

- `csi.cte.cpl.thalesgroup.com/guardpoint_tuning_configmap: <tuning-config-map-name>`
- `<tuning-config-map-name>`: ConfigMap name which contains the tuning parameters. For example, `cte-claim-gp-tuning-cm`.

cte-csi-claim.yaml

```

apiVersion: v1
kind: PersistentVolumeClaim
  metadata:
    name: cte-claim
    annotations:
      # GuardPolicy from CipherTrust Manager

```

```

csi.cte.cpl.thalesgroup.com/policy: policy_1
#(Following is not required if using dynamic provisioning)
csi.cte.cpl.thalesgroup.com/source_pvc: nfs-test-claim
csi.cte.cpl.thalesgroup.com/guardpoint_tuning_configmap: <tuning-config-m
ap-name>

spec:
  storageClassName: <CHANGE ME to the storageclass name deployed e.g. csi-test-s
c>
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Ki

```

1. Apply the CTE PVC, type:

```
kubectl apply -f cte-csi-claim.yaml
```

Now you can use the CTE PVC with an application pod. All of the configuration changes take affect once the Guard Policy is created when the application pod deploys.

2. Validate the applied changes with the corresponding `cte-csi-node-xxxxx` pod log (with the `cte-csi` driver logLevel 5):

```
kubectl logs --namespace=kube-system cte-csi-node-xxxxx cte-csi
```

Example Response

```

[I0626 10:17:28.160501 3725529 guardpoint.go:1634] Guard Policy tuning is enab
led for cte-pvc[cte-claim] namespace[default] policy[policy_1] path[/tmp 2e1df
83a-33a5-11ef-823e-3e833176f6ce]
[I0626 10:17:28.166396 3725529 chroot.go:108] Executing /usr/bin/voradmin secf
s config debug_all 8 /tmp/2e1df83a-33a5-11ef-823e-3e833176f6ce on Chroot UUID
e51ad1bb-3809-46b2-8497-40aec3d9eee1
[I0626 10:17:28.214394 3725529 chroot.go:108] Executing /usr/bin/voradmin secf
s config max_worker_threads 20 /tmp/2e1df83a-33a5-11ef-823e-3e833176f6ce on Ch
root UUID e51ad1bb-3809-46b2-8497-40aec3d9eee1
[I0626 10:17:28.258587 3725529 chroot.go:108] Executing /usr/bin/voradmin secf
s config writeback_cache_nfs 1 /tmp/2e1df83a-33a5-11ef-823e-3e833176f6ce on Ch
root UUID e51ad1bb-3809-46b2-8497-40aec3d9eee1

```

Using "On the Run" tuning

Use the existing ConfigMap, or a new ConfigMap, with the data field `run.<config-parameter-name>:<config-value>`:

1. After adding/updating the parameter, apply the ConfigMap, type:

```
kubectl apply -f <gp-tuning-config-map.yaml>
```

Where

<gp-tuning-config-map.yaml>: ConfigMap yaml file which contains the tuning parameter

2. Add the annotation to the CTE PVC using the `kubectl annotate` command, type:

```
kubectl annotate pvc <cte-pvc-name> csi.cte.cpl.thalesgroup.com/guardpoint_tuning_configmap='<tuning-config-map-name>'
```

Where

- <cte-pvc-name>: CTE-PVC name, for example, `cte-claim`
- <tuning-config-map-name>: ConfigMap name which contains the tuning parameters. For example, `cte-claim-gp-tuning-cm`

After applying the annotation, the applied configuration takes effect within a minute.

3. Validate applied changes with the corresponding `cte-csi-node-xxxxx` pod log, type:

```
kubectl logs --namespace=kube-system cte-csi-node-xxxxx cte-csi
```

Example Response

```
[I0626 10:33:26.887047 3725529 guardpoint.go:1745] Runtime GuardPoint tuning is enabled. ConfigMap[cte-claim-gp-tuning] in CTE-PVC[cte-claim1] namespace[default] annotation for volumeID[2e1df83a-33a5-11ef-823e-3e833176f6ce] guardPolicy[pvc-clone-policy1] path[/tmp/2e1df83a-33a5-11ef-823e-3e833176f6ce] for regID[e51ad1bb-3809-46b2-8497-40aec3d9eee1]
[I0626 10:33:26.889947 3725529 chroot.go:108] Executing /usr/bin/secfsd -log_level 7 /tmp/2e1df83a-33a5-11ef-823e-3e833176f6ce on Chroot UUID e51ad1bb-3809-46b2-8497-40aec3d9eee1
[I0626 10:33:26.929057 3725529 chroot.go:130] Log level [7][all][ /tmp/2e1df83a-33a5-11ef-823e-3e833176f6ce]
```

Create a CTE-CSI protected Persistent Storage Claim by deploying an Application/Staging pod

To create a CTE-CSI protected Persistent Storage Claim by deploying an application pod:

1. Create the yaml file to deploy a pod and protect the contents of the Persistent Volume.

```

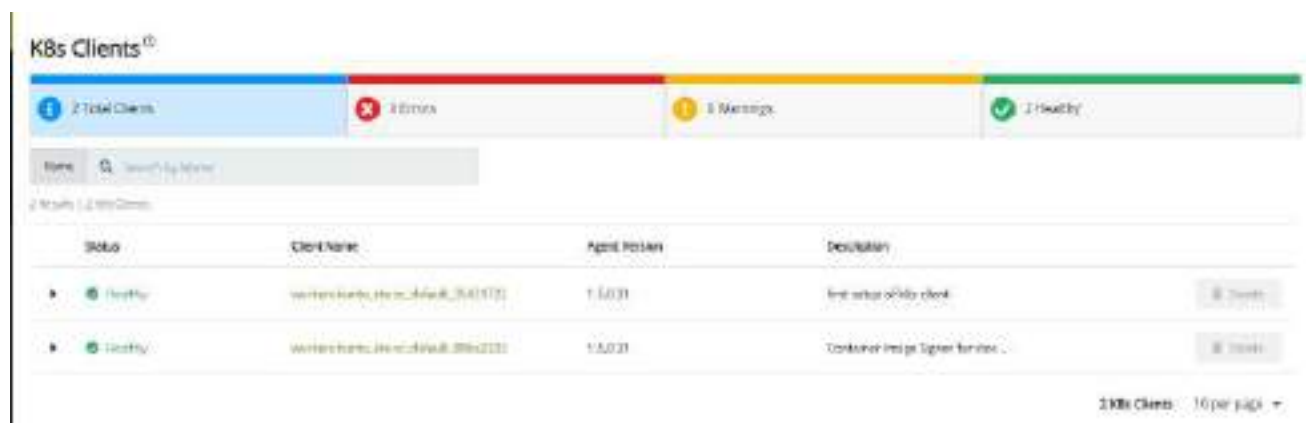
apiVersion: v1
kind: Pod
metadata:
  name: cte-csi-pod
spec:
  volumes:
    - name: test-vol
      persistentVolumeClaim:
        claimName: cte-test-claim1
  containers:
    - name: ubuntu
      image: ubuntu
      volumeMounts:
        - mountPath: "/data"
          name: test-vol
      command:
        - "sleep"
        - "604800"
      imagePullPolicy: IfNotPresent
      restartPolicy: Always

```

2. Deploy the pod, type:

```
kubectl apply -f cte-csi-pod.yaml
```

After the pod is successfully deployed and it running, you can see the client registered in K8s client.



The screenshot shows the 'K8s Clients' dashboard. At the top, there are four status bars: '2 Total Clients' (blue), '2 Errors' (red), '1 Messages' (yellow), and '1 Healthy' (green). Below these is a search bar and a table with 2 results. The table has columns for Status, Client Name, Agent Version, and Description. Both clients are in a 'Healthy' status.

Status	Client Name	Agent Version	Description
Healthy	writer-k8s-client-464836173	1.5.0.31	first writer policy client
Healthy	writer-k8s-client-464836173	1.5.0.31	Container Image Tigger service

At the bottom right, it says '2 K8s Clients' and '10 per page'.

3. To check the status of the pod, type:

```
kubectl get all
```

Response

NAME	READY	STATUS	RESTARTS
AGE			
pod/cte-csi-pod	1/1	Running	0
116m			
pod/cte-csi-user-pod	1/1	Running	0
134m			
pod/cte-staging-pod9jhtn	1/1	Running	0
133m			
pod/cte-staging-podt8jxs	1/1	Running	0
116m			

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
P	PORT(S)	AGE	
service/kubernetes	ClusterIP	10.96.0.1	<none>
443/TCP	224d		

After the pod is successfully deployed and running, you can see the clients registered in the Kubernetes client.

4. Login to the pod and check if the files are encrypted:

```
kubectl exec -it cte-csi-pod /bin/bash
```

Response

```
kubectl exec [POD] -- [COMMAND]
root@cte-csi-pod:/ cd /data/
root@cte-csi-pod:/data mkdir sub_dir
root@cte-csi-pod:/data cd sub_dir/
root@cte-csi-pod:/data/sub_dir echo "testfile" >> test
root@cte-csi-pod:/data/sub_dir
root@cte-csi-pod:/data/sub_dir cat test
testfile
```

5. Check the same file on the Server side:

```
root@aws-thales-dockerregistry:~ cd /nfs-share/sub_dir/
root@aws-thales-dockerregistry:/nfs-share/sub_dir ls -l
total 8
-rw-r--r-- 1 root root 4102 Dec 13 20:04 test
root@aws-thales-dockerregistry:/nfs-share/sub_dir cat test
```

```
EROV=====bu4廟41kCroot@aws-thales-dockerregistry:/nfs-share/sub_dir#
```

Using Taints and Tolerations

Note

Using Taints and Tolerations is **optional**.

You can apply Taints and Tolerations to the CTE for Kubernetes application pod that you are running. You can also create your staging pod with toleration.

Taints allow a node to repel a set of pods. **Tolerations** are applied to pods. Tolerations allow the scheduler to schedule pods with matching taints as long as it's allowed by the other associated parameters.

Taints and Tolerations work together to ensure that pods are not scheduled onto inappropriate nodes. Once one or more taints are applied to a node, the node should not accept any pods that do not tolerate the taints.

For more information, see [Kubernetes Taints and Tolerations](#).

Using Taints and Tolerations with CTE for Kubernetes

CTE for Kubernetes now supports nodes with taints. You can specify a toleration in a `pod_spec` for the staging pod of a CTE for Kubernetes volume. To do this, CTE for Kubernetes added annotation support to a CTE for Kubernetes PVC. This annotation uses a base64 encoded string for input. Kubernetes requires the pod to pass the info to the cluster as a single base64 encoded string. The only way to create this is to manually encode the yaml file that contains the Taints and Tolerations. CTE for Kubernetes takes the yaml file and extracts any toleration definitions contained in it and appends them to the staging pod when that PVC is mounted anywhere in the cluster. Any values in the YAML file that are not related to toleration are ignored.

Note

To apply Taints and Tolerations to a pod, you **must** first shut down the pod. Additionally, support for toleration **must** be added at PVC creation time. Modifying an existing PVC with the added toleration is **not** supported because the CTE PVC was not created with the required immutable volume attributes.

Applying Taints and Tolerations

1. Create and add the taint to a node, type:

```
kubectl taint nodes <nodeName> <keyName>=<valueName>:<taintEffect>
```

Example:

```
kubectl taint nodes node1 key1=value1:NoExecute
```

2. Create a tolerations yaml file. The tolerations must match the taint created in the previous step. The file should look similar to the following:

```
cat <cte_tolerations>.yaml
```

Response:

```
spec:
  tolerations:
  - key: "key1"
    operator: "Equal"
    value: "value1"
    effect: "NoExecute"
```

Note

The taint effect **NoSchedule** does not work with Taints and Tolerations for CTE for Kubernetes.

3. Write the tolerations yaml file to a text file while generating the base64 encoding and appending the file with the string, type:

```
cat <cte_tolerations>.yaml | base64 -w0
```

Response:

```
c3BlYzoKICB0b2x1cmF0aW9uczoKICAtIGtleTogImtleTEiCiAgICBvcGVyYXRvcjogIkVxdWFsIgogICAgdmFsdWU6ICJ2YWx1ZTEiCiAgICBlZmZlY3Q6ICJob0V4ZW51dGUicG==
```

4. Once the base64 string is generated, the PVC is generated. Append the value to the PVC. For example:

```
cat <cte_pvc_fileName>.yaml
```

Response:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
```

```

name: cte-csi-claim
annotations:
  csi.cte.cpl.thalesgroup.com/policy: <policyName>
  csi.cte.cpl.thalesgroup.com/source_pvc: <pvcName>-claim
  csi.cte.cpl.thalesgroup.com/spec_append: c3BlYzoKICB0b2xlcmlF0aW9uczoKI
CAAtIGtltTogImtltTEiCiAgICBvcGVyYXRvcjogIkVxdWFsIgogICAgdmFsdWU6ICJ2YWx1ZTEiCiA
gICBlZmZlY3Q6ICJob0V4ZWV1dGUlCg==
spec:
  storageClassName: <storageClassName>
  accessModes:
    - ReadWriteMany
resources:
  requests:
    storage: 10Gi

```

For more information, see the [Command line tool \(kubectl\)](#).

CTE Staging Container Resource Settings

This feature allows you to configure container resource settings for the CTE staging pod. This is useful if, for example, a default policy for resource settings is enabled for the container, but is not configured with CPU and memory resource limits, because this could cause the staging pod to terminate.

To configure:

1. Create a yaml file which contains the container resources:
 - a. Print the contents of the CTE tolerations file, type:

```
cat <cte_tolerations>.yaml
```

Response:

```

spec:
  containers:
    - name: <container-name>
      resources:
        limits:
          cpu: "500m"
          memory: "200Mi"
        requests:
          cpu: "500m"
          memory: "100Mi"

```

2. Generate the base64 encoded string, type:


```
cat <cte_tolerations>.yaml | base64 -w0
```

Response

```
c3BlYzoKwqAgY29udGFpbmVyczoKwqAgLSzOgrCoCDCoCDCoCBjcHU6ICI1MDBtIgrCoCDCoCD
CoCDCoCBtZW1.....sKgIMKgIMKgIMKgIG1lbW9yeTogIjEwME1pIgo=
```

3. Append the PVC parameter `csi.cte.cpl.thalesgroup.com/spec_append` with the base64 string, type:

```
cat <cte_pvc_fileName>.yaml
```

Response:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cte-csi-claim
  annotations:
    csi.cte.cpl.thalesgroup.com/policy: <policyName>
    csi.cte.cpl.thalesgroup.com/source_pvc: <source-PVC-Claim>
    csi.cte.cpl.thalesgroup.com/spec_append: c3BlYzoKwqAgY29udGFpbmVyczoKw
qAgLSzOgrCoCDCoCDCoCBjcHU6ICI1MDBtIgrCoCDCoCDCoCBtZW1.....sKgIMKg
IMKgIMKgIG1lbW9yeTogIjEwME1pIgo=
spec:
  storageClassName: <CTE-storageClassName>
  accessModes:
    - ReadWriteMany
resources:
  requests:
    storage: 10Gi
```

Note

You can use all of the supported spec append members with `csi.cte.cpl.thalesgroup.com/spec_append`. That means that you can enable/apply multiple features (toleration, container resource settings, or both features simultaneously).

For example, using both tolerations and container resource settings:

- `cat csi.cte.cpl.thalesgroup.com/spec_append`

Response:

```
spec:
  tolerations:
  - key: "testkey1"
    operator: "Equal"
    value: "testvalue1"
    effect: "NoExecute"
  containers:
  - name: ubuntu1
    resources:
      limits:
        cpu: "500m"
        memory: "200Mi"
      requests:
        cpu: "250m"
        memory: "100Mi"
```

Allow only Trusted Pods to mount to CTE for Kubernetes volumes

CTE for Kubernetes allows only trusted pods to access protected data volumes and attach to CTE for Kubernetes claims if you activate this optional feature. CTE for Kubernetes uses signature sets to validate the pod as trusted. Signatures are in the form of key-value pairs which contain the image name and the corresponding hash value. After mapping the image signature with the received signature set, CTE for Kubernetes will allow, or prevent, mounting of the the encrypted volume.

Overview of CTE for Kubernetes for Support for Trusted Pods

Support for Trusted Pods is disabled by default. At least one signature set, with a container image digest, must be attached to a security policy to enable this feature. When the request to mount/publish the volume is received:

1. CTE for Kubernetes determines whether Support for Trusted Pods is enabled or not by scanning for the presence of signature rules.
2. Support for Trusted Pods is skipped (default case) if the security policy does not include container signature sets.
3. CTE for Kubernetes fetches the digests for all containers in the running pod.
4. CTE for Kubernetes tries to match the digests of running pods to a container image signature set. Support for Trusted Pods is considered successful only if all digests within a running pod have an entry in same signature set.
5. CTE for Kubernetes checks the logs to determine if it succeeded. If no signature rules exist, the Support for Trusted Pods is disabled.

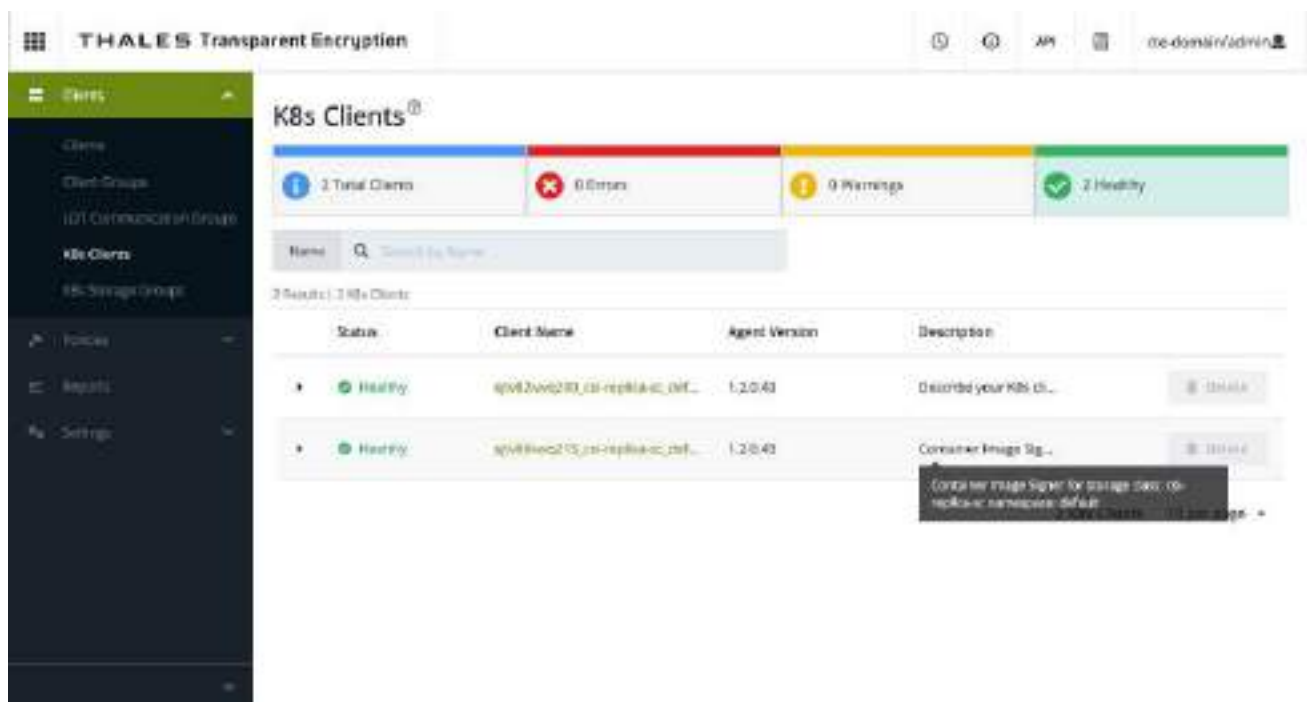
Note

Updating the signature set in the policy may lead to a small delay for CipherTrust Manager to push the signature back to the client. A Kubernetes timeout retry could also add a small delay.

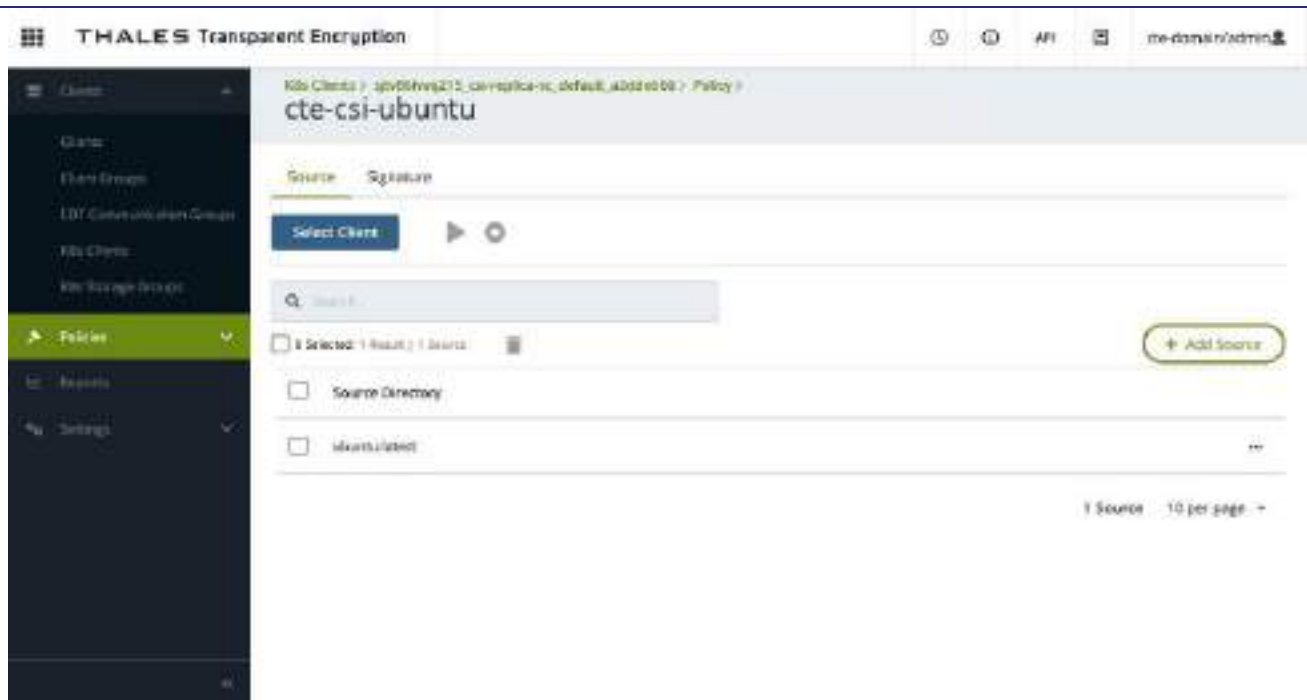
- A new container named `cte-csi-signer` is added to a controller pod. It will monitor CTE for Kubernetes storage classes for the following operations: `creation`, `deletion` and `updating`. It manages automatic registration as a signer service to CipherTrust Manager.

Workflow for obtaining a Signature Set on CipherTrust Manager and adding it to a Policy

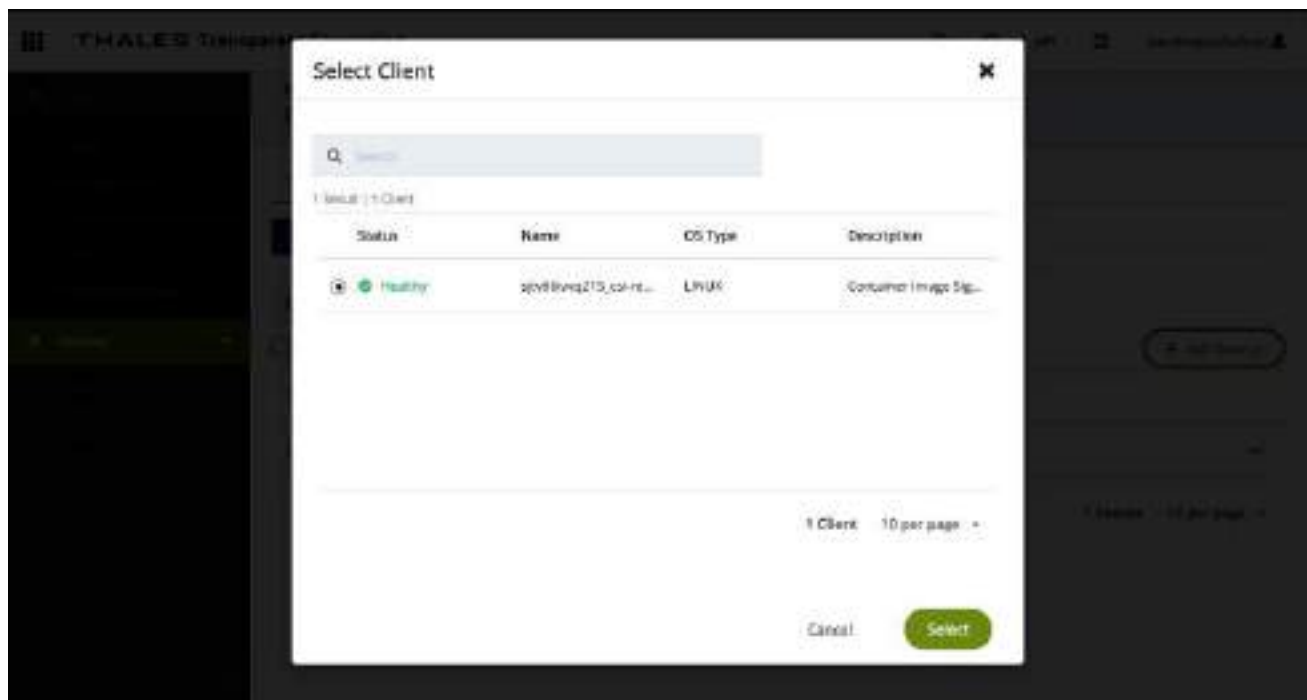
- After [signing the container image](#) for a specific storage class and namespace, it should display in CipherTrust Manager in the Kubernetes client.



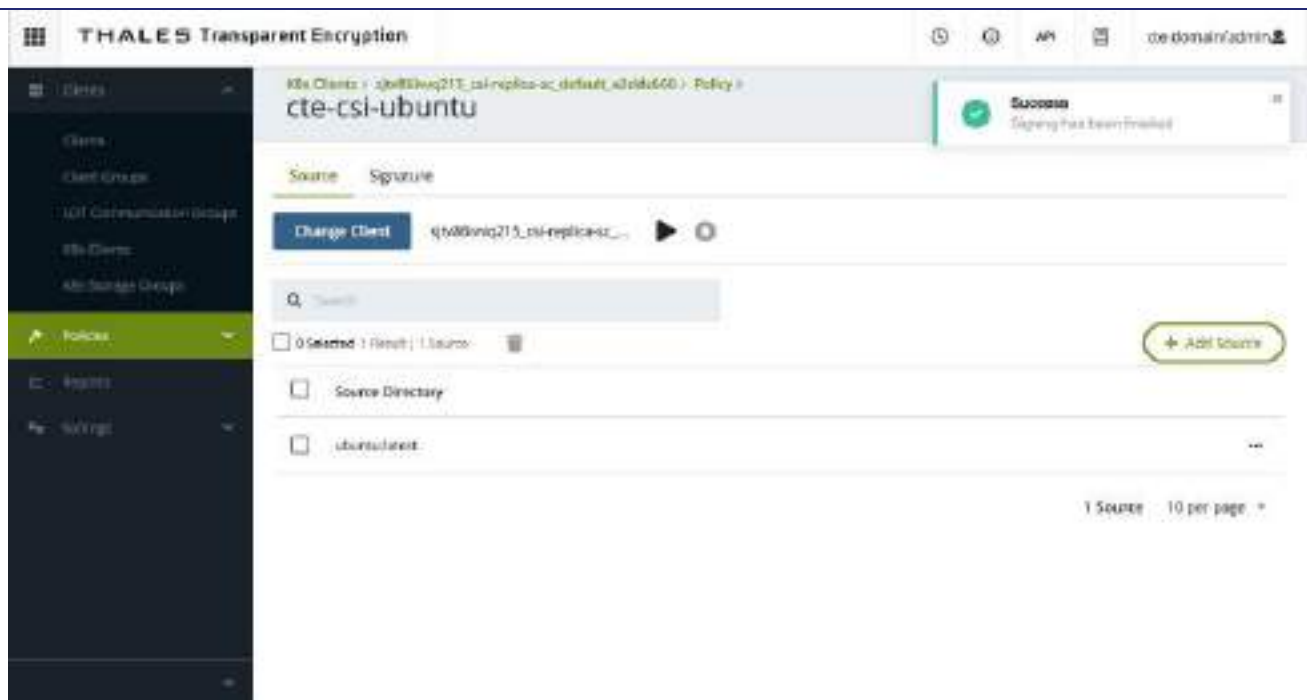
- Add a container image source and select a client for signing in the Signature Set page.



3. Selecting a client for signing.



4. Displays a message after a successful signing operation.



Creating a Signature Set on CipherTrust Manager and adding it to the Policy

Create the signature set in CipherTrust Manager:

1. See [Creating Signature Sets for Container Images](#). Follow those instructions for creating your Signature Set.
2. Add the following as a signature rule in the Kubernetes policy to enable Trusted Pod enforcement:
 - Signature set, which contains all of the signatures for all of the containers to which you want to have access

Note

Unlike normal CipherTrust Manager signature sets, CTE for Kubernetes signature sets **do not** attach to process sets.

Data Transformation

- Using Initial Data Transformation for CTE for Kubernetes
- Applying Data Transformation Rotation for CTE for Kubernetes
- Decryption using Data Transformation with CTE for Kubernetes

Using Initial Data Transformation for CTE for Kubernetes

The Data Transformation utility is an application that encrypts data-in-place in a GuardPoint. To perform dataxform for CTE for Kubernetes, two additional steps are required:

1. An Data Transformation GuardPolicy, along with a CTE for Kubernetes production policy, must be added to the CTE for Kubernetes storage group on CipherTrust Manager.
2. You must add an Data Transformation policy name in the CTE for Kubernetes claim against the PVC. The name of the claim is `cte-csi-claim.yaml`.

For the standard Data Transformation steps, see [Data Transformation](#) for more information.

Prerequisites

- Verify that you have valid backup files of the data to be encrypted.

Note

You will need to stop **ALL** access and services to the data being encrypted during part of this procedure. Access will **NOT** be restored until the encryption process is complete. Make sure that you plan for this outage and that users know the data will be inaccessible for some time.

Create a Storage Group

Prior to the deployment of any `yaml` files for registration, you must create a [CTE for Kubernetes Storage Group](#).

Create Policies

In CipherTrust Manager, you must create policies before you can transform the data. See [Creating Policies](#) for more information.

1. Create an **Data Transformation policy** with **Policy Type**: CTE for Kubernetes.
2. Toggle the **Data Transformation** button to **on**. This policy is only for encrypting the clear text data in persistent volumes where the policy is applied.

Create Policy ✕

1 General Info 2 Security Rules 3 Key Rules 4 Data Transformation 5 Confirmation

Review the provided policy details.

1 General Info
 Name: dataxform-policy
 Policy Type: CTE for Kubernetes
 Description:

2 Security Rules

Resource Set	User Set	Process Set	Action	Effect
*			key_op	permit,applykey

3 Key Rules

Resource Set	Current Key Name
	clear_key

4 Data Transformation Rules

Resource Set	Transformation Key Name
	key?

[Back](#) [Save](#)

3. Create a **CTE for Kubernetes Production** policy with **Policy Type**: CTE for Kubernetes.
4. Make sure that the Data Transformation toggle is set to **off**.
5. For the key rule, you must use the **same key** as in the Data Transformation policy.

Create Policy ✕

1 General Info 2 Security Rules 3 Key Rules 4 Confirmation

Review the provided policy details.

1 General Info
 Name: csi-prod-policy
 Policy Type: CTE for Kubernetes
 Description:

2 Security Rules

Resource Set	User Set	Process Set	Action	Effect
*			el_ops	permit,audit,applykey

3 Key Rules

Resource Set	Key Name
	key?

[Back](#) [Save](#)

Apply GuardPolicies

- Add both policies, that you just created in the previous section, to the K8s Storage Group.

Adding an Annotation for Data Transformation to a claim

- Add an annotation for the Data Transformation policy to your `yaml` claim file. For example:

cte-csi-claim.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cte-claim
  annotations:
    csi.cte.cpl.thalesgroup.com/dataxform_policy: <dataxformPolicyName>
    csi.cte.cpl.thalesgroup.com/policy: production-policy //This must match
    your CTE CSI Policy name.
    csi.cte.cpl.thalesgroup.com/source_pvc: nfs-test-claim // nfs source persistent
    volume claim
spec:
  storageClassName: <CHANGE to the storageclass name that you deployed. For example:
  e.g. csi-test-sc>
  accessModes:
    - ReadWriteMany
resources:
  requests:
    storage: 1Ki
```

Note

- The Data Transformation will start as soon as the protected pod is deployed using the above cte-claim. Check that the logs are updated or described in the Application Pod. You can check the status of Dataxform by describing the CTE-PVC with the following annotation:

```
csi.cte.cpl.thalesgroup.com/dataxform_status: <dxfs-status>
```

Where

```
<dxfs-status>: [running-<WorkerNodeName> | completed | invalidPolicy | failed]
```


Warning

- Do not delete the protected pod while Data Transformation is in progress.
- The `dataxform_auto_lock` file is created in the Guard Policy path. Do not edit/delete this file.

Applying Data Transformation Key Rotation to CTE for Kubernetes

Data encryption keys are the keys used to encrypt data in a Guard Policy. Key rotation is the process of changing the encryption key used to encrypt your Guard Policy data. Key Rotation is also called Rekeying. Changing Guard Policy encryption keys increases security. Rekeying with Data Transformation requires that you change the current key of the production policy on your Guard Policy to a new key.

Warning

All users and applications will be blocked from accessing the data until Data Transformation is finished executing. Make sure that you plan for this outage and that users know the data will be inaccessible for some time.

Performing a Key Rotation

Delete Pods and PVC

1. Delete all of the application pods which are using `cte-claim`:

```
kubect1 delete -f <applicationPodName>.yaml
```

Note

Users can also use the CTE PVC (`# kubect1 describe pvc <CTE-PVC>`) to find the list of all of the pods that are using a specific CTE PVC.

2. Delete the CTE PVC:

```
kubect1 delete -f cte-csi-claim.yaml
```

Create New Polices and Keys

Create a new Data Transformation and a new production **Policy**, with required **Key** rules on CipherTrust Manager.

Alternatively, (recommended), clone the original policies, (Data Transformation and Production policy) that you used to encrypt the data.

- If creating new policies, make sure to set **Policy Type: CTE for Kubernetes**.
- Toggle the Data Transformation button to **on** while creating the Data Transformation policy.
- Toggle the Data Transformation button to **off** while creating the production policy.

Updating Key Rules

Update the key rules, for both policies, by editing the Key Rule, and the Data Transformation Rule, and changing the Current Key Name and Transformation Key Name to reflect the new key to use.

For Example:

If the Guard Policy data is encrypted with Key1, and you want to rotate the key from Key1 to Key2:

Data Transformation clone/new policy key rule update:

```
Key Rules (Current Key Name): key1
Data Transformation Rules(Transformation Key Name): key2
```

Production clone/new policy key rule update:

```
Key Rules(Key Name): key2
```

Applying the New/Cloned Policies

1. Add the newly created/cloned policies to CipherTrust Manager K8s Storage Groups. See [Configuration Concepts](#) for more information about creating CTE for Kubernetes storage groups in CipherTrust Manager.
2. Update the data transformation and production policy names in the CTE claim yaml file (`cte-csi-claim.yaml`) with the following annotations.

```
csi.cte.cpl.thalesgroup.com/datatransform_policy: <NewDatatransformPolicy> // new
ly created datatransform policy
csi.cte.cpl.thalesgroup.com/policy: <NewProductionPolicy> // newl
y created production policy
```

Example

```
apiVersion: v1
```

```

kind: PersistentVolumeClaim
metadata:
  name: cte-claim
  annotations:
    csi.cte.cpl.thalesgroup.com/dataxform_policy: dataxform-key1-to-key2-rotation-policy // newly created dataxform standard policy
    csi.cte.cpl.thalesgroup.com/policy: csi-prod-with-key2-policy // newly created production policy
    csi.cte.cpl.thalesgroup.com/source_pvc: nfs-test-claim // nfs source persistent volume claim spec:
    storageClassName: <CHANGE ME to the storageclass name deployed e.g. csi-test-sc>
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Ki

```

3. Deploy the CTE claim file `cte-csi-claim.yaml`:

```
kubectl apply -f cte-csi-claim.yaml
```

4. Add the `dataxform_cleanup` annotation to the NFS source PV from `nfs-pv.yaml`. See [Create a Persistent Storage \(PV\) YAML file description](#) for more information.

```
kubectl annotate pv <SOURCE_PV_NAME> csi.cte.cpl.thalesgroup.com/dataxform_cleanup='require'
```

****<SOURCE_PV_NAME>**: PersistentVolume name from `nfs-pv.yaml` file**

5. Deploy the Application Pod which uses the `cte-claim`.

```
kubectl apply -f <applicationPodName>.yaml
```

Note

- When Data Transformation is running, the Guard Policy does not display as active on CipherTrust Manager, because it does not send any Guard Policy details to CipherTrust Manager. Only after the production policy is applied to the Guard Policy do the details populate CipherTrust Manager.
- The Data Transformation will start rekeying as soon as the protected pod is deployed. Check that the

logs are updated or described in the Application Pod.

Warning

- Do not delete the protected pod while Data Transformation is in progress.
- The `dataxform_auto_lock` file is created in the Guard Policy path. Do not edit/delete this file.

Decryption using Data transformation with CTE for Kubernetes

Guard Policy data can be unencrypted using Data Transformation.

The steps are same as with the section [Applying Data Transformation Rotation for CTE for Kubernetes](#), except for policy creation which requires specific key rules for decryption.

Create policies for decryption with required keys on CipherTrust Manager

1. Create new policies (Data Transformation standard policy and Production policy) with required key rules.
 - Alternatively, clone the original policies that you used to encrypt the data.
2. Update the key rules for both of the policies by editing the Key Rule, and the Data Transformation Rule, and changing the Current Key Name and Transformation Key Name to `clear_key`.

For Example:

If Guard Policy data is encrypted with Key2:

Data Transformation clone/new policy key rule update:

```
Key Rules (Current Key Name): Key2
Data Transformation Rules(Transformation Key Name): Clear_key
```

Production clone/new policy key rule update:

```
Key Rules(Key Name): Clear_key
```

3. Save and apply the policies.

Volume Management

- Volume Cloning Support
- Volume Snapshot and Restoring
- Volume Resizing and Expansion

Volume Cloning Support

A clone is an exact duplicate of an existing Kubernetes Volume. It can be consumed as any standard Volume would be consumed. A Volume clone allows for specifying existing PVCs in the `dataSource` field of the `.yaml` file to indicate that you want to clone a volume.

Considerations

- Cloning support is only available with dynamic volume provisioning. Cloning cannot occur on static/manual volumes.

Make sure that the storage provisioning driver that you use supports dynamic volume provisioning and volume cloning support. Check the storage driver list and respective storage driver documentation ([CSI Drivers doc](#)) to ensure support.

- You can only clone a PVC when it exists in the same namespace as the destination PVC.
- Newly created PVCs are independent objects. They can be consumed, cloned, snapshot, or deleted independently and without consideration for the original source PVC. This means that the source is not linked to the newly-created clone so it can be modified or deleted without affecting the original clone.
- The source CTE-PVC must be bound and available.

Note

Do not create a clone of a CTE-PVC if a Pod is using it. This may cause data corruption because the PVC is not quiesced. If the source CTE-PVC is in use, then you must take down all of the pods before creating the clone. This will cause some downtime.

Cloning an existing CTE-PVC

Prerequisites

- The source CTE-PVC that you want to clone contains encrypted data and is bound and available on a system using dynamic provisioning.

Provisioning a Storage Class

In the following example:

- It uses a supported NFS-CSI driver
- The application pod is running with the CTE-PVC (cte-claim)
- The claim is using the CTE-StorageClass (cte-test-sc) and is bound and available
- `cte-test-sc` is using `nfs-csi-sc-imm` (`nfs.csi.k8s.io`) source storageClass for NFS

cte-csi-storageClass.yaml (CTE-SC)

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cte-test-sc
provisioner: csi.cte.cpl.thalesgroup.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
allow_source_delete: false
parameters:
  # Domain name or IP address of the CipherTrust Manager (Required)
  key_manager_addr: 192.168.70.1

  # Name of the CipherTrust Manager K8s Storage Group (Required)
  k8_storage_group: k8s-group1

  # Kubernetes Secret with CM registration token (Required)
  registration_token_secret: cm-reg-token

  # To provision a volume dynamically, add the underlying source storageClass.
  # CTE-K8s driver creates source_pvc dynamically on this source storageClass.
  # A new volume will be provisioned by the respective underlying provisioner.
  # (Optional) (Required only for Dynamic PVC)
  # Taking an nfs provisioner driver (nfs.csi.k8s.io) example: nfs-csi-sc-imm
  source_storage_class: <underlying_source_sc_name>

  # When specified and set to "true", this parameter is used to automatically
  # delete the dynamically provisioned sourcePVC. This might delete the actual
  # data volume, depending
  # on the underlying provisioner driver implementation. Default is set as "false". (Optional)
  allow_source_pvc_delete: "false"

  # When specified, this parameter is used to automatically add the
  # csi.cte.cpl.thalesgroup.com/policy parameter to the CTE-K8s PVC based on
  # the requested parameters. (Optional) (Required if source_storage_class is set)
  default_policy: <policy_1>

  # Time in minutes to wait before unregistering from the CipherTrust Manager
  # once all volumes have been unguarded. Parameter must be added as a string
  # integer value. Default "10" minute. (Optional)
```

```
registration_period: "10"
```

PVC Claim Example

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cte-claim
  annotations:
    # If the following field is used, then this policy is used as the default_policy: <policy_1> in CTE-SC.
    csi.cte.cpl.thalesgroup.com/policy: policy_1
    # If the following field is used then the volume bounded with this pvc1 must be a dynamically provisioned volume, (optional)
    csi.cte.cpl.thalesgroup.com/source_pvc: <pvc1>
spec:
  storageClassName: cte-test-sc
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

Examples for PV, PVC, SC

kubectl get sc

```
:~/volumeSnapshot$ kubectl get sc,pvc,pv
NAME                                     PROVISIONER                                RECLAIMPOLICY
storageclass.storage.k8s.io/cte-test-sc  csi.cte.cpl.thalesgroup.com              Delete
Immediate                                true                                     13m
storageclass.storage.k8s.io/nfs-csi-imm  nfs.csi.k8s.io                           Delete
Immediate                                false                                    56d
```

Note

nfs-csi-sc-imm is the underlying source storageClass. Use in the CTE-SC (cte-csi-storageClass.yaml) parameter `source_storage_class: <underlying_source_sc_name>`

kubectl get pvc

```
NAME                                     CAPACITY  ACCESS MODES  STATUS  VOL
UME                                     STORAGECLASS
persistentvolumeclaim/cte-claim
c-b168e8d9-e26b-4515-aab2-91dc241ce74a  2Gi       RWX           Bound   pv
cte-test-sc
```

```
<unset>          13m
persistentvolumeclaim/cte-claim-unprotected-cprfnzjxjxg3gpyabrqh5d   Bound    pv
c-4183880c-6480-43e8-9564-b9f06fe557bf    2Gi      RWX      nfs-csi-imm
<unset>          13m
```

Note

cte-claim-unprotected-: name of the source PVC which is either created by the CTE-K8s driver if `csi.cte.cpl.thalesgroup.com/source_pvc: <pvc1>` is omitted in the source CTE-PVC, or any existing dynamically created source PVC used as `csi.cte.cpl.thalesgroup.com/source_pvc: cte-claim-unprotected-<UID#>` in the source CTE-PVC.

kubectl get pv

```
NAME                                     CAPACITY  ACCESS MOD
ES   RECLAIM POLICY   STATUS   CLAIM
STORAGECLASS VOLUMEATTRIBUTESCLASS  REASON  AGE
persistentvolume/pvc-4183880c-6480-43e8-9564-b9f06fe557bf    2Gi      RWX
Delete      Bound    default/cte-claim-unprotected-cprfnzjxjxg3gpyabrqh5d
nfs-csi-imm  <unset>          13m
persistentvolume/pvc-b168e8d9-e26b-4515-aab2-91dc241ce74a    2Gi      RWX
Delete      Bound    default/cte-claim
cte-test-sc  <unset>          13m
```

Verifying that the source CTE-PVC is valid to for cloning

1. Get the sourcePVC name which is attached to the source CTE-PVC, type:

```
kubectl get pv $(kubectl get pvc <Source_CTE-PVC_Name> -n <NameSpace> -o wide
--no-headers -o custom-columns=":spec.volumeName") -o wide --no-headers -o cus
tom-columns=":spec.csi.volumeAttributes" | awk '{for (i=1;i<=NF;i++){if ($i
~/csi.cte.cpl.thalesgroup.com\/source_pvc:/) {print $i}}}' | awk -F: '{print
$2}'
```

`<Source_CTE-PVC_Name>`: source CTE-PVC name. For example: `cte-claim`.

`<NameSpace>`: namespace in which the above CTE-PVC is deployed.

Example Response:

```
cte-claim-unprotected-<UID#>
```

2. Verify that the sourcePVC, if bounded with dynamically-provisioned volume, type:


```
kubectl get pv $(kubectl get pvc <SourcePVC_Name> -n <NameSpace> -o wide --no-headers -o custom-columns=":spec.volumeName") -o wide --no-headers -o custom-columns=":spec.csi"
```

- `<SourcePVC_Name>`: SourcePVC name obtained from previous command.
- `<NameSpace>`: namespace in which the source CTE-PVC is deployed.

Example Response

- If the source volume is provisioned-dynamically, then the output will contain data, for example:

```
map[driver:nfs.csi.k8s.io map[csi.storage.k8s.io/pv/name:pvc-caff4502-0123-4391-927a-5efa5cbb12d9 csi.storage.k8s.io/pvc/name:cte-claim-unprotected-mi5eclhwd4dyhjrjeqicbd csi.storage.k8s.io/pvc/namespace:default server:<some-IP> share:<share-path> storage.kubernetes.io/csiProvisionerIdentity:1724596308482-728-nfs.csi.k8s.io subdir:pvc-caff4502-0123-4391-927a-5efa5cbb12d9] volume Handle:<Some-IP>.#<share-path>#pvc-caff4502-0123-4391-927a-5efa5cbb12d9##]
```

- If the sourcePVC contains a volume which is not dynamically-provisioned then the output will not contain data.

Note

- If the sourcePVC contains a volume that is bounded but not dynamically-provisioned, then you cannot clone the source, CTE-PVC (for example: cte-claim).

Creating a Clone

To provision a volume clone:

1. Add a reference to an existing CTE Persistent Volume Claim (source CTE-PVC) in the same namespace of the data Source field of a new CTE Persistent Volume Claim (clone CTE-PVC).

cte-csi-claim-clone.yaml (clone CTE-PVC)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: clone-of-cte-claim
  annotations:
    csi.cte.cpl.thalesgroup.com/policy: policy_1
    # The following field is not required. A new sourcePVC will be created
```

by the cte-k8s driver and a new volume will be provisioned with cloned data.

```
# csi.cte.cpl.thalesgroup.com/source_pvc: <source-pvc>
spec:
  storageClassName: cte-test-sc
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

Note

- The namespace of the clone, CTE-PVC (`cte-csi-claim-clone.yaml`), must be the same as the namespace of the source CTE-PVC (`cte-csi-claim.yaml`)
- You must specify a capacity value for `spec.resources.requests.storage` with the clone CTE-PVC. The value specified must be the same or larger than the capacity of the source volume.
- The `csi.cte.cpl.thalesgroup.com/source_pvc` annotation in the clone CTE-PVC (`cte-csi-claim-clone.yaml`) is not required. A new sourcePVC (`source_pvc`) is created by the CTE for Kubernetes driver and a new volume is created with the cloned data.
- The Policy used with the clone CTE-PVC (`cte-csi-claim-clone.yaml`) must be the same, as the policy used with the source CTE-PVC (`cte-csi-claim.yaml`).
- The underlying storage driver provisioned/used with the source CTE-PVC and the clone CTE-PVC must be the same. Cloning with different storage provisioners is not supported.
- You can use the same, or a different, CTE-Storage Class (`spec.storageClassName`) with the clone CTE-PVC, but the underlying source Storage Class driver must be the same for the source CTE-PVC and the clone CTE-PVC.
- You can use the same, or a different, volume Access Mode (`spec.accessModes`) with the clone CTE-PVC.

2. Deploy the clone CTE-PVC, type

```
kubectl apply -f cte-csi-claim-clone.yaml
```

Based on the CTE storage class `volumeBindingMode`, and the underlying source storage class `volumeBindingMode`, CTE for Kubernetes provisions a new volume with cloned data as follows:

CTE Storage Class (<code>cte-test-sc</code>) (<code>volumeBindingMode</code>)	Source Storage Class (<code><underlying_source_sc_name></code>) (<code>volumeBindingMode</code>)	Comment
Immediate	Immediate	Provision volume with cloned data when deploying the clone CTE-PVC (<code>cte-csi-claim-clone.yaml</code>).
Immediate	WaitForFirstConsumer	CTE for Kubernetes does not support this <code>volumeBindingMode</code> combination.
WaitForFirstConsumer	Immediate	Provision volume with cloned data after deploying the clone CTE-PVC (<code>cte-csi-claim-clone.yaml</code>) and the Application Pod on this CTE-PVC.
WaitForFirstConsumer	WaitForFirstConsumer	Provision volume with cloned data after deploying the clone CTE-PVC (<code>cte-csi-claim-clone.yaml</code>) and the Application Pod on this CTE-PVC.

Volume Snapshot and Restoring

Introduction

Kubernetes volume snapshots provide Kubernetes users with the ability to copy a volume at a specific point in time. You can use this copy to restore a volume back to a prior state, or to provision a new volume. Use snapshots to protect your workloads and achieve business-critical recovery point objectives in your disaster recovery plan, or if you have compliance needs to periodically save data.

Similar to how API resources `PersistentVolume` and `PersistentVolumeClaim` are used to provision volumes for users and administrators, `VolumeSnapshotContent` and `VolumeSnapshot` API resources are provided to create volume snapshots for users and administrators.

Terminology

- `VolumeSnapshotClass`: Similar to how `StorageClass` provides a method for administrators to describe the classes of storage they offer when provisioning a volume, `VolumeSnapshotClass` provides a method for describing the classes of storage when provisioning a volume snapshot.
- `VolumeSnapshot`: User requests a snapshot of a volume. It is similar to `PersistentVolumeClaim`.
- `VolumeSnapshotContent`: Snapshot taken from a volume in the cluster that has been provisioned by an administrator. It is a resource in the cluster just like a `Persistent Volume` is a cluster resource.

Considerations

- The CSI storage drivers that support provisioning volume snapshots, the ability to provision dynamic volumes, and can be used with the CTE for Kubernetes Volume Snapshot and Restore feature are referenced on the [Kubernetes CSI Drivers List](#).
- API Objects: `VolumeSnapshot`, `VolumeSnapshotContent`, and `VolumeSnapshotClass` are Custom Resource Definitions (CRD). They are not part of the core API.

A CRD is an extension of the Kubernetes API that is not necessarily available in the default Kubernetes installation. It represents a customization of a particular Kubernetes installation. Without the required CRDs installed, attempts to create a `VolumeSnapshot` fail. Validate the CRDs by typing:

```
kubectl api-resources -o wide | grep -i VolumeSnapshot
```

If the snapshot CRD is installed and uses the snapshot API request, make sure that the volume snapshot controller is deployed. Snapshot CRDs may get installed by default by installing the respective storage snapshot controller. Alternatively, it may require installing the snapshot CRD along with the snapshot controller installation. Check with the appropriate storage driver documentation for installation instructions for snapshot CRDs and the snapshot controller.

Volume Snapshot Types

There are two methods for provisioning snapshots:

Pre-provisioned

A cluster administrator creates these `VolumeSnapshotContents`. They carry the details of the volume snapshot, on the storage system, which is available for use by cluster users. They exist in the Kubernetes API and are available for consumption.

Note

This method **IS NOT** supported by CTE for Kubernetes.

Dynamically provisioned

You can take a snapshot dynamically from a `PersistentVolumeClaim`. The `VolumeSnapshotClass` specifies the storage provider-specific parameters to use when taking a snapshot. You can create a snapshot for CTE volumes (CTE-PVC) and you can restore using the new CTE-PVCs.

Note

This method **IS** supported by CTE for Kubernetes.

Setup

The following example assumes that:

- CTE for Kubernetes setup is available
- It uses Dynamically Provisioned volumes
- It uses the compatible NFS-CSI driver
- The Application Pod is running with a CTE-PVC (`cte-claim-restore`)
- `cte-claim-restore` is using CTE-StorageClass (`cte-test-sc`) and is bound and available
- `cte-test-sc` is using the `nfs-csi-sc-imm` (`nfs.csi.k8s.io`) source storageClass for NFS

cte-csi-storageclass.yaml (CTE-SC)

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cte-test-sc
provisioner: csi.cte.cpl.thalesgroup.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
parameters:
  # Domain name or IP address of the CipherTrust Manager (Required)
  key_manager_addr: 192.168.70.1

  # Name of the CipherTrust Manager K8s Storage Group. (Required)
  k8_storage_group: k8s-group1

  # Kubernetes Secret with CipherTrust Manager registration token (Required)
  registration_token_secret: cm-reg-token

  # To provision a volume dynamically, add the underlying source storageClass.
  # The CTE-K8s driver will create a source_pvc dynamically on this source stor
ageClass.
  # A new volume will be provisioned by the underlying provisioner. (Optiona
l) (Required only for Dynamic PVC)
  # Taking an NFS provisioning driver (`nfs.csi.k8s.io`) example: `nfs-csi-sc-i
mm`
  source_storage_class: <underlying_source_sc_name>

  # When specified and set to "true", this parameter will automatically
```

```

    # delete the dynamic sourcePVC. This might delete the actual data volume, depending
    # on the underlying provisioner driver implementation. Default is set to "false". (Optional)
    allow_source_pvc_delete: "false"

    # When specified, this parameter will automatically add the `csi.cte.cpl.thalesgroup.com/policy`
    # parameter to the CTE-K8s PVC based on the requested parameters. (Optional) (Required if `source_storage_class` is set)
    default_policy: <policy_1>

    # Time in minutes to wait before unregistering from the CipherTrust Manager
    # once all volumes have been unguarded. Parameter must be added as a string
    # integer value. Default "10" minute. (Optional)
    registration_period: "10"

```

The following three parameters are specific to dynamic provisioned (sourcePVC and sourcePV) in the above CTE-K8s Storage Class definition:

- `source_storage_class`
- `default_policy`
- `allow_source_pvc_delete`

cte-csi-claim.yaml (source CTE-PVC)

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cte-claim-restore
  annotations:
    # If this field is used then this policy is used whether or not `default_policy` in CTE-SC is provided or not.
    csi.cte.cpl.thalesgroup.com/policy: policy_1
    # (optional) If this field is used then the volume bounded with this pvc1 must be a dynamically provisioned volume.
    #csi.cte.cpl.thalesgroup.com/source_pvc: <pvc1>
spec:
  storageClassName: cte-test-sc
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 5Gi

```

Examples Scripts for PV, PVC, SC with the deployments listed above

kubectl get sc

NAME	PROVISIONER	RECLAIMPO

```

LICY    VOLUMEBINDINGMODE    ALLOWVOLUMEEXPANSION    AGE
storageclass.storage.k8s.io/cte-test-sc    csi.cte.cpl.thalesgroup.com    Delete
Immediate    true    13m
storageclass.storage.k8s.io/nfs-csi-imm    nfs.csi.k8s.io    Delete
Immediate    false    56d

```

- **CTE-SC:** `cte-test-sc`
- **Underlying source SC:** `nfs-csi-sc-imm`

kubectl get pvc

```

NAME                                STATUS    VOL
UME                                STORAGECLASS
VOLUMEATTRIBUTESCLASS    AGE
persistentvolumeclaim/cte-claim-restore    Boun
d    pvc-b168e8d9-e26b-4515-aab2-91dc241ce74a    2Gi    RWX    cte-tes
t-sc    <unset>    13m
persistentvolumeclaim/cte-claim-restore-unprotected-cprfnzjxjxg3gpyabrqh5d    Boun
d    pvc-4183880c-6480-43e8-9564-b9f06fe557bf    2Gi    RWX    nfs-cs
i-imm    <unset>    13m

```

- **CTE-PVC:** `cte-claim-restore`
- **Source PVC:** `cte-claim-restore-unprotected-<UID>`

kubectl get pv

```

NAME                                CAPACITY    ACCESS MOD
ES    RECLAIM POLICY    STATUS    CLAIM    REASON    AGE
STORAGECLASS    VOLUMEATTRIBUTESCLASS
persistentvolume/pvc-4183880c-6480-43e8-9564-b9f06fe557bf    2Gi    RWX
Delete    Bound    default/cte-claim-restore-unprotected-cprfnzjxjxg3gpyab
rqh5d    nfs-csi-imm    <unset>    13m
persistentvolume/pvc-b168e8d9-e26b-4515-aab2-91dc241ce74a    2Gi    RWX
Delete    Bound    default/cte-claim-restore
cte-test-sc    <unset>    13m

```

- **CTE-PV:** `pvc-<UID>`
- **Source PV:** `pvc-<UID>`

Validate Source Setup for taking Snapshots

1. Get the `sourcePVC` name that is attached to the source CTE-PVC, type:

```

kubectl get pv $(kubectl get pvc <Source_CTE-PVC_Name> -n <NameSpace> -o wide
--no-headers -o custom-columns=":spec.volumeName") -o wide --no-headers -o cus

```

```
tom-columns=":spec.csi.volumeAttributes" | awk '{for (i=1;i<=NF;i++){if ($i ~ /csi.cte.cpl.thalesgroup.com\/source_pvc:/) {print $i}}}' | awk -F: '{print $2}'
```

- `<Source_CTE-PVC_Name>`: source CTE-PVC name, for example: `cte-claim-restore`
- `<NameSpace>`: namespace in which the above CTE-PVC is deployed

Response

```
cte-claim-unprotected-<UID>
```

2. Verify that the sourcePVC is bounded with a dynamically provisioned volume, type:

```
kubectl get pv $(kubectl get pvc <SourcePVC_Name> -n <NameSpace> -o wide --no-headers -o custom-columns=":spec.volumeName") -o wide --no-headers -o custom-columns=":spec.csi"
```

- `<SourcePVC_Name>`: SourcePVC name obtained from previously used command
- `<NameSpace>`: Namespace in which source CTE-PVC is deployed

Response

If the source volume is dynamically provisioned, then the output will contain data, for example:

```
map[driver:nfs.csi.k8s.io map[csi.storage.k8s.io/pv/name:pvc-caff4502-0123-4391-927a-5efa5cbb12d9 csi.storage.k8s.io/pvc/name:cte-claim-unprotected-mi5eclhw d4dyhjrjeqicbd csi.storage.k8s.io/pvc/namespace:default server:<some-IP> share:<share-path> storage.kubernetes.io/csiProvisionerIdentity:1724596308482-728-nfs.csi.k8s.io subdir:pvc-caff4502-0123-4391-927a-5efa5cbb12d9] volumeHandle:<Some-IP>.#<share-path>#pvc-caff4502-0123-4391-927a-5efa5cbb12d9##]
```

If the source volume **is not** dynamically provisioned, then the output will **not** contain data.

Note

If the sourcePVC has a bound volume which is not dynamically provisioned, then you cannot take a Volume Snapshot on the source CTE-PVC (i.e. `cte-claim-restore`).

Create a Snapshot

1. Create a `VolumeSnapshotClass` object to specify the CTE-K8s driver and `deletionPolicy` for your volume snapshot. You can reference `VolumeSnapshotClass` objects when you create `VolumeSnapshot` objects.

`cte-csi-snapshotclass.yaml`

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: cte-snapclass
  parameters:
    # Add the parameters if required for any storage specific. These parameters will be passed to the storage source snapshotClass by the CTE-K8s driver.
    <key1>: <value1>
    <key2>: <value2>
#The following field must be used as it is.
driver: csi.cte.cpl.thalesgroup.com
deletionPolicy: Delete
```

Volume snapshot classes have parameters that describe volume snapshots belonging to the volume snapshot class. Different parameters may be accepted depending on the specific storage driver. Users must provide those required parameters with CTE-SnapshotClass (`cte-csi-snapshotclass.yaml`).

****Sample Parameters****

The following are sample parameters that are used with different cloud storage provisioners. Check the specific storage driver documentation for the supported parameters based on the requirements.

```
parameters:
  snapshot-type: images
  image-family: <IMAGE_FAMILY>
  storage-locations: us-east2
  type: backup
  resourceGroup: <EXISTING_RESOURCE_GROUP_NAME>
  incremental: true
  # The driver field value is static. Do not change it.
  driver: csi.cte.cpl.thalesgroup.com
```

1. Apply the CTE-SnapshotClass manifest, type:

```
kubectl apply -f cte-csi-snapshotclass.yaml
```

2. Create a Volume Snapshot Object, which is a request for a snapshot of an existing `PersistentVolumeClaim` object.

```
cte-csi-snapshot.yaml
```

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: cte-claim-restore-snapshot
spec:
  volumeSnapshotClassName: cte-snapclass
  source:
    # CTE-PVC on which the user wants to take the snapshot
    persistentVolumeClaimName: cte-claim-restore
```

Note

The namespace of the Volume Snapshot (cte-claim-restore-snapshot) must be the same as the namespace of the source Persistent Volume Claim (cte-claim-restore). Creating a snapshot from volumeSnapshotContent **is not** supported with CTE for Kubernetes. You cannot reference volumeSnapshotContentName as a source (spec.source).

3. Apply the CTE-Snapshot manifest, type:

```
kubectl apply -f cte-csi-snapshot.yaml
```

Once the previous CTE-Snapshot manifest, (cte-csi-snapshot.yaml) deploys, the CTE-K8s driver creates a source Snapshot Class for the underlying storage driver. CTE for Kubernetes also creates a source Snapshot on the sourcePVC. Then, the underlying storage driver (for example: `nfs.csi.k8s.io`) takes a snapshot on a physical volume.

Examples

Reference the following examples for VolumeSnapshotClass (vsc), VolumeSnapshot (vs), and VolumeSnapshotContent (vsc) from the deployments listed in the previous section.

kubectl get vsc

```
NAME
DRIVER                                DELETIONPOLICY  AGE
volumesnapshotclass.snapshot.storage.k8s.io/cte-snapclass
csi.cte.cpl.thalesgroup.com          Delete          2m53s
volumesnapshotclass.snapshot.storage.k8s.io/cte-snapclass-sourcesnapshotclass-f61
0823c  nfs.csi.k8s.io          Delete          2m49s
```

- CTE-Snapshot Class: cte-snapclass

- Source Snapshot Class (underlying storage snapshotClass): `cte-snapclass-sourcesnapshotclass-<UID>` // This is created by CTE-K8s driver based on input provided in CTE-SnapshotClass (`cte-csi-snapshotclass.yaml`) and the source CTE-PVC provided in CTE-Snapshot (`cte-csi-snapshot.yaml`)

kubecttl get vsc

```

NAME
READYTOUSE    RESTORESIZE    DELETIONPOLICY    DRIVER    VOLUMES
NAPSHOTCLASS    VOLUMESNAPSHOT
VOLUMESNAPSHOTNAMESPACE    AGE
volumesnapshotcontent.snapshot.storage.k8s.io/snapcontent-67a7d12c-67f3-472f-95e
e-28f0f73d4a5c    true    1966888    Delete    nfs.csi.k8s.io
cte-snapclass-sourcesnapshotclass-f610823c    cte-claim-restore-snapshot-sourcesna
pshot-f610823c    default    2m49s
volumesnapshotcontent.snapshot.storage.k8s.io/snapcontent-f610823c-b567-42b9-955
8-d59c9487ad7b    true    0    Delete    csi.cte.cpl.thalesgr
oup.com    cte-snapclass    cte-claim-restore-snapshot
default    2m49s

```

- CTE-SnapshotContent: `snapcontent-<UID>`
- Source SnapshotContent: `snapcontent-<UID>`

kubecttl get vs

```

NAME
READYTOUSE    SOURCEPVC    SOURCESNAPSHOTCONTENT
RESTORESIZE    SNAPSHOTCLASS    SNAPSHOTCONTENT
CREATIONTIME    AGE
volumesnapshot.snapshot.storage.k8s.io/cte-claim-restore-snapshot
true    cte-claim-restore
0    cte-snapclass    snapcontent-f610823c-b
567-42b9-9558-d59c9487ad7b    2m28s    2m49s
volumesnapshot.snapshot.storage.k8s.io/cte-claim-restore-snapshot-sourcesnapshot-
f610823c    true    cte-claim-restore-unprotected-cprfnzjxxjxg3gpyabrqh5d
1966888    cte-snapclass-sourcesnapshotclass-f610823c    snapcontent-67a7d12c-6
7f3-472f-95ee-28f0f73d4a5c    2m29s    2m49s

```

- CTE-Snapshot: `cte-claim-restore-snapshot` // CTE for Kubernetes does not take any actual snapshot, that is why RESTORE SIZE is 0 for CTE-Snapshot
- Source Snapshot: `cte-claim-restore-snapshot-sourcesnapshot-<UID>` // Actual snapshot data is created by the underlying source driver (for example: `nfs.csi.k8s.io`)

Validate the Snapshot

Before restoring the volume from the snapshot, make sure that the snapshot is complete and ready to use. The snapshot status `READY_TO_USE` should be set to `true`. This indicates that the snapshot is successful. To verify, type:

```
kubectl get volumesnapshot -n <Namespace> -o custom-columns='NAME:.metadata.name,READY:.status.readyToUse'
```

- `<Namespace>`: Namespace in which the snapshot is deployed.

Verifying Snapshot Name

User can describe the snapshot to check the status:, type:

```
kubectl describe vs -n <Namespace> <CTE_Snapshot_Name>
```

- `<Namespace>`: Namespace in which the snapshot is deployed
- `<CTE_Snapshot_Name>`: CTE-Snapshot name, for example `cte-claim-restore-snapshot`

Restore the Volume Snapshot

You can reference a `VolumeSnapshot` in a `PersistentVolumeClaim` to provision a new volume with data from an existing volume, or restore a volume to a state that you captured in the snapshot. To reference a `VolumeSnapshot` in a `PersistentVolumeClaim`, add the `dataSource` field to your `PersistentVolumeClaim`.

Restore CTE-PVC (`cte-claim-restore.yaml`)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cte-claim-restore
  annotations:
    # Make sure that the policy is correct. The policy key must be used correctly. Check the policy
    # that is used the source CTE-PVC (i.e. cte-claim-restore)
    csi.cte.cpl.thalesgroup.com/policy: <policy_1>
    # The 'csi.cte.cpl.thalesgroup.com/source_pvc' annotation is not required. A new sourcePVC is
    # created by the cte-k8s driver and a new volume is provisioned with restored data
    #csi.cte.cpl.thalesgroup.com/source_pvc: <source-pvc>
spec:
  storageClassName: cte-test-sc
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
  dataSource:
    kind: VolumeSnapshot
    # Name of the CTE-Snapshot to which the user wants to restore the data
    name: cte-claim-snapshot
    apiGroup: snapshot.storage.k8s.io
```

Notes and Considerations

- The namespace of the restore `PersistentVolumeClaim` **must** be the same as the namespace of the `VolumeSnapshot`.
- Make sure to use the correct policy (`csi.cte.cpl.thalesgroup.com/policy: <policy_1>`) with the restore CTE-PVC (`cte-csi-claim-restore.yaml`). The snapshot is taken on encrypted data, and after restoring the data, the correct key must be provided with the policy in order to decrypt and access the data. If the Key rules are not correct with the policy, then the data will be corrupted while accessing it. Check the source CTE-PVC (`cte-csi-claim.yaml`) policy for key information.
- Make sure that the restore CTE-PVC (`cte-csi-claim-restore.yaml`) is using the same underlying source storage driver as the source CTE-PVC (`cte-csi-claim.yaml`). Restoring a volume with a different source storage driver is not supported. For example, if the source CTE-PVC is using `pd.csi.storage.gke.io` as the underlying source driver and the snapshot was created with it, then restoring the volume with a different storage driver will fail to restore the volume.
- You can create/restore unlimited CTE-PVC's from volume snapshot, as long as the snapshot is available.

Apply the Manifest

Use the following `.yaml` script as an example when restoring a Volume Snapshot.

```
kubect1 apply -f cte-csi-claim-restore.yaml
```

Based on the CTE Storage Class `volumeBindingMode` and the underlying source Storage Class `volumeBindingMode`, a new volume is provisioned with restored data as follows:

CTE Storage Class (<code>cte-test-sc</code>) (<code>volumeBindingMode</code>)	Source Storage Class (<code><underlying_source_sc_name></code>) (<code>volumeBindingMode</code>)	Comment
Immediate	Immediate	Provision volume with restored data when deploying the restored CTE-PVC (<code>cte-claim-restore.yaml</code>).
Immediate	WaitForFirstConsumer	CTE for Kubernetes does not support this <code>volumeBindingMode</code> combination.
WaitForFirstConsumer	Immediate	Provision volume with restored data after deploying the restore CTE-PVC (<code>cte-claim-restore.yaml</code>) and the Application Pod on this CTE-PVC.
WaitForFirstConsumer	WaitForFirstConsumer	Provision volume with restored data after deploying the restore CTE-

CTE Storage Class (<code>cte-test-sc</code>) (<code>volumeBindingMode</code>)	Source Storage Class (<code><underlying_source_sc_name></code>) (<code>volumeBindingMode</code>)	Comment
		PVC (<code>cte-claim-restore.yaml</code>) and the Application Pod on this CTE-PVC.

Volume Resizing and Expansion

Overview

You can now resize an existing volume by editing the `PersistentVolumeClaim` (PVC) object. You no longer have to manually interact with the storage backend, or delete and recreate PV and PVC objects, to increase the size of a volume. Kubernetes will automatically expand the volume, using storage backend, and it will also expand the underlying file system in-use by the Pod, without requiring any downtime if the underlying storage provisioner can support it.

Prerequisites

- Before performing an expansion, make sure that the source volume (PV) is provisioned dynamically, by adding the underlying source `storageClass`. If not, volume expansion fails.
- Make sure that underlying storage driver allows for volume expansion. Check the respective storage provider's documentation to verify.

Method

Perform volume expansion by editing the existing `PersistentVolumeClaim` (PVC) object. You can expand the volume **online** or **offline** based on the specific storage driver support.

Online

Expands a PVC while the PVC is still in-use by the Pod.

- Pods referencing the resized volume do not need to be restarted. Instead, the file system will automatically be resized while in use as part of volume expansion.

Offline

Expanding a PVC which is not in-use by any Pod.

- If the underlying storage driver can only support offline expansion, users must take down their Pod before expanding the volume.

Note

- File system expansion does not happen until a pod references the resized volume, so if no pods referencing the volume are running file system expansion does not happen.
- Some storage drivers support Online expansion, some do not. Refer to the [Storage driver list](#) for your storage provider information.

Expanding the CTE-PVC Volume

You can use either command: `kubectl patch` or `kubectl edit` to expand the volume size.

1. Using the `kubectl patch` command:

```
kubectl patch pvc <CTE-PVC_Name> -n <Namespace> --type merge -p '{"spec":{"resources":{"requests":{"storage":"<New_Size>"}}}}'
```

Example

This example requests expansion to 5G:

```
kubectl patch pvc cte-claim -n ns1 --type merge -p '{"spec":{"resources":{"requests":{"storage":"5Gi"}}}}'
```

2. Using the `kubectl edit` command:

- a. Type:

```
kubectl edit pvc -n
```

Example

```
kubectl edit pvc cte-claim -n ns1
```

- a. Update the **resource > request > storage field** with the required size, for example 5Gi.

Validating volume expansion

The duration for volume expansion depends on the size of the new volume requested and the type of storage driver provisioner.

- You can verify it by viewing the `cte-csi-controller-xxx` log:

```
kubectl logs cte-csi-controller-6f669d4b94-8xdcl -n kube-system cte-csi
```

- For more details, describe the CTE-PVC and sourcePVC. To verify this information, type:

```
kubectl describe pvc <pvc_name> -n <Namespace>

kubectl get pvc <pvc_name> -n <Namespace> -o yaml
```

- Watch the `status.conditions` field of the CTE-PVC, and sourcePVC, to see if the resize has completed.
- Refer to the storage driver documentation for more specific details.

Unsupported Functions

- Shrinking persistent volumes is not supported.
- Only dynamically provisioned PVCs whose storage class has the `allowVolumeExpansion` parameter set to `true` are expandable.

Advanced Configuration and Troubleshooting

- [Using external certificates with CTE for Kubernetes](#)
- [Troubleshooting](#)

Using External Certificates with CTE for Kubernetes

You can use External Certificates for communication between CTE for Kubernetes and CipherTrust Manager. This is optional.

Note

Install external certificate before registering CTE for Kubernetes with CipherTrust Manager.

Overview

CipherTrust Transparent Encryption can now use an external certificate, available at a user-defined path, to communicate with CipherTrust Manager.

Prerequisites

The external certificate must be:

- On the file system
- In PEM format

A key pair must already exist for the client:

- Must have Encryption type of either:
 - sha256WithRSAEncryption
 - ecdsa-with-SHA384
- Must be Encrypted with a pass phrase

Initial setup

1. Obtain your external CA certificate.
2. Create a certificate using the external CA certificate and key.

CipherTrust Manager Setup

To setup CipherTrust Manager to communicate through an external certificate:

1. Import the CA certificate into the CipherTrust Manager, click **CA > External > Add External CA**.
2. In the **Add External CA** dialog, copy and paste the `<ca_certificate_name>.pem` file content from the UI page and provide a user-friendly name.

For more information, see [Using an Externally Generated Server Certificate for an Interface](#)

3. Add the CA certificate to the list of trusted sources for the web interface, click **Admin Settings > Interfaces > web > Edit > External Trusted CAs**.
4. Restart the web server, click **Admin Settings > Services > web > Restart**.
5. Create a registration token for the CTE agent. See [Creating a Registration Token](#) for more information.

Note

After adding an external certificate to CipherTrust Manager, you must associate the client profile with the registration token so that it is updated to use the external CA.

CTE for Kubernetes Setup for Client Registration

1. Add the Registration token to the configuration file `cte-csi-regtoken.yaml` with the

`registration_token` parameter.

```
apiVersion: v1
kind: Secret
metadata:
  name: cm-reg-token
type: Opaque
data:
# This is a base64 encoded registration token. To generate:
echo -n <CM REGISTRATION TOKEN STRING> | base64 -w 0
  cm-reg-token: <registration token in base64 format>
```

2. Create a Kubernetes secret(s) with client certificate details:

```
kubectl create secret generic <client-secret-name> --from-file=<clientName>.cr
t=client_cert.pem --from-file=<clientName>.key=client_key.pem --from-file=<cli
entName>.passphrase=<passphrase>
```

where:

- * `<client-secret-name>` = unique name for secret
- * `<clientName>` = unique name for client
- * `<passphrase>` = passphrase for client

Example

```
kubectl create secret generic <client-secret-name> --from-file=client1.crt=cli
ent_cert.pem --from-file=client1.key=client_key.pem --from-file=client1.passph
rase=passphrase
```

You can also use files from a **specific path** as a Partial or Absolute path:

```
kubectl create secret generic <client-secret-name> --from-file=<clientName>.cr
t=./client1/client_cert.pem --from-file=client1.key=./client1/client_key.pem
--from-file=client1.passphrase=./client1/passphrase
```

To create multiple client certificate details (certificate, key and passphrase) with the secret:

```
kubectl create secret generic <client-secret-name> --from-file=<client1Name>.c
rt=<client1_cert.pem> --from-file=<client1Name>.key=<client1_key.pem> --from-f
ile=<client1Name>.passphrase=<client1-passphrase> --from-file=<client2Name>.c
rt=<client2_cert.pem> --from-file=<client2Name>.key=<client2_key.pem> --from-f
ile=<client2Name>.passphrase=<client2-passphrase>
```

where:

- * `<client-secret-name>` = unique name for secret
- * `<clientName>` = unique name for client
- * `<passphrase>` = passphrase for client

Example

```
kubectl create secret generic client1-client2-secret --from-file=client1.crt=client1_cert.pem --from-file=client1.key=client1_key.pem --from-file=client1.passphrase=client1-passphrase --from-file=client2.crt=client2_cert.pem --from-file=client2.key=client2_key.pem --from-file=client2.passphrase=client2-passphrase
```

If certificates are on **different paths**, provide the filename with the path (Partial or Absolute path)

```
kubectl create secret generic client1-client2-secret --from-file=client1.crt=./client1/client_cert.pem --from-file=client1.key=./client1/client_key.pem --from-file=client1.passphrase=./client1/passphrase --from-file=client2.crt=./client2/client_cert.pem --from-file=client2.key=./client2/client_key.pem --from-file=client2.passphrase=./client2/passphrase
```

Note

There must be an additional client certificate detail added to the secret parameter `external_ca_client_secret: <client-secret-name>`. This additional detail is required for each storage class deployment. This is because there is an additional requirement for Signer Registration for each storage class. This Signer Registration is an additional default registration).

3. Create the storage-class yaml file with a client secret name:

While creating the storage-class configuration file `cte-storageclass.yaml` add the following parameter:

```
external_ca_client_secret: <client-secret-name>
```

Also make sure that the registration token secret name from the `cte-csi-regtoken.yaml` is correct and contains the following parameter:

```
registration_token_secret: cm-reg-token
```

cte-storageclass.yaml

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-test-sc
provisioner: csi.cte.cpl.thalesgroup.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
parameters:
  # Domain name or IP address of the CipherTrust Manager (Required)
  key_manager_addr: 192.168.70.1
  # Name of the CipherTrust Manager K8s Storage Group. (Required)
  k8_storage_group: test-group
  # Kubernetes Secret with CipherTrust Manager registration token (Required)
  registration_token_secret: cm-reg-Token
  # Kubernetes secret with External CA signed client certificate (Optional)
  external_ca_client_secret: <client-secret-name>
  # Description will be displayed in CipherTrust Manager (Optional)
  client_description: "K8s client"
  # Time in minutes to wait before unregistering from the CipherTrust Manager. Default is 10 minutes.
  # Once all volumes have been unguarded. Parameter must be added as a string integer value. (Optional)
  registration_period: "10"

```

Note

- Each storage-class will have its own client secret with a client certificate, key and passphrase details.
- If one storage-class is being used over multiple worker nodes, then the client secret will have multiple client details (certificate, key and passphrase)

Troubleshooting

This section contains the following troubleshooting topics:

Inspecting Deployment

Deployment of CTE for Kubernetes typically uses a single controller pod and at least one worker pod per Kubernetes node. Inspect the deployment using `kubectl get` commands to access basic output.

Example

Get the CTE for Kubernetes pod names that were deployed on the Kubernetes cluster, type:

```
NAMESPACE="kube-system"
kubectl get pods -n ${NAMESPACE} -o wide | grep cte-csi
```

Response

```
cte-csi-controller-0      5/5      Running    0         25h      192.168.77.182    node
1.example.com            <none>    <none>
cte-csi-node-28srz        4/4      Running    0         25h      192.168.77.181    node
1.example.com            <none>    <none>
cte-csi-node-cjzzn        4/4      Running    0         25h      192.168.76.68     contro
1.example.com            <none>    <none>
```

Getting the Agent version

The version of CTE for Kubernetes that you are running displays in the log files for each pod:

```
NAMESPACE="kube-system"
for podName in `kubectl get pods -n ${NAMESPACE} | grep cte-csi-[node,controller] | cut -d " " -f1`; do
    echo -n "${podName}: "
    kubectl logs -n ${NAMESPACE} ${podName} cte-csi | grep "Version:"
done
```

Response

```
cte-csi-controller-7d7d8bd46b-wbffd: I0308 17:25:13.608929 1 cte.go:139] Version: 1.6.0.xx
cte-csi-node-8qx5q: I0308 17:25:11.643673 2218968 cte.go:139] Version: 1.6.0.xx
cte-csi-node-h8q8q: I0308 17:25:13.465795 2205645 cte.go:139] Version: 1.6.0.xx
```

Inspecting Events

CTE for Kubernetes relies on Kubernetes event infrastructure for diagnostics of problems with a CTE for Kubernetes volume. Any errors in attaching a volume to a pod will display in the event logs. Look at all of the events in a Kubernetes namespace by typing:

```
NAMESPACE="kube-system"
kubectl get event -n ${NAMESPACE}
```

To target events of a specific pod, type:

```
PODNAME="my_application_pod"
NAMESPACE="kube-system"
kubectl get event -n ${NAMESPACE} --field-selector involvedObject.name=${PODNAME}
```

Inspecting logs

Examining logs for CTE for Kubernetes could offer supplementary insights that might not be conveyed through Kubernetes Events.

Deployment of CTE for Kubernetes is split between two different types of pods:

- **Controller Server:** Manages the volume manipulation (Dynamic Volume Provisioning, Volume Cloning, Volume Expansion, Volume Snapshot & Restore) activities. Denoted by a `cte-csi-controller-X` pod name `cte-csi-controller-X` pod name.
- **Node server:** Manages attaching CTE for Kubernetes volumes to pods. Denoted by a `cte-csi-node-XXXXX` pod name.

Inspecting the Controller Server Log

The logs for the CTE for Kubernetes controller server pod are distributed across two containers:

- The `cte-csi` server contains logs relevant to dynamic provisioning for CTE for Kubernetes persistent volumes, volume management (Volume cloning, Volume Resizing, Volume Snapshot & Restore, and Volume affinity).
- The `cte-csi-signer` contains logs relevant to the signer client. (The Signer client is used in the pod attestation feature). Node activities (node addition, deletion) for the feature is set to automatically cleanup for CTE clients.

Use the two relevant `kubectl` logs commands to view the logs, type:

```

NAMESPACE="kube-system"
kubectl logs -n ${NAMESPACE} cte-csi-controller-0 cte-csi
kubectl logs -n ${NAMESPACE} cte-csi-controller-0 cte-csi-signer

```

Inspecting the Node Server Log

The logs for the CTE for Kubernetes pods are distributed across two containers:

- The `cte-csi` container contains logs relevant to CTE for Kubernetes activity, encompassing details like volume mounting and registration
- The `cte-agent-logs` contain logs for the CTE encryption engine agent, VMD, CipherTrust Manager server communication logs, and file access logs.

Inspecting the Node Server requires first identifying which Kubernetes node the application pod is scheduled on. Once that node has been identified, find the CTE for Kubernetes node server running on that node.

After you obtain the pod name, use the two relevant `kubectl logs` commands to view the logs, type:

```

NAMESPACE="kube-system"
kubectl logs -n ${NAMESPACE} cte-csi-node-XXXXX cte-csi

```

```
kubectl logs -n ${NAMESPACE} cte-csi-node-XXXXX cte-agent-logs
```

Updating CTE-K8s Log level on a live CTE deployment

The CTE-K8s driver logging is useful to debug the kubernetes resources for CTE, and activity related to volumes, registration, mount, etc.

It currently has a log level of <1-5>. The default log level is set to 1, which provides minimal logging. You can set the Log Level from 1-5. Set the log level to 5 for the maximum debug logging information.

In CTE for Kubernetes v1.6.0 and subsequent versions, you can change this log level on a live CTE for Kubernetes deployment, without disrupting any running deployment. When you install CTE for Kubernetes, it creates a default ConfigMap, `cte-k8s-config` in the same namespace where the CTE-K8s driver/agent is installed.

By default CTE for Kubernetes is installed in the `kube-system` namespace. The namespace could be different, based on the CTE for Kubernetes installation command.

To verify the CTE for Kubernetes installation namespace, type:

```
kubectl get pods --all-namespaces -o wide | grep cte-csi-controller
```

In the ConfigMap, `cte-k8s-config`, the `log_level` parameter reports the CTE-K8s Log Level. You can update the `log_level` to change the CTE-K8s log level.

Updating the CTE-CTE-K8s Log Level

You can update the log level parameter with the `kubectl patch` or `kubectl edit` command:

To update with the `kubectl patch` command, type:

```
kubectl patch cm cte-k8s-config -n <CTE-K8s-Namespace> --type merge --patch '{ "data": { "log_level": "<LEVEL>" } }'
```

Where:

- `<CTE-K8s-Namespace>`: Namespace in which CTE for Kubernetes is installed.
- `<LEVEL>`: Log level: 1-5; 1 is for minimal logs and 5 for maximum logs.

Example:

```
kubectl patch cm cte-k8s-config -n kube-system --type merge --patch '{ "data": { "log_level": "5" } }'
```

To update with the `kubectl edit` command, type:

```
kubectl edit cm cte-k8s-config -n <CTE-K8s-Namespace>
```

- `<CTE-K8s-Namespace>`: Namespace in which CTE for Kubernetes is installed.

Note

Update the `log_level` value accordingly.

Problems with Registration

CTE for Kubernetes automatically registers to CipherTrust Manager based on demand for volumes on a node. Failure to register is typically due to the registration token being either invalid or the token has no more client capacity. These types of errors are reported back through Kubernetes so analyzing the log files of the troubled pod reveals the registration failure message seen by the agent.

Troubleshooting Trusted Pods failures

The following are two examples of trusted pod failures:

Example 1

In `cte-csi-node` logs, the following error indicates that a running pod digest was not found in any signature set attached to a security policy.

```
E0310 06:41:16.614410 1687839 server.go:106] GRPC error: rpc error: code = Internal desc = Pod did not pass attestation checks: rpc error: code = Internal desc = Found no signature set for container ubuntu with digest 2adf22367284330af9f832ffefb717c78239f6251d9d0f58de50b86229ed1427
```

Example 2

In `cte-csi-node` logs, the following error indicates that running pod digests cannot be matched to the same signature set. There were two containers (ubuntu and ubuntu2) with digests included in different signature sets. Partial matches are displayed to help with troubleshooting.

```
E0407 09:07:33.393609 1492044 server.go:106] GRPC error: rpc error: code = Internal desc = Pod did not pass attestation checks: rpc error: code = Internal desc = Pod attestation failed. Unable to match all pod digests to same signature set. Partial matches:
Signature set policy-sigset2: [ubuntu:8ae9bafbb64f63a50caab98fd3a5e37b3eb837a3e0780b78e5218e63193961f9]
Signature set policy-sigset3: [ubuntu2:69665d02cb32192e52e07644d76bc6f25abeb5410edc1c7a81a10ba3f0efb90a]
```

Backing up Databases after Encryption

After encrypting a database, CipherTrust Transparent Encryption cannot make a backup of the database. Both scheduled and manual backup fail. The problem was the user's policy. A policy used in this scenario must follow a few rules.

With a CBC_CS1 key, a guarded file is modified to have a 4096 byte header holding key information. When an **Apply Key** effect is specified, the CipherTrust Transparent Encryption code adjusts the length and file offset for

this header. Without an **Apply Key** effect, the size and access of the offset include the CBC_CS1 header.

Thales recommends that you modify the first rule of your policy. Remove the action entry for `f_rd_att` from the first rule and add a new rule before it:

```
**action**: f_rd_att

**effect**: Permit, Apply Key
```

Policy processing starts with the first rule and continues until a matching rule is found. The effect for the matching rule is then applied.

For the `f_rd_att` action, this results in the SECFS code including the CBC_CS1 key header and adjusts the file size value. Without the Apply Key effect, the file size includes the CBC_CS1 header size and the file appears as 4096 bytes larger than its real size.

FSGroupID is not working with NFS shared storage volume

The `fsgroup` ID Security Context option allows an administrator to change the permissions of volumes before a pod starts. Some users have found that when adding the `fsgroup` ID to the SecurityContext section in the pod yaml file, it does not work as expected in the NFS storage environment.

The reason is that the `fsgroup` ID Security Context option is not supported with NFS volumes. It is only supported with local storage. This is a limitation from Kubernetes and not an issue with the CTE-U fuse driver.

- See [Configure a Security Context for a Pod or Container](#) in the Kubernetes documentation for more information.

Kubernetes Node reboot procedure

If you need to reboot a Kubernetes node, you must follow the proper order for commands:

1. To stop scheduling new pods on a node, type:

```
kubectl cordon ${NODE_NAME}
```

2. To stop all running pods in the node, type:

```
kubectl drain --ignore-daemonsets --force ${NODE_NAME}
```

Note

Does not stop the DaemonSet. It allows CTE-K8s nodes to complete deregistration.

3. Check to see if a finalizer exists for that node, type:

```
kubect1 get storageclass ${CTE_STORAGE_CLASS} -o jsonpath='{.metadata.finalizers}' | grep ${NODE_NAME}
```

4. Finding a finalizer with the node name on a StorageClass is an indication that the node is still registered with CipherTrust Manager. CTE-K8s may need to wait for the deregistration period for a complete deregistration to occur. Once this command returns nothing, you can reboot, type:

```
reboot
```

5. After rebooting, you can resume scheduling the pods on a node, type:

```
kubect1 uncordon ${NODE_NAME}
```



CTE for Kubernetes

Version: v.1.7.0 (latest)

30 January 2026

Contents

- CTE for Kubernetes Integrations and Solutions Guides 3
 - Integrating CTE for Kubernetes with Rancher Kubernetes Management Platform 3
 - Integrating CTE for Kubernetes with Tanzu Kubernetes Grid..... 6

CTE for Kubernetes Integrations and Solutions Guides

- [Integrating CTE for Kubernetes with Rancher Kubernetes Management Platform](#)
- [Integrating CTE for Kubernetes with VMware Tanzu Kubernetes Grid](#)

Integrating CTE for Kubernetes with Rancher Kubernetes Management Platform

Rancher is an open source software platform that enables Enterprise Kubernetes Container Management. It provides a CNCF-certified (Cloud Native Computing Foundation) Kubernetes distribution that can run on any Kubernetes cluster or distribution. It solves common Kubernetes installation complexities by removing most host dependencies, thereby presenting a stable path for deployment, upgrades & rollbacks.

This document describes installing Rancher and deploying a few workloads to a Kubernetes cluster provisioned by Rancher.

Test Environment

- **CTE for Kubernetes Agent:** 1.2.0
- **CipherTrust Manager:** 2.12.0
- **OS:** Ubuntu 20.04
- **Rancher:** 1.5.1
- **Kubernetes cert-manager:** 1.5.1

Prerequisite

Create a 3-node Kubernetes cluster in which all of the worker nodes and master node are in the **Ready** state.

Policy and Policy Elements

CTE for Kubernetes Policy

Name	Access
	all_ops with Permit, Apply Key, Audit
	all_ops with Deny, Audit
Default	No Access

Steps to integrate Rancher with CTE for Kubernetes Cluster

You will need to use K3s for installation. K3s is a lightweight Kubernetes distribution created by Rancher Labs.

Create a Kubernetes cluster for Rancher

- Use the default K3s installation script, type:

```
curl -sfL https://get.k3s.io INSTALL_K3S_CHANNEL="v1.21" K3S_KUBECONFIG_MODE="644" sh -
```

Testing the cluster

In the previous step, K3s created a new Kubernetes cluster and installed the `kubectl` CLI which you can use to directly interact with the Kubernetes API.

- To list all the nodes in the cluster and check their status, type:

```
kubectl get nodes
```

System Response

```
All Pods are Running
```

Install Helm

Helm is a package manager for Kubernetes. It is used as the installation tool for Rancher when deploying Rancher onto a Kubernetes cluster.

1. Download Helm CLI, type:

```
curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 \ | bash
```

2. Check for the version, type:

```
helm version --client
```

3. Create a soft symlink between rancher, `/etc/rancher/k3s/k3s.yaml` and the Kubernetes config file, `~/.kube/config` so that helm can interact with your cluster, type:

```
mkdir -p ~/.kube ln -s /etc/rancher/k3s/k3s.yaml ~/.kube/config
```

4. Validate the connection, type:

```
helm ls --all-namespaces
```

Install Kubernetes Cert-Manager

Kubernetes cert-manager is an add-on that automates the management and issuance of TLS certificates from various issuing sources.

1. Obtain the latest version of the `cert-manager`, type:

```
kubectl apply --validate=false -f https://github.com/jetstack/certmanager/releases/download/v1.5.1/cert-manager.crds.yaml
```

2. Add the Helm repository for Jetstack:

```
helm repo add jetstack https://charts.jetstack.io
```

3. Install `cert-manager`.

```
helm install cert-manager jetstack/cert-manager --namespace cert-manager --version v1.5.1 --create-namespace
```

4. Once the Helm chart is installed, you can monitor the rollout status of `cert-manager`.

```
kubectl -n cert-manager rollout status deploy/cert-manager
```

System Response

```
Waiting for deployment cert-manager rollout to finish: 0 of 1 updated replicas are available... deployment "cert-manager" successfully rolled out
```

5. You can also monitor the rollout status of `cert-manager-webhook`.

```
kubectl -n cert-manager rollout status deploy/cert-manager-webhook
```

Install Rancher

Install Rancher in HA mode on your Rancher01 Kubernetes cluster.

1. Add rancher-stable as a Helm repository, type:

```
helm repo add rancher-stable https://releases.rancher.com/servercharts/stable
```

2. Install Rancher using the Helm install command, type:

```
helm install rancher rancher-stable/rancher \
--namespace cattle-system \
--set hostname=rancher.${vminfo:rancher01:public_ip}.sslip.io \
--set replicas=1 \
--set bootstrapPassword=Rancher0nK3s \
--create-namespace \
```

3. Verify that Rancher is successfully deployed, type:

```
kubectl -n cattle-system rollout status deploy/rancher
```

4. Get container details for the cattle-system namespace, type:

```
kubectl get pod -n cattle-system
```

5. List all pods in all namespaces, type:

```
kubectl get pods --all-namespaces
```

6. Verify Rancher is ready to access, type:

```
curl -kv https://rancher.${vminfo:Rancher01:public_ip}.sslip.io 2>&1 | grep
-q "dynamiclistener-ca"; if [ $? != 0 ]; then echo "Rancher isn't ready yet";
sleep 5; continue; fi; break; done; echo "Rancher is Ready";
```

Integrating CTE for Kubernetes with Tanzu Kubernetes Grid

Tanzu Kubernetes Grid is an enterprise-ready Kubernetes runtime that unifies management of Kubernetes clusters, vSphere-optimized pods, containerized workloads, and VMware virtual machines.

This document describes installing Tanzu and deploying a few workloads to a Kubernetes cluster.

Test Environment

- **CTE for Kubernetes Agent:** 1.4.0
- **CipherTrust Manager:** 2.15.0

-
- **OS:** ubuntu-2204-kube-v1.28.4+vmware.1-tkg.1.ova
 - **VMware Tanzu Kubernetes Grid:** 2.5.0
 - **VMware Tanzu CLI:** 1.1.0

Prerequisites

Prior to Setting up CipherTrust Transparent Encryption with Tanzu, you must:

- Download the OS `.ova` from the [VMware Tanzu product page](#)
- Install Tanzu CLI for use with Tanzu Kubernetes Grid v2.5
- Install Tanzu CLI Plugins for TKG v2.5
- Install Kubernetes CLI (kubectl)
- Prepare VMware vSphere Server
- Create an SSH Key Pair
- DeployStandalone Management Clusters

References

For more information about deploying Tanzu, reference the following documents:

- [Prepare to Deploy Management Clusters to vSphere](#)
- [Interoperability Result](#)

Deploying Standalone Management Clusters

1. On the system on which you downloaded and installed the Tanzu CLI, type:

```
tanzu management-cluster create -ui
```

2. Click **Deploy** for VMware vSphere.
3. Enter the vCenter Single Sign-On username and password for a user account that has the required privileges for the Tanzu Kubernetes Grid operation.

Note

The account name must include the domain, for example administrator@vsphere.local.

4. Click **Connect**.
5. If the **Disable Verification** option is deselected in the previous screen, then you must manually verify the SSL thumbprint of the vCenter Server certificate and click **Continue**.
6. If you are deploying a management cluster to a vSphere 7 or vSphere 8 instance, confirm whether you want to proceed with the deployment.
7. Select the Datacenter in which to deploy the management cluster from the Datacenter drop-down menu.
8. Manually paste the contents of the key into the text box to add your SSH public key and click **Next**.
9. In the Management Cluster Settings:
 - a. Select a size from the predefined CPU, memory, and storage configurations.

Note

Minimum configuration: 2 CPUs and 4 GB memory

- b. Enter a name for your management cluster.
 - c. Select the **Machine Health Checks** option if you want to activate it.

Note

You can activate, or deactivate, Machine Health Checks on clusters after deployment using the CLI.

- d. Select the **Enable Audit Logging** option if you want to record requests made to the Kubernetes API server.
 - e. For **Worker Node Instance Type**, select the configuration for the worker node virtual machine (VM).
 - f. For **Control Plane Endpoint Provider**, select **Kube-vip**.
 - e. For **Control Plane Endpoint**, enter a static virtual IP address, or FQDN, for API requests to the management cluster.

Management Cluster Settings

Development

Single control plane node. Recommended for a development environment.

Instance type:

MANAGEMENT CLUSTER NAME (OPTIONAL)

CONTROL PLANE ENDPOINT PROVIDER

WORKER NODE INSTANCE TYPE

MACHINE HEALTH CHECKS ☒ Enable

CONTROL PLANE ENDPOINT

ACTIVATE AUDIT LOGGING ☒ Enable

NEXT

10. Click **Next** two times until you get to the **vSphere Resources** screen.
 - a. Select the VM folder in which to place the management cluster VMs.
 - b. Select the vSphere datastores for the management cluster to use.

Note

The storage policy for the VMs can **only** be specified when you deploy the management cluster from a configuration file.

- c. For the **Specify Availability Zones**, choose where to place the management cluster nodes, and then fill in the specifics:

Resources

Resource Pool: /Tanzu-DS/host/10.10.10.10/Resources/TKG, VM Folder: /Tanzu-DS/vm, Datastore: /Tanzu-DS/datastore/datastore 2

Specify the Resources

VM FOLDER

Specify Availability Zones

☒ No AZs ☐ Cluster based AZs ☐ Host group based AZs

DATASTORE

CLUSTERS, HOSTS, AND RESOURCE POOLS

☒ TKG

NEXT

11. Configure the Kubernetes network. Use the IP addresses from the Kubernetes CIDR block. Click **Next** when finished.

Kubernetes Network Settings

CNI Provider: Antrea

Network Name: /Tanzu-DS/network/vm-network

Cluster Service CIDR: 100.64.0.0/12

Cluster Pod CIDR: 100.96.0.0/11

Proxy Settings

☐ Activate Proxy Settings

Next

12. Select the base OS image.

OS image: /Tanzu-DS/vm/Templates/ubuntu-2204-efi-kube-v1.28.4+vmware.1

OS Image with Kubernetes v1.28.4+vmware.1-tkg.1

OS IMAGE: /Tanzu-DS/vm/Templates/ubuntu-2204-efi-kube-v1.28.4+vm

Next

13. Finalize the deployment, click **Review configuration**.
14. Click **Deploy Management Cluster** to deploy the cluster.

Post Deployment

After the deployment completes, you can see one control plane and one worker node.



1. After the deployment completes, a configuration `yaml` file is created in `.../.config/tanzu/tkg/clusterconfigs/<random_name>.yaml`.
2. Create a copy of this `yaml` file and name it: `<newName>.yaml`.
3. Edit the `yaml` file. Provide a static IP address for the `VSPHERE_CONTROL_PLANE_ENDPOINT`.
4. Save the file.

Create a Workload Cluster

1. Create workload cluster, type:

```
tanzu cluster create workernew -f <newName>.yaml
```

2. Scale the worker node count to three, type:

```
tanzu cluster scale tkgwork01 --controlplane-machine-count 1 --worker-machine-count 3
```

3. Create a Tanzu cluster list, type:

```
tanzu cluster list
```

4. Get the Kubernetes node list, type:

```
kubectl get nodes
```

5. Retrieve the information about the current Kubernetes context, type:

```
kubectl config get-contexts
```

Installation

See [Install CTE for Kubernetes](#) for more information.

Policy and Policy Elements

Use a policy with the following access:

CTE for Kubernetes Policy

Name	Access
	all_ops with Permit, Apply Key, Audit
	all_ops with Deny, Audit
Default	No Access



CTE for Kubernetes

Version: v.1.7.0 (latest)

30 January 2026

Contents

- CTE for Kubernetes Signing Tool 3
 - Signing Tool 3
 - Introduction to the CTE Signing Tool 3
 - Installing and Using the CTE Signing Tool 3

CTE for Kubernetes Signing Tool

- [Signing Tool](#)

Signing Tool

- [Introduction to the Signing Tool](#)
- [Installing and Using the Signing Tool](#)

Introduction to the CTE Signing Tool

CTE for Kubernetes uses standard signature sets to test if a pod in the cluster has permission to use a CTE for Kubernetes Storage Group policy. Currently, CTE for Kubernetes is unable to pass signature sets to CipherTrust Manager like CipherTrust Transparent Encryption can. To alleviate this problem, Thales has created a utility that can populate CipherTrust Manager with signatures from other containers.

Specifically, the utility:

1. Calculates a list of binary paths.
2. Includes the URL and tag of a container from which to get the image.
3. Collects CipherTrust Manager authentication credentials.
4. Downloads the container image and collects signatures from it.
5. Calculates the SHA256 sum of all of the binary paths requested.
6. Requests authentication with CipherTrust Manager.
7. Pushes signatures collected from other containers to CipherTrust Manager.

Obtaining the Tool

This tool is available from the [Thales Support Portal](#). Search for **CipherTrust Transparent Encryption for Kubernetes**. The file will be a tar file. Decompress it with `tar -xvf`.

Installing and Using the CTE Signing Tool

To use this tool to gather signatures from other container images and then push them to CipherTrust Manager, complete the following steps:

1. Download a container image:
 - a. From a **public** repository, type:

```
./cte-sign --cdi --image=<image name> --tag=<image version> --arch=<amd64/arm64/..>
```

b. From a **private** repository, type:

```
./cte-sign --cdi --image=<image name with url> --tag=<image version> --repo-user=<repo username> --repo-password=<repo password> --arch=<amd64/arm64/..>
```

2. Get a list of signature set IDs from CipherTrust Manager, type:

```
./cte-sign --cm --signature-sets list --ip=<CM IP> --user=<CM username> --password=<CM password> --cm-domain=<CM Domain>
```

3. Create a signature set for the process set or Container Image Digest set, type:

```
./cte-sign --cm --signature-sets create --sigset-name=<signature-set-name> --sigset-type=<signature-set-type> --ip=<CM IP> --user=<CM username> --password=<CM password> --cm-domain=<CM Domain>
```

4. Send the signatures collected from the downloaded container image to CipherTrust Manager in a signature set ID, type:

```
./cte-sign --cm --signature-sets add-signatures --sigset-name=<signature-set-name> --path=<source-path> --ip=<CM IP> --user=<CM username> --password=<CM password> --cm-domain=<CM Domain>
```

5. Cleanup and remove the image directory, type:

```
./cte-sign --cleanup
```

6. Retrieve container image digests

a. Retrieve digests for pod spec yaml file:

```
./cte-sign --digest --podspec=<pod spec file name> --kubeconfig=<kubeconfig file path>
```

b. Retrieve digest for public image:

```
./cte-sign --digest --image=<image name> --tag=<image tag>
```

c. Retrieve digest for private image:

```
./cte-sign --digest --image=<image name> --tag=<image tag> --repo-user=<repo username> --repo-password=<repo password>
```

7. Retrieve and push Container Image Digest to CipherTrust Manager

a. Retrieve and push for public image:

```
./cte-sign --digest --push-to-cm --image=<image name> --tag=<image tag> --ip=<CM IP> --user=<CM username> --password=<CM password> --sigset-name=<signature-set-name> --cm-domain=<CM Domain>
```

b. Retrieve and push for private image:

```
./cte-sign --digest --push-to-cm --image=<image name> --tag=<image tag> --repo-user=<repo username> --repo-password=<repo password> --ip=<CM IP> --user=<CM username> --password=<CM password> --sigset-name=<signature-set-name> --cm-domain=<CM Domain>
```

c. Retrieve and push for public/private images using pod spec yaml file:

```
./cte-sign --digest --push-to-cm --podspec=<pod spec file name> --ip=<CM IP> --user=<CM username> --password=<CM password> --sigset-name=<signature-set-name> --cm-domain=<CM Domain>
```

CLI Argument Definitions

Argument	Definition
--arch	Image architecture: amd64, arm64, ...
--cleanup	Delete the container image files and directories.
--cm-domain	(Optional) Provide this option only if an operation belongs to a specific domain, other than the root domain.
--image	Container image name/URL.
--ip	CipherTrust Manager IP address.
--kubeconfig	kubeconfig file for retrieving pod digests.
--password	Password of CipherTrust Manager credential.
--path	The binary path from the downloaded container image, for the files that need to be pushed to CipherTrust Manager. ex: /usr/bin

Argument	Definition
--podspec	Pod specification yaml file.
--repo-password	Password/AccessToken for the private repository.
--repo-user	Username for private repository access.
--sigset-name	The signature set name where signatures will be contained.
--sigset-type	The signature set type. Application: for process signatures. Container-Image for container image signature.
--tag	Container image version.
--user	Username of CipherTrust Manager credential.

CipherTrust Transparent Encryption for Kubernetes



Challenge: Securing Applications for Kubernetes Environments

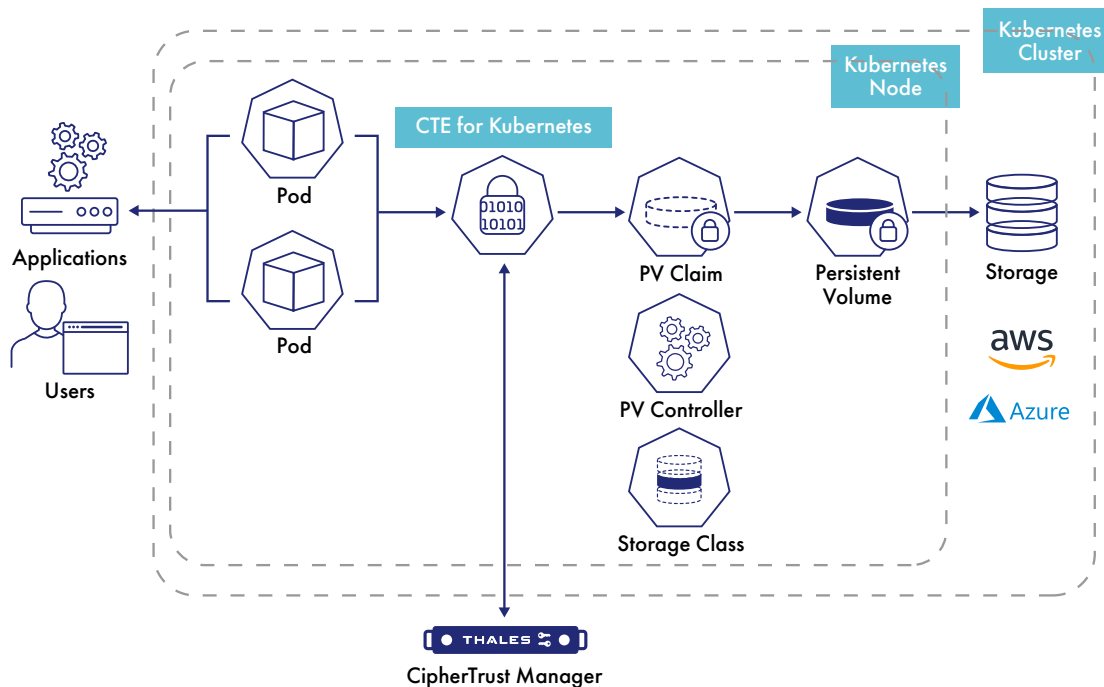
Modern applications are increasingly built using containers, which are microservices packaged with their configurations and dependencies. Kubernetes is an open-source software for deploying and managing these containers. Containerized applications can be delivered, deployed and managed faster with Kubernetes to provide improved efficiency through re-usable modular components, cost savings through optimized resource utilization and reduced licensing expenses. However, there are several risks:

- **Privileged user abuse.** By default, Docker processes run with root privileges, administrators have full access to all tenant secrets. This level of untethered access poses multiple risks. Organizations could be subject to privilege escalation attacks, if administrators have unchecked access to container images and the data stored within them.
- **Cross container access.** Poor configuration of permissions can result in multiple containers having access to information that should remain private. Further, when containers are hosted in shared virtualized or cloud environments, critical information can be exposed to third parties.

- **Compliance risks.** Many compliance mandates require stringent access controls and auditing data access. However, many security teams have limited controls available to manage and track access to data held within containers and images. As a result, these teams find it difficult to comply with relevant security policies and regulatory mandates.

Solution: CipherTrust Transparent Encryption for Kubernetes

CipherTrust Transparent Encryption for Kubernetes delivers in-container capabilities for encryption, access controls, and data access logging, that enables organizations to establish strong safeguards around data in Kubernetes environments. With this extension for CipherTrust Transparent Encryption, data protection can be applied on a per-container basis, both to secure data inside of containers and in external storage accessible from containers, all centrally managed from the CipherTrust Manager.



Benefits

CipherTrust Transparent Encryption for Kubernetes provides:

- **Compliance.** This extension of CipherTrust Transparent Encryption addresses compliance requirements and regulatory mandates for protecting sensitive data such as— payment cards, healthcare records or other sensitive assets.
- **Protection from Privileged-User Threats.** This solution offers encryption with data access controls, enabling privileged users, such as Docker or OpenShift cluster administrators, to operate as regular users, without gaining unauthorized access to sensitive data.
- **Achieve Robust Security.** CipherTrust Transparent Encryption for Kubernetes enforces data security policies wherever the container is stored or used – data centers, virtualized environments, even in cloud implementations. Deploy and use containers where needed for cost effectiveness, control, or performance without having to make any changes to applications, containers, or infrastructure sets.

- **Granular Access Controls and Visibility.** CipherTrust Transparent Encryption for Kubernetes offers the detailed visibility and control you need to comply with the most stringent policies and mandates. With this Kubernetes security solution, enterprises can establish granular access policies based on specific users, processes, and resource sets within containers. Finally, this solution can establish isolation between containers, so only authorized containers can access sensitive information.

CipherTrust Manager

The CipherTrust Manager centralizes key, policy, and log management for the CipherTrust Data Security Platform including CipherTrust Transparent Encryption. CipherTrust Manager is available in both virtual and physical form-factors that integrate with FIPS 140-2 validated Thales TCT Luna T-Series and third-party Hardware Security Modules (HSMs) for securely storing master keys with highest root of trust.

Features

- **Comprehensive Data Security Safeguards.** CipherTrust Transparent Encryption for Kubernetes extends CipherTrust Transparent Encryption, enabling security teams to establish data security controls inside of containers. With this extension, you can apply encryption, access controls, and data access logging on a per-container basis. Encryption can be applied to data generated and stored locally within the container and to data mounted in the container by network file systems.
- **Scalable Transparent Encryption.** Provides data security controls without having to make any changes to applications, containers or infrastructure sets. It enables a single policy to be applied to all containers within a Kubernetes cluster, or distinct policies applied to each container within a cluster. This solution can scale up or shrink down a Kubernetes environments as business needs change.

About Thales Trusted Cyber Technologies

Thales Trusted Cyber Technologies, a business area of Thales Defense & Security, Inc., is a trusted, U.S. provider of cybersecurity solutions dedicated to U.S. Government. We protect the government's most vital data from the core to the cloud to the edge with a unified approach to data protection. Our solutions reduce the risks associated with the most critical attack vectors and address the government's most stringent encryption, key management, and access control requirements.

For more information, visit www.thalestct.com

CipherTrust Transparent Encryption Container Security



Challenge: Building continuous container security into your CI/CD pipeline

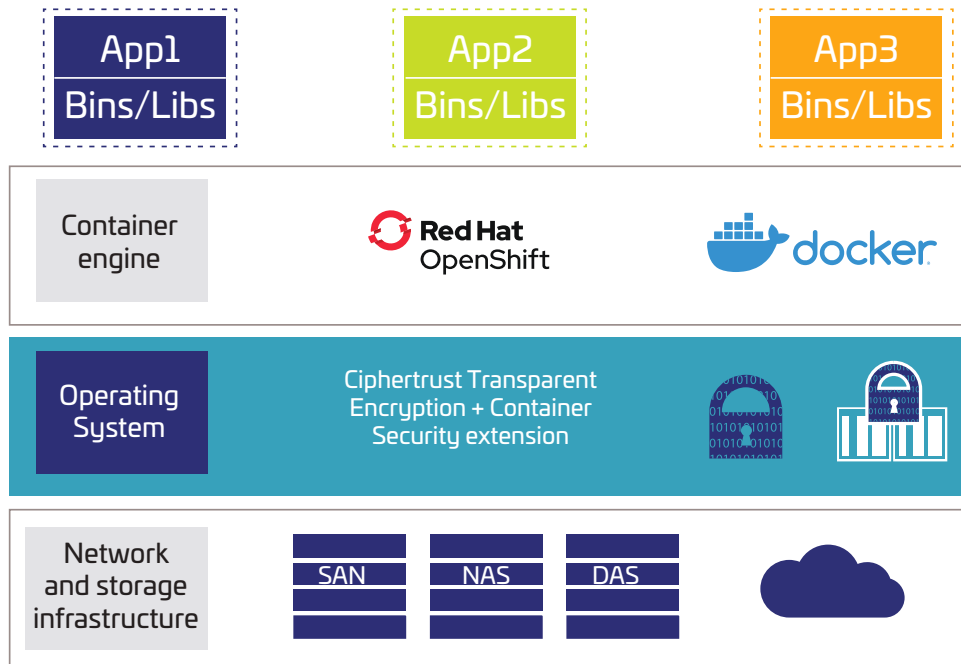
As enterprises deploy containers with a continuous integration and delivery (CI/CD) pipeline, security must keep up. Container technologies provide significant benefits including increased portability, delivery and innovation, improved efficiency through reusable modular components, and cost savings through optimized resource utilization and reduced licensing expenses. However, there are also risks:

- **Privileged user abuse.** By default, Docker processes run with root privileges, and OpenShift cluster administrators have full access to all tenant secrets. This level of privileged access poses multiple risks. For example, container administrators may have unchecked access to images and the data stored within them, and organizations could be subject to privilege escalation attacks.
- **Cross container access.** Poor configuration of permissions can result in multiple containers having access to information that should remain private. Further, when containers are hosted in shared virtualized or cloud environments, critical information can be exposed to third parties.

- **Compliance risks.** Many compliance mandates require strong controls and auditing data access. However, many security teams have limited controls available to manage and track access to data held within containers and images. As a result, these teams find it difficult to comply with relevant internal security policies and regulatory mandates.

Solution: CipherTrust Transparent Encryption Container Security

CipherTrust Transparent Encryption Container Security delivers in-container capabilities for encryption, access controls, and data access logging, so organizations can establish strong safeguards around data in dynamic container environments. This solution from Thales is a software license for CipherTrust Transparent Encryption that enables security teams to establish controls inside of containers. With this extension, encryption, access controls, and data access audit logging can be applied on a per-container basis, both to data inside of containers and to external storage accessible from containers.



Benefits

CipherTrust Transparent Encryption Container Security provides:

- **Compliance.** This extension of CipherTrust Transparent Encryption delivers the encryption, data access control, and auditing capabilities you need to address compliance requirements and regulatory mandates. You can protect sensitive data — whether your organization manages payment cards, healthcare records or other sensitive assets.
- **Protection from Privileged-User Threats.** CipherTrust Transparent Encryption Container Security offers encryption with data access control, enabling privileged users, such as Docker or OpenShift cluster administrators, to work as usual, without unauthorized access to sensitive data.
- **Achieve Robust Security.** CipherTrust Transparent Encryption Container Security enforces data security policies wherever the container is stored or used – data centers, virtualized environments, even in cloud implementations. Deploy and use containers where needed for cost effectiveness, control, or performance without having to make any changes to applications, containers, or infrastructure sets.

Features

- **Comprehensive Data Security Safeguards.** CipherTrust Transparent Encryption Container Security extends CipherTrust Transparent Encryption, enabling security teams to establish data security controls inside of containers. With this extension, you can apply encryption, access control, and data access logging on a per-container basis. Encryption can be applied to data generated and stored locally within the container and to data mounted in the container by network file systems.
- **Transparent Encryption.** No changes to containers required. Establish data security controls without having to make any changes to applications, containers or infrastructure sets. The container security solution even supports common container microservices deployment models. This enables single policies that can apply to all containers running on a container host instance and allows distinct policies for each container or a mix of policy types.
- **Granular Access Controls and Visibility.** CipherTrust Transparent Encryption Container Security offers the detailed visibility and control you need to comply with the most stringent policies and mandates. With this container security solution, enterprises can establish granular access policies based on specific users, processes, and resource sets within containers. Finally, this solution can establish isolation between containers, so only authorized containers can access sensitive information.

About Thales Cloud Protection & Licensing

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.

> cpl.thalesgroup.com <    

Contact us – For all office locations and contact information, please visit cpl.thalesgroup.com/contact-us

CipherTrust Transparent Encryption for Kubernetes



Challenge: Securing Applications for Kubernetes Environments

Modern applications are increasingly built using containers, which are microservices packaged with their configurations and dependencies. Kubernetes is an open-source software for deploying and managing these containers. Containerized applications can be delivered, deployed and managed faster with Kubernetes to provide improved efficiency through re-usable modular components, cost savings through optimized resource utilization and reduced licensing expenses. However, there are several risks:

- **Privileged user abuse.** By default, Docker processes run with root privileges, administrators have full access to all tenant secrets. This level of untethered access poses multiple risks. Organizations could be subject to privilege escalation attacks, if administrators have unchecked access to container images and the data stored within them.
- **Cross container access.** Poor configuration of permissions can result in multiple containers having access to information that should remain private. Further, when containers are hosted in shared virtualized or cloud environments, critical information can be exposed to third parties.
- **Compliance risks.** Many compliance mandates require stringent access controls and auditing data access. However, many security teams have limited controls available to manage and track access to data held within containers and images. As a result, these teams find it difficult to comply with relevant security policies and regulatory mandates.

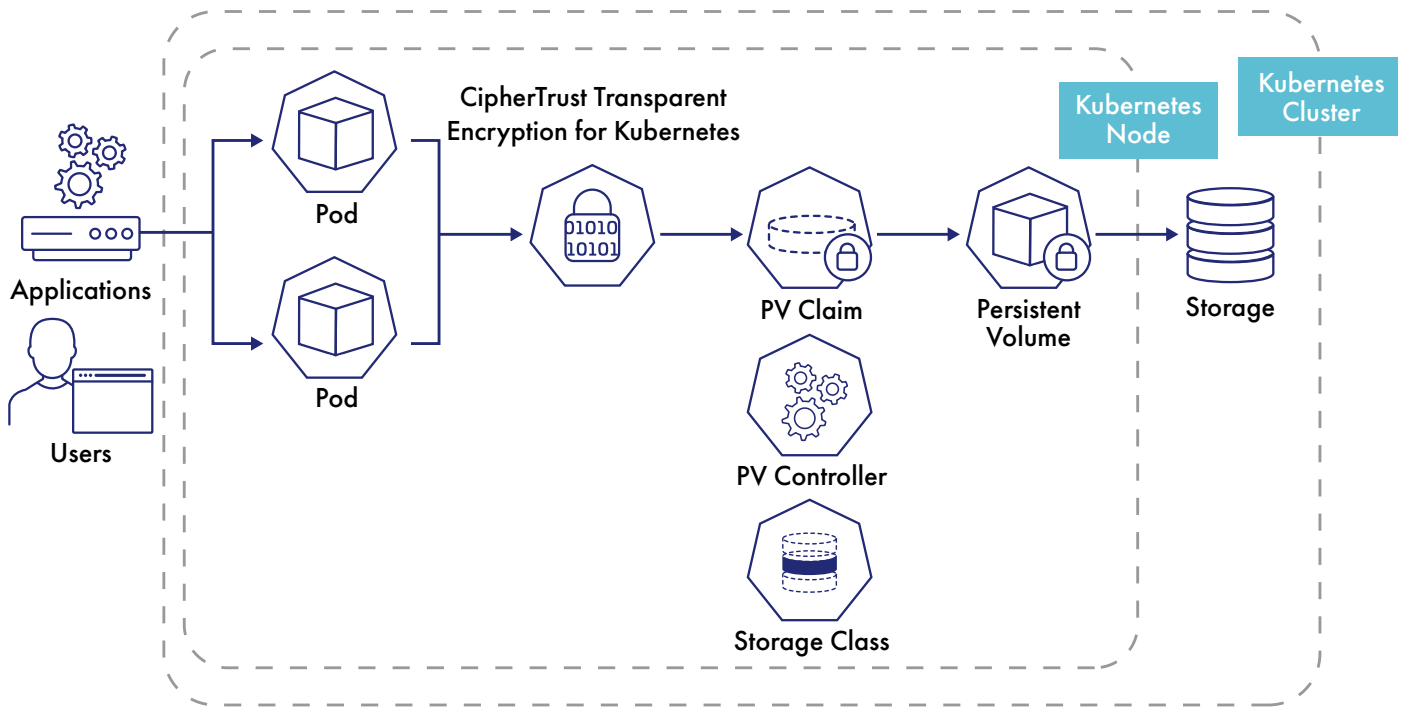
Solution: CipherTrust Transparent Encryption for Kubernetes

CipherTrust Transparent Encryption for Kubernetes delivers in-container capabilities for encryption, access controls, and data access logging, that enables organizations to establish strong safeguards around data in Kubernetes environments. With this extension for CipherTrust Transparent Encryption, data protection can be applied on a per-container basis, both to secure data inside of containers and in external storage accessible from containers, all centrally managed from the CipherTrust Manager..

Benefits

CipherTrust Transparent Encryption for Kubernetes provides:

- **Compliance.** This extension of CipherTrust Transparent Encryption addresses compliance requirements and regulatory mandates for protecting sensitive data such as— payment cards, healthcare records or other sensitive assets.
- **Protection from Privileged-User Threats.** This solution offers encryption with data access controls, enabling privileged users, such as Docker or OpenShift cluster administrators, to operate as regular users, without gaining unauthorized access to sensitive data.
- **Achieve Robust Security.** CipherTrust Transparent Encryption for Kubernetes enforces data security policies wherever the container is stored or used – data centers, virtualized environments, even in cloud implementations. Deploy and use containers where needed for cost effectiveness, control, or performance without having to make any changes to applications, containers, or infrastructure sets.



Features

- **Comprehensive Data Security Safeguards.** CipherTrust Transparent Encryption for Kubernetes extends CipherTrust Transparent Encryption, enabling security teams to establish data security controls inside of containers. With this extension, you can apply encryption, access controls, and data access logging on a per-container basis. Encryption can be applied to data generated and stored locally within the container and to data mounted in the container by network file systems.
- **Scalable Transparent Encryption.** Provides data security controls without having to make any changes to applications, containers or infrastructure sets. It enables a single policy to be applied to all containers within a Kubernetes cluster, or distinct policies applied to each container within a cluster. This solution can scale up or shrink down a Kubernetes environments as business needs change.
- **Granular Access Controls and Visibility.** CipherTrust Transparent Encryption for Kubernetes offers the detailed visibility and control you need to comply with the most stringent policies and mandates. With this Kubernetes security solution, enterprises can establish granular access policies based on specific users, processes, and resource sets within containers. Finally, this solution can establish isolation between containers, so only authorized containers can access sensitive information.

CipherTrust Manager

The CipherTrust Manager centralizes key, policy, and log management for the CipherTrust Data Security Platform including CipherTrust Transparent Encryption. It is available in both virtual and physical form-factors for securely storing master keys with an elevated root of trust. These appliances can be deployed on-premises as well as in private or public cloud infrastructures.

About Thales Cloud Protection & Licensing

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.

Solution Brief

CipherTrust Tokenization for Executive Peace of Mind

cpl.thalesgroup.com

THALES
Building a future we can all trust

When you need:

- **Compliance for security regulations such as PCI DSS**
- **Ciphertext that looks like your data**
- **Reduced audit scope and burden, proof you have done due diligence**

The Challenge

Executives today are under increasing pressure to protect sensitive customer data. Data breaches can result in financial penalties, public disclosure, and even personal liability. Regulations like PCI DSS, HIPAA, and GDPR demand strong safeguards for personal data—but achieving compliance without hindering operations is difficult.

Data protection solutions often exist in silos. Developers are burdened with crypto implementation. Policy enforcement is inconsistent and the audit trail is fragmented.

How Thales Can Help

Thales provides Tokenization as part of a unified Data Security Platform that supports Tokenization, Encryption, Data Masking and Redaction so that you have the right data protection for every type of data you have—regardless of where it resides; in an application or a database, on premises or in the cloud.

CipherTrust Tokenization provides both Vaulted and Vaultless Tokenization solutions enabling you to:

- Protect customer data whether it is in applications or databases
- Replace sensitive values with format-preserving tokens
- Choose vaulted or vaultless models to fit your systems
- Centralize control of policies, keys and audit trails

Both Vaulted and Vaultless options integrate with CipherTrust Manager and a FIPS 140-3 Level 3 HSM, helping you prove due diligence and reduce scope.

CipherTrust Tokenization Solutions

CipherTrust Vaulted Tokenization is available with:

- CipherTrust Vaulted Tokenization (CT-V) - offers non-mathematically reversible tokens to protect data via RESTful services, SOAP web services, Java SDKs and .Net SDKs within a customer's environment. Supports Oracle, Microsoft SQL Server, MySQL and Informix databases as its vault.

In contrast to vaulted tokenization, CipherTrust Vaultless Tokenization, does not store any sensitive user data, thereby reducing your potential attack surface and risk.

CipherTrust Vaultless Tokenization solutions include:

- CipherTrust Batch Data Transformation (BDT) - offers high-performance tokenization and encryption for databases and structured files
- CipherTrust Application Data Protection (CADP) - offers tokenization and encryption for field-level data protection as a simple-to-integrate library for developers
- CipherTrust RESTful Data Protection (CRDP) - offers tokenization and encryption for field-level data protection as a RESTful service
- CipherTrust Data Protection Gateway (DPG) - offers tokenization and encryption for transparent field-level data protection to any RESTful web service or microservice leveraging REST APIs
- CipherTrust Database Protection (CDP) - offers tokenization and encryption for transparent, column level data protection for a wide range of databases.

BDT, CADP, CRDP and DPG support Static Data Masking and Redaction. CADP, CRDP and DPG additionally support Dynamic Data Masking.

Thales helps you secure sensitive data across any system without operational complexity. And it gives you the ability to demonstrate accountability when it matters most.

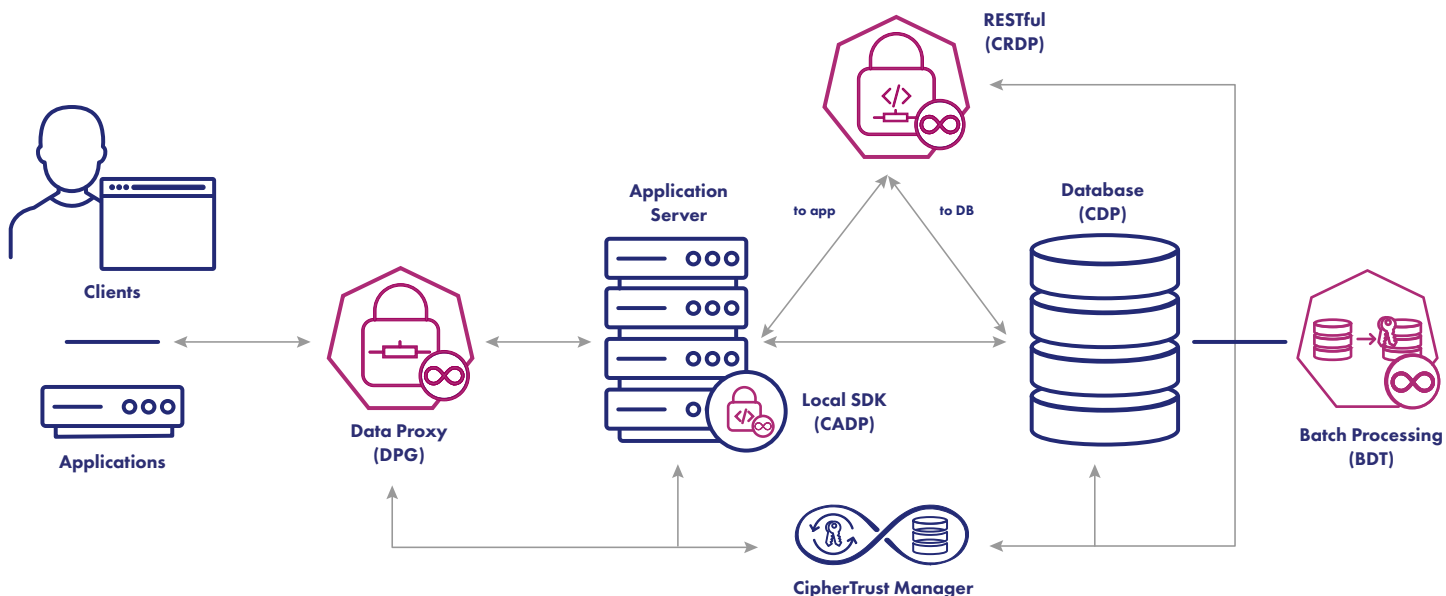
Thales Solution Benefits

- Reduce PCI DSS compliance scope and associated audit burden
- Protect PII, PHI, and cardholder data even if systems are breached
- Avoid liability by proving data was protected and access was controlled
- Combine Tokenization with Data Masking and Redaction to deliver end-to-end control
- Enable innovation in cloud and analytics environments without increasing risk
- Tokenize data at rest, in motion, and in use

Core Use Cases

- Format-preserving Tokenization for applications and databases
- Vaulted or Vaultless protection for data in cloud, hybrid, or on-prem environments
- Tokenization, Encryption, Data Masking and Redaction on one platform

CipherTrust Vaultless Tokenization Solutions



Conclusion

Thales Tokenization solutions reduce executive exposure by removing sensitive customer data from high-risk systems. It simplifies compliance, lowers audit costs, and strengthens your ability to respond confidently to regulators, customers, and the board.

About The CipherTrust Data Security Platform

The award-winning CipherTrust Data Security Platform is an integrated set of data-centric solutions that remove complexity from data security, accelerate time to compliance, and secure cloud migrations. Thales is proud to have been recognized as an Overall Leader in the KuppingerCole Leadership Compass on Data Security Platforms as well as a Strong Performer in the Forrester Wave and to be featured in Gartner's Market Guide to Data Security Platforms.

The CipherTrust Platform unifies data discovery, classification, data protection, and centralized management for keys and secrets into a single platform. This results in fewer resources dedicated to security operations, ubiquitous compliance controls, and significantly reduced risk across your business.

About Thales

Thales is a global leader in cybersecurity, helping the most trusted organizations protect critical applications, data, identities, and software anywhere, at scale. Through Thales' integrated platforms, customers achieve better visibility of risks, defend against cyber threats, close compliance gaps, and deliver trusted digital experiences for billions of consumers every day.

CipherTrust Tokenization



Anonymizing Data for Security and Compliance

Tokenization reduces the cost and effort required to comply with internal security policies and regulatory mandates such as the European Union's Global Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI-DSS).

CipherTrust Tokenization offers application-level tokenization services in two convenient solutions that deliver complete customer flexibility: Vaultless Tokenization with Dynamic Data Masking and Vaulted Tokenization. Both solutions secure and anonymize sensitive assets—whether they reside in the data center, big data environments or the cloud.

Key Advantages

Efficiently Reduce PCI DSS Compliance Scope

Tokenization can remove card holder data from PCI DSS scope, saving on costs associated with compliance.

Protect Personally Identifiable Information (PII)

With CipherTrust tokenization, PII protection is gained without encryption key management by the software developer

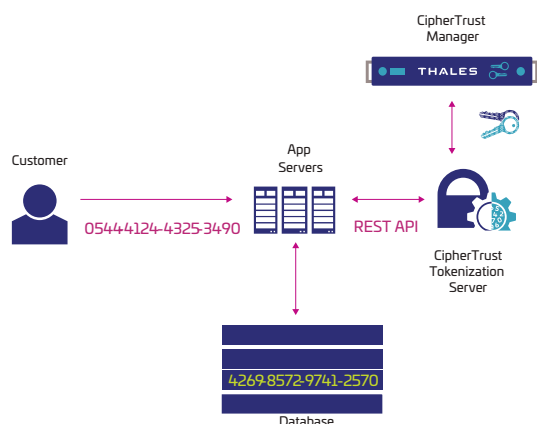
Architect for Your Requirements: Vaultless or Vaulted, and Cloud-Friendly

Both CipherTrust Tokenization solutions leverage CipherTrust Manager as a secure encryption key source. All-software, cloud friendly, is readily available, including with FIPS 140-2 Level 3 Root of Trust.

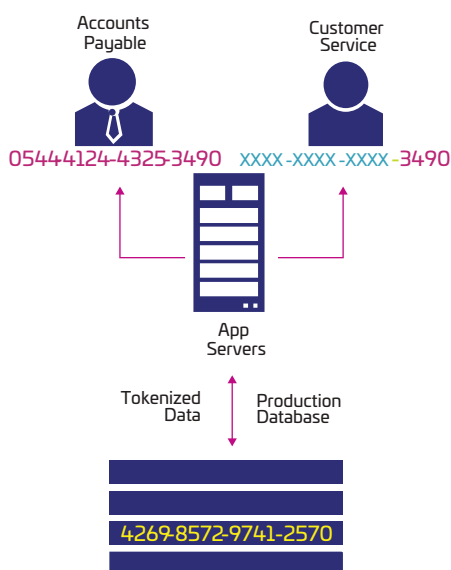


Vaultless Tokenization

CipherTrust Vaultless Tokenization protects data at rest while its policy-based Dynamic Data Masking capability protects data in use. A RESTful API in combination with centralized management and services enables tokenization implementation with a single line of code per field. Vaultless Tokenization is provided by dedicated, distributed-cluster-capable Tokenization Servers, offering full separation of duties. Tokenization management and configuration including an operational dashboard with convenient tokenization configuration workflows occurs in a graphical user interface.



Dynamic Data Masking policies define whether a tokenized field is returned fully or partially masked based on user identification controlled by an AD or LDAP server. For example, the policies could enable customer service representatives to see only the last four digits of credit card numbers, while account receivables staff could access the full credit card number.

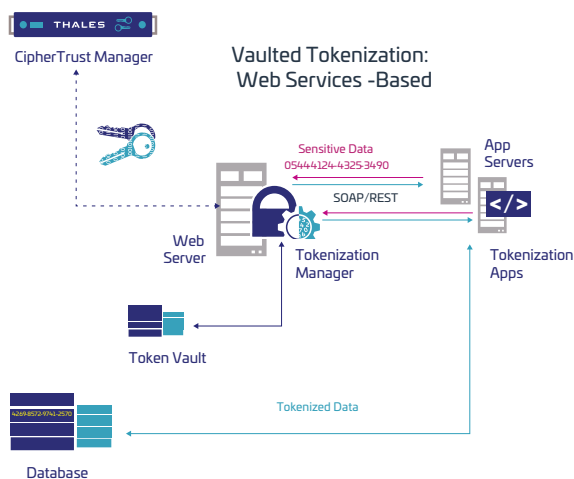
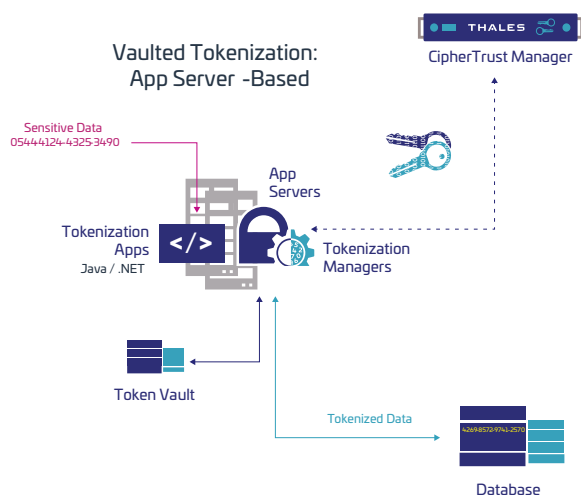


Vaulted Tokenization

CipherTrust Vaulted Tokenization offers non-disruptive format-preserving tokenization with a wide range of existing formats and the ability to define custom tokenization formats.

Vaulted Tokenization provides a high level of security for data with the most sensitivity.

Architectural Choice is available by deploying the Vaulted Tokenization infrastructure on each application server or deploying it centrally on a web server with application servers using simply RESTful APIs for tokenization.



CipherTrust Data Security Platform

CipherTrust Tokenization is part of the CipherTrust Data Security Platform, which unifies data discovery, classification, data protection, and unprecedented granular access controls, all with centralized key management. This simplifies data security operations, accelerates time to compliance, secures cloud migrations and reduces risk across your business. You can rely on Thales CipherTrust Data Security Platform to help you discover, protect and control your organization's sensitive data, wherever it resides.

CipherTrust Vaultless Tokenization with Dynamic Data Masking

Efficiently de-identifying sensitive data and reducing
PCI DSS audit scope



Contents

3	Introduction
5	The Challenge
5	So many security demands, so little time
6	The Solution
6	Efficiently de-identify sensitive data
6	How vaultless tokenization works
7	Deliver Robust Data Security
7	Strong access controls
7	Flexible tokenization with dynamic data masking
8	Dynamic Data Masking of Credit Cards for Customer Service
9	Tokenization options
9	Deployment and administrative simplicity
10	Efficient scalability and agility
10	Key benefits
10	Sample implementations
11	About Thales

Introduction

Today's enterprise security teams have a lot on their plates and a lot on their minds. According to the Thales 2021 Global Data Threat Report, data breaches remain a widespread challenge. In the survey for the report, respondents indicated that 56% of organizations have ever experienced some form of data breach, with 41% of them having experienced a breach in the last 12 months. Breaches are a trailing indicator of security effectiveness. Turning to forward-looking metrics, 47% of respondents reported an increase in volume, severity and/or scope of cyberattacks!

But as you read this paper, consider that 46% of respondents said that they have avoided a breach notification because leaked data was encrypted or tokenized! Attacks are becoming more sophisticated and persistent while insider threats continue to be an Achilles heel. According to the 2017 Thales Data Threat Report¹, 88% feel at least somewhat vulnerable to data threats. The report also proves those concerns are well founded: 42% of organizations experienced a data breach at some point.

Tokenization is a critical tool for enterprises to meet compliance standards, and reduce the risk of loss posed by this increasingly hostile environment. Most often, tokenization has been a great way for enterprises to reduce the cost of their Payment Card Industry Data Security Standard (PCI DSS) audits. Tokens are used to replace credit card numbers, removing the need for many servers to undergo expensive yearly audits. But use cases for tokenization are expanding today to safeguard more types of data and in more environments as the locations where this data is used proliferate.

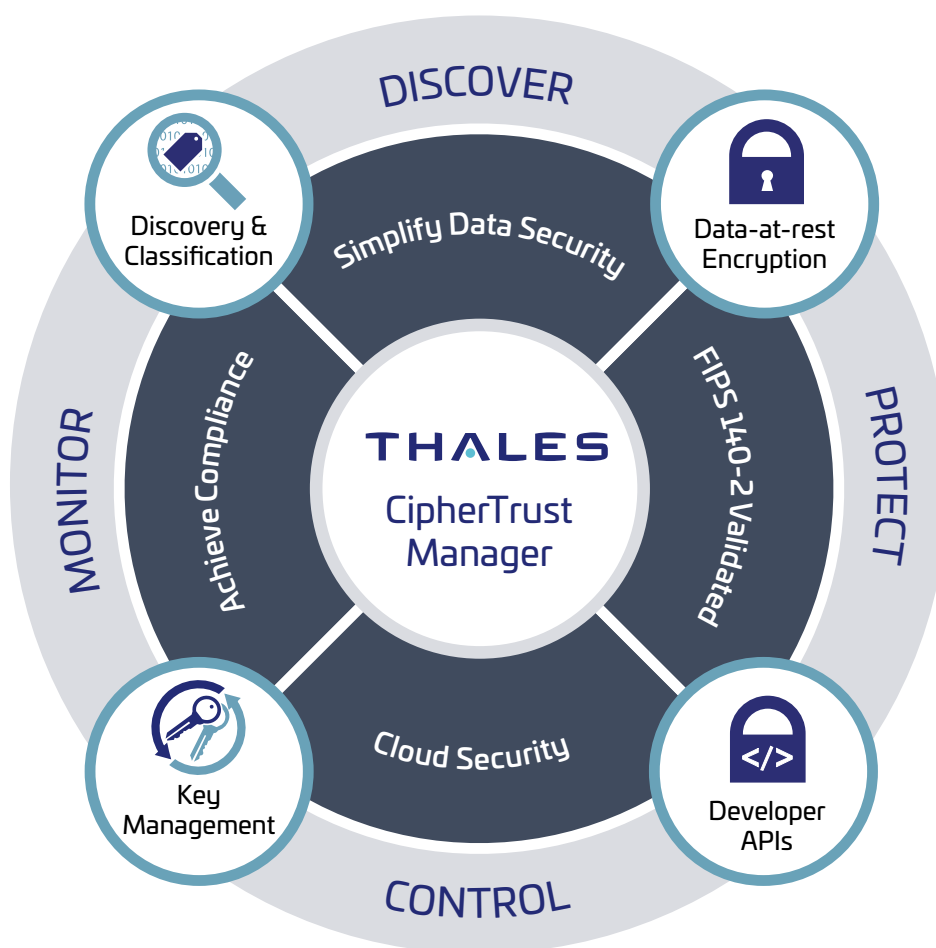
Driven by data privacy and sovereignty regulations around the world, the data types that need to be protected are no longer just credit cards, but also include U.S. Social Security numbers, Drivers Licenses, U.K. National Insurance numbers, Passport numbers, and National Identity numbers in many countries around the world among other data. And the need to use this sensitive data in advanced technology environments like cloud and big data leads to further complications.

In this white paper we'll first look at how CipherTrust Vaultless Tokenization works, key characteristics of the solution, and then examine sample implementations including PCI DSS compliance, customer contact centers, cloud, big data and more.

As noted earlier, tokenization is especially useful in PCI DSS compliance. In the past meeting requirements of the standard has been very complex and expensive, with extensive yearly audits of servers hosting credit card data required, or solutions that were limited in bandwidth and required extensive infrastructure to support.

CipherTrust Tokenization now offers a better way to meet these requirements. The solution's vaultless tokenization capabilities enable security teams to remove databases from PCI DSS scope, and so significantly reduce compliance cost and efforts. Implementations typically result in minimal disruption and administrative overhead and result in breakthroughs in operational efficiency.

CipherTrust Vaultless Tokenization leverages CipherTrust Manager as its key source and, of course, is a component of the CipherTrust Data Security Platform, which delivers a comprehensive range of data security and compliance controls that help IT teams efficiently address their requirements for encryption, privileged user access control, file access auditing, and more. With a single infrastructure, policy and encryption key management environment, the platform reduces the overall costs for organizations to implement and use data security tools while applying strong protections for data.



The Challenge

So many security demands, so little time

For today's security teams, it seems virtually everything is proliferating:

- **Threats.**

Risks posed by nation-states, cyber criminals, and malicious insiders continue to grow more sophisticated and more persistent

- **Data and repositories.**

The volume and diversity of assets and repositories that need to be protected continues to see explosive growth. Instead of just managing the traditional data center, which was challenging enough, there are now virtualized systems, big data environments, outsourced services, private clouds, public clouds, and hybrid models to contend with

- **Mandates and tools.**

As the number of repositories continues to grow, so do the number of systems that fall within the scope of PCI DSS and other regulations. Further, the scope and complexity of these regulatory mandates also continues to expand. As a result, security teams are compelled to deploy more safeguards and tools in order to establish the controls these mandates require. In particular, security teams face increased pressures to prepare for PCI DSS 3.0 audits, and adapt their security approaches as cardholder data moves into new environments—such as cloud and big data implementations—and is shared with outsourced service providers, such as analytics, development, and engineering firms

All this proliferation continues to place increasing demands on security teams—but these teams don't see their time, staffing, or budgets undergoing any commensurate expansion. To contend with these realities, many security professionals have explored the use of tokenization, which has the potential to provide a convenient way to protect sensitive assets. At a high level, tokenization is the process of replacing a sensitive record with a token, effectively a meaningless placeholder for the data that can't be exploited or reverted back to its original form

While tokenization offers the potential to address a wide range of security and compliance objectives, traditional tokenization tools and solutions have been far too costly and complex to procure and manage. These tools typically require security teams to invest in expensive appliances and contend with software agents that are difficult to deploy and manage. More than ever, security teams need to be able to leverage the benefits of tokenization—and they need to do so in a scalable, efficient, and cost effective manner.

The Solution

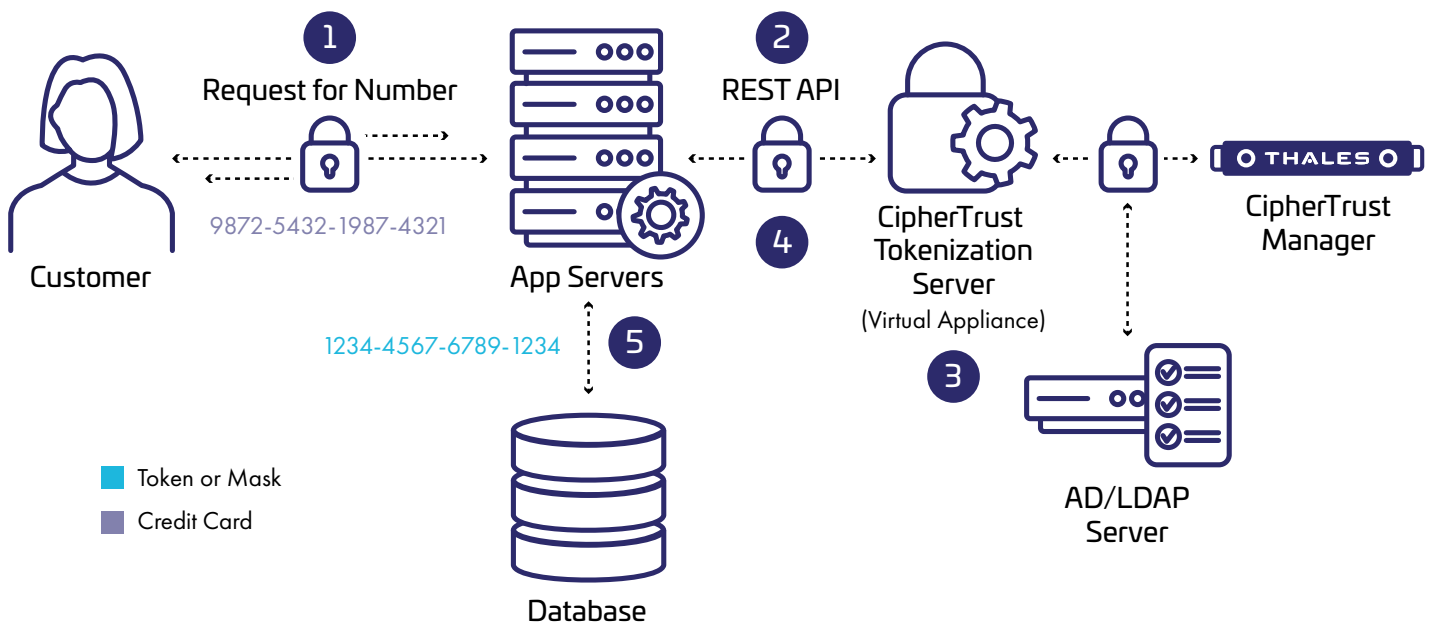
Efficiently de-identify sensitive data

The CipherTrust Data Security Platform offers tokenization capabilities that can dramatically reduce the costs and effort associated with complying with security policies and PCI DSS. With CipherTrust Vaultless Tokenization with Dynamic Data Masking, organizations can efficiently address objectives for securing sensitive assets and cardholder records— whether they reside in the data center, big data environments, or the cloud.

How vaultless tokenization works

For illustration purposes, we'll outline the process of tokenizing a credit card number. The CipherTrust Tokenization Server acts as a central hub for tokenizing records and managing access to tokens and clear-text data.

1. The customer submits a credit card number into an application.
2. The application submits the credit card number to the Tokenization Server via the REST API.
3. The Tokenization Server holds in memory a secure table created based on an AES-256 encryption key seed provided by the CipherTrust Manager. This table holds with random matches for credit card numbers – the token matching that card is identified.
4. Through the REST API, the Tokenization Server returns the token to the application.
5. The application server then submits the tokenized value into the database.



Deliver Robust Data Security

Strong access controls

Thales recommends creating at least two user accounts, or user groups from LDAP, be available on to the Tokenization Server:

- The first account or LDAP group with the permission to tokenize but not to detokenize, and the second account or LDAP group with the permission to detokenize
- The user account with tokenization privileges can then be utilized by multiple servers, including a public-facing web server, for example. This allows data to be tokenized immediately as it is brought in through applications
- Accounts or LDAP groups with permission to have detokenization privileges should be limited to a specific server that needs these permissions. For example, detokenization privileges could be restricted to a server that's responsible for submitting credit card numbers to a remote card processing service
- Any internal LDAP groups that have rights to detokenize data (Application administrators, for instance) should be strictly limited to the minimum number of members required

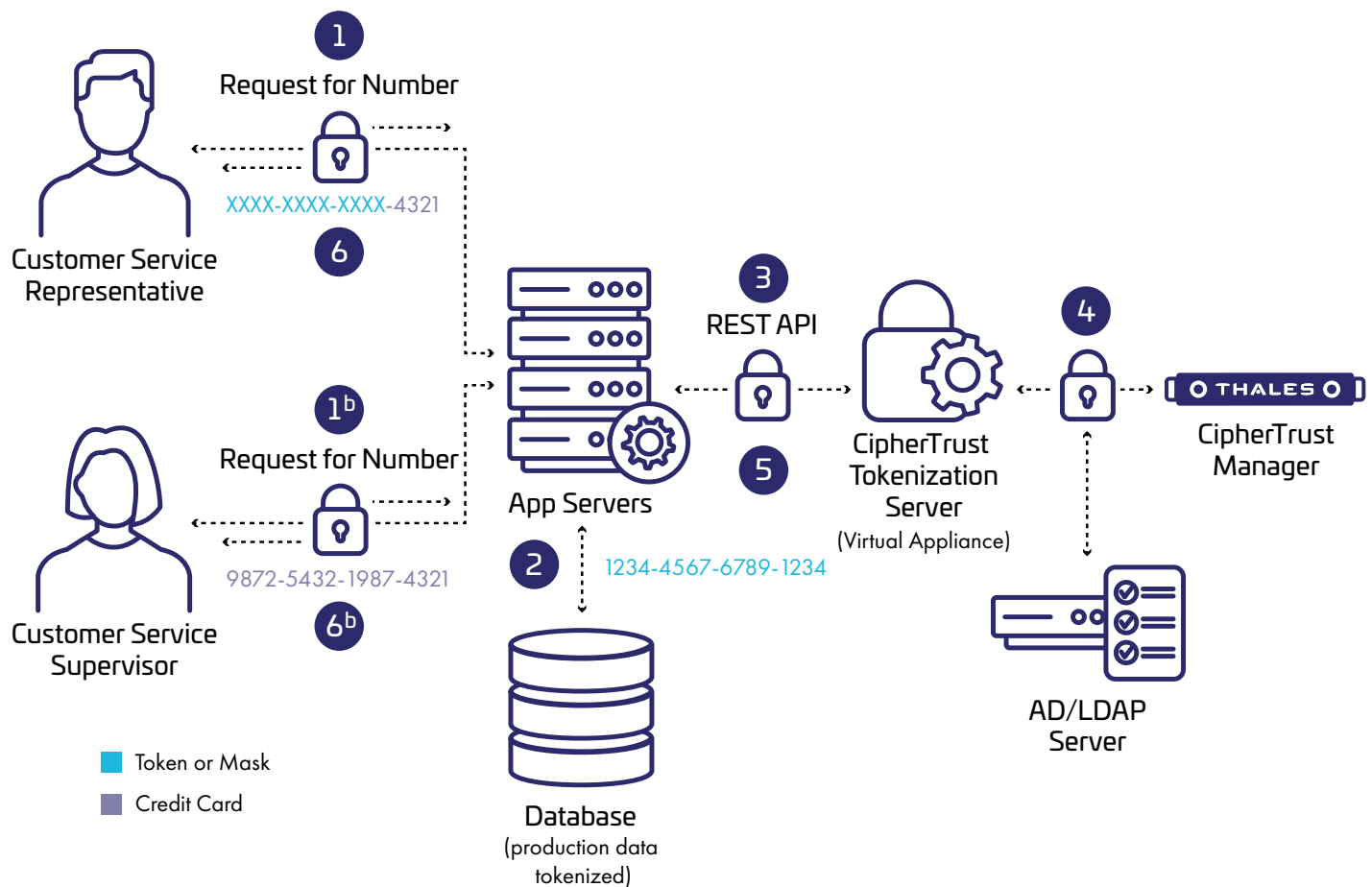
Tokenized data can be used with databases or files needed for development, QA and Test, or deployed to Big Data environments without risk to sensitive information. For credit card data, servers using tokenized information are removed from expensive and time consuming yearly audits as well. Further, Thales offers a high availability, standardsbased, FIPS 140-2 and Common Criteria validated policy management platform with the CipherTrust Manager that can provide maximum security for cryptographic keys and policies used with tokenization.

Flexible tokenization with dynamic data masking

CipherTrust Vaultless Tokenization also offers the flexibility to establish varying levels of data redaction. Security administrators can establish settings to have an entire field tokenized or to dynamically mask only part of a field, keeping the remainder in the original clear-text. When combined with LDAP-based integration to Active Directory and other directory services solutions, this capability enables uses such as call centers. One example of call center usage is limiting credit card number visibility for customer service representatives to the last four digits of a customer's credit card, while customer service supervisors can see the entire number.

The process of setting these policies is done using an intuitive administrative interface. Through the interface, administrators can choose exactly how they want data to be redacted, including specifying whether to replace a value with such characters as a pound sign, asterisk, or hyphen.

Dynamic Data Masking of Credit Cards for Customer Service



With CipherTrust Vaultless Tokenization, policies could be implemented so that a customer service supervisor could see credit card numbers in the clear, while a customer service representative would only see the last four digits.

Initial conditions:

- Credit card data in a product database is tokenized
- A policy is set for the customer services representatives group in LDAP that allows them to see only the last four digits of a credit card number
- A second policy allows members of the customer services supervisor group to see the entire number

Workflow

1. Customer service representative request to see a credit card number
2. The application calls the token from the production databases
3. The application server then passes the token and User ID information to the Tokenization Server using a RESTful API
4. The Tokenization Server calls directory services requesting group information for the User ID
5. Based on the User ID, customer services group membership, and the policy that customer services representatives can only view the last four numbers of a credit card, the token is processed, and the credit card number retrieved. Data is then dynamically masked, with the card number set to XXXX-XXXX-XXXX-4321, where 4321 are the last four digits of the actual card. XXXX-XXXX-XXXX-4321 is then returned to the application server
6. The application server displays XXXX-XXXX-XXXX-4321 to the customer services representative

Tokenization options

As well as offering the capability to dynamically mask data for display, CipherTrust Tokenization offers options for tokenizing a variety of fixed length data types – including Numeric, Text and Date formats. Options for data types to be tokenized include specific modes for:

- Numbers – Best for credit cards, U.S. Social Security numbers and other purely numeric data
- Text – All letters, all printable ASCII, All ASCII or AlphaNumeric for use with applications such as U.K. National Insurance numbers, drivers license numbers and others that include text only or mixed text and numbers
- Partial tokenization – Select only a portion of the fixed length field to be tokenized – keeping a portion of the data as the original information
- One-time use tokens – Used only when the data will never be de-tokenized
- Dates – including formats with various separators. Can also be modified with a start and end year
- Prefix – Prefixes can be appended to all tokens – for instance to designate a drivers license (DL)

Deployment and administrative simplicity

With CipherTrust Vaultless Tokenization, you can capitalize on a range of features that streamline upfront implementation and ongoing operations. CipherTrust Vaultless Tokenization

- **Enables efficient, enterprise-wide administration**

With the capabilities offered by the CipherTrust Data Security Platform, you can choose from a range of technologies and employ the mix that's optimally suited to an organizations specific projects and use cases. At the same time, enterprises gain the cost savings and operational benefits of working with solutions that can be centrally and uniformly managed. With the CipherTrust Data Security Platform from Thales, organizations can centrally manage keys for Vaultless Tokenization, as well as for CipherTrust Transparent Encryption, CipherTrust Application Data Protection, other CipherTrust products, and third-party devices.

- **Offers non-disruptive implementation**

With the solution's capabilities, you can restrict access to sensitive assets, yet at the same time, format the protected data in a way that reduces the operational impact typically associated with encryption and other obfuscation techniques. For example, organizations can tokenize a credit card field in a database, yet keep the tokenized information in a format that is compatible with associated applications. Further, tokens can appear to be real credit card numbers and pass LUHN validation, so tokenization does not break existing validation processes.

- **Eliminates manual efforts and complexity**

Tokenization occurs at the application layer, and it streamlines all the application development efforts associated with implementing tokenization in an enterprise. With the solution, developers don't have to manually institute identity management or redaction policies. CipherTrust Vaultless Tokenization offers an easy-to-use REST API for integration with the CipherTrust Tokenization Server, so that application developers can simply and quickly add tokenization and dynamic data masking to applications.

- **Provides investment protection**

Customers that have already invested in the CipherTrust Data Security Platform can get even more out of their investments. By leveraging existing CipherTrust Manager implementations (required for all CipherTrust Data Security Platform products), it is quick and cost effective to add Vaultless Tokenization through an all-software deployment.

Efficient scalability and agility

CipherTrust Vaultless Tokenization delivers the performance needed to address the operational demands of the most processing-intensive environments, enabling organizations to do millions of tokenization or detokenization operations per second. The Tokenization Server runs on virtual machines and can be quickly and efficiently scaled up and scaled down to accommodate changing workloads. For redundancy, Thales recommends deploying a cluster with at least two nodes. A three-node or four-node cluster multiplies the throughput, while providing added redundancy. Performance realized will vary depending on hardware platforms, cluster sizes, and client parallelism employed.

Key benefits

- **Reduce PCI DSS compliance effort and scope**

By leveraging CipherTrust Vaultless Tokenization, you can minimize the repositories and processes that can gain access to payment data in the clear, organizations can significantly reduce PCI DSS compliance costs and efforts.

- **Fully leverage cloud, big data, and outsourced models**

CipherTrust Vaultless Tokenization enables organizations to more fully leverage cloud services, big data models, and outsourced environments, while retaining the security controls required. For example, tokenized data can be easily migrated tokenized data to a public cloud, without exposing sensitive data to unauthorized access, even access by administrators in the cloud provider's environment.

- **Establish broad safeguards around sensitive assets**

Unlike other dynamic data masking tools, the CipherTrust solution tokenizes sensitive fields in the production database. As a result, enterprises can establish comprehensive safeguards around sensitive assets across the organization. Because the source and display data remain protected, this includes strong safeguards against cyberattacks from criminals and nation-states, as well as from insider abuse.

- **Minimize data security overhead and training**

With the CipherTrust Data Security Platform, enterprises can leverage a platform to centrally manage keys for Vaultless Tokenization and as well for the full range of the platform's encryption and access controls solutions. By leveraging a central platform that offers comprehensive capabilities, organizations can minimize the cost, effort, and training associated with managing point-products.

Sample implementations

PCI DSS 3.X Compliance

The more locations that hold primary account numbers (PANs), the more locations that fall under PCI DSS regulations. As the number of locations grow, so too do the cost and effort associated with establishing and sustaining compliance, preparing for audits, and so on. With CipherTrust Tokenization, you can tokenize PANs, and significantly restrict the number of users, processes, and repositories that can get access to this data in the clear. As a result, organizations can dramatically reduce compliance efforts and costs. Further, through such capabilities as Luhn check support, Vaultless Tokenization enables organizations to implement tokenization with minimal impact on applications and processes. In addition, the CipherTrust Data Security Platform offers many additional controls you need to address many PCI DSS requirements, including least privileged access controls, encryption, key management, and file access audit logs.

Customer contact centers

Vaultless Tokenization helps mitigate the threats that can be posed by support staff exploiting their access privileges to steal or leak sensitive data. Dynamic Data Masking can be effective both for securing sensitive data in internal contact centers and in outsourced or geographically distributed support environments. For example, enabling a support agent to see only the last 4 digits of a Social Security Number. As a result, agents can conduct any required customer verification processes, without being put in a position where they can maliciously or inadvertently leak sensitive and regulated PII.

Outsourced Development

For many organizations, application development is done by an increasingly dispersed set of teams, including geographically distributed employees as well as external contractors and service providers. While these distributed teams have helped improve the cost efficiency and agility of development efforts, they can pose significant challenges for those organizations that manage sensitive and regulated data. From initial development to testing and staging, developers need access to data repositories. If all data is left in the clear, it exposes a range of sensitive regulated data and intellectual property to any number of threats. With CipherTrust Vaultless Tokenization, organizations can quickly and efficiently provide internal and external development teams with access to corporate data repositories, while tokenizing the specific assets that pose security or compliance risks.

A complementary product, CipherTrust Batch Data Transformation, also helps organization to quickly tokenize large data sets for this purpose when needed.

Big Data

To capitalize on the opportunities presented by big data, organizations need to aggregate information from a broad range of repositories—and in the process can create massive collections of highly sensitive and proprietary data. The CipherTrust Data Security Platform enables organizations to fully leverage big data, without introducing risk to the business. Sensitive records can be tokenized before they are incorporated into big data repositories, so big data scientists, developers, and business analysts don't gain access to sensitive data that is not required for their work. At the same time, the solution's flexible tokenization options ensure that tokenized records don't disrupt big data migration, calculation, and analytics processes.

Cloud Services

When your organization moves to cloud computing models, whether infrastructure as a service (IaaS), software as a service (SaaS), platform as a service (PaaS), or hybrid approaches, you may offload a number of responsibilities and efforts, but your organization ultimately still bears responsibility for the security of its sensitive and regulated data. With CipherTrust Vaultless Tokenization, your organization can leverage the cloud models it needs, while managing tokenization policies locally. As a result, you can retain complete control and visibility over who can access sensitive assets in the clear, no matter where tokenized information ultimately resides.

Static Data Masking

Organizations can replace sensitive assets with tokens, so that outsourced application development and testing teams can run QA and analytics using realistic looking data, without having access to any sensitive assets. Once sensitive values in the database are tokenized, enterprises simply need to create a copy of the production database and give that copy to the outsourced development team. Since these teams don't have access to both the CipherTrust Tokenization Server and user credentials to detokenize, they won't be able to access sensitive data in the clear. This protects the business, maintains compliance, and enables outsource development teams to access and test realistic production data.

Just as noted above under Outsourced Development, CipherTrust Batch Data Transformation helps organization to quickly tokenize large data sets for this purpose when needed.

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.



Contact us

For all office locations and contact information,
please visit cpl.thalesgroup.com/contact-us

> cpl.thalesgroup.com <



CipherTrust Data Security Platform

CipherTrust Tokenization Server

Installation, Administration, and Programming Guide

Version 2.6.0

CipherTrust Data Security Platform
CipherTrust Tokenization Server Installation, Administration, and Programming Guide
Version 2.6.0
December 28, 2020, Document Version 2
Copyright © 2009- 2020 Thales Group, Inc. All rights reserved.

CipherTrust, Thales, and other Thales trademarks and logos are trademarks or registered trademark of Thales eSecurity, Inc. in the United States and a trademark or registered trademark in other countries.

All other products described in this document are trademarks or registered trademarks of their respective holders in the United States and/or in other countries.

The software ("Software") and documentation contains confidential and proprietary information that is the property of Thales eSecurity, Inc. The Software and documentation are furnished under license from Thales and may be used only in accordance with the terms of the license. No part of the Software and documentation may be reproduced, transmitted, translated, or reversed engineered, in any form or by any means, electronic, mechanical, manual, optical, or otherwise.

The license holder ("Licensee") shall comply with all applicable laws and regulations (including local laws of the country where the Software is being used) pertaining to the Software including, without limitation, restrictions on use of products containing encryption, import or export laws and regulations, and domestic and international laws and regulations pertaining to privacy and the protection of financial, medical, or personally identifiable information. Without limiting the generality of the foregoing, Licensee shall not export or re-export the Software, or allow access to the Software to any third party including, without limitation, any customer of Licensee, in violation of U.S. laws and regulations, including, without limitation, the Export Administration Act of 1979, as amended, and successor legislation, and the Export Administration Regulations issued by the Department of Commerce, or in violation of the export laws of any other country.

Any provision of any Software to the U.S. Government is with "Restricted Rights" as follows: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277.7013, and in subparagraphs (a) through (d) of the Commercial Computer-Restricted Rights clause at FAR 52.227-19, and in similar clauses in the NASA FAR Supplement, when applicable. The Software is a "commercial item" as that term is defined at 48 CFR 2.101, consisting of "commercial computer software" and "commercial computer software documentation", as such terms are used in 48 CFR 12.212 and is provided to the U.S. Government and all of its agencies only as a commercial end item. Consistent with 48 CFR 12.212 and DFARS 227.7202-1 through 227.7202-4, all U.S. Government end users acquire the Software with only those rights set forth herein. Any provision of Software to the U.S. Government is with Limited Rights. Thales is Thales eSecurity, Inc. at Suite 710, 900 South Pine Island Road, Plantation, FL 33324.

THALES PROVIDES THIS SOFTWARE AND DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT OF THIRD PARTY RIGHTS, AND ANY WARRANTIES ARISING OUT OF CONDUCT OR INDUSTRY PRACTICE. ACCORDINGLY, THALES DISCLAIMS ANY LIABILITY, AND SHALL HAVE NO RESPONSIBILITY, ARISING OUT OF ANY FAILURE OF THE SOFTWARE TO OPERATE IN ANY ENVIRONMENT OR IN CONNECTION WITH ANY HARDWARE OR TECHNOLOGY, INCLUDING, WITHOUT LIMITATION, ANY FAILURE OF DATA TO BE PROPERLY PROCESSED OR TRANSFERRED TO, IN OR THROUGH LICENSEE'S COMPUTER ENVIRONMENT OR ANY FAILURE OF ANY TRANSMISSION HARDWARE, TECHNOLOGY, OR SYSTEM USED BY LICENSEE OR ANY LICENSEE CUSTOMER. THALES SHALL HAVE NO LIABILITY FOR, AND LICENSEE SHALL DEFEND, INDEMNIFY, AND HOLD THALES HARMLESS FROM AND AGAINST, ANY SHORTFALL IN PERFORMANCE OF THE SOFTWARE, OTHER HARDWARE OR TECHNOLOGY, OR FOR ANY INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS, AS A RESULT OF THE USE OF THE SOFTWARE IN ANY ENVIRONMENT. LICENSEE SHALL DEFEND, INDEMNIFY, AND HOLD THALES HARMLESS FROM AND AGAINST ANY COSTS, CLAIMS, OR LIABILITIES ARISING OUT OF ANY AGREEMENT BETWEEN LICENSEE AND ANY THIRD PARTY. NO PROVISION OF ANY AGREEMENT BETWEEN LICENSEE AND ANY THIRD PARTY SHALL BE BINDING ON THALES.

Protected by U.S. patents:

6,678,828

6,931,530

CipherTrust Tokenization Server (CTS) Installation, Administration, and Programming Guide

7,143,288

7,283,538

7,334,124



Contents

Preface	i
1 Introduction to CTS	1
What is CTS?	1
CTS core services	1
Components in a CTS deployment	2
CipherTrust Tokenization Server	2
Keystore	3
LDAP/Active Directory server (optional)	3
Remote logging server (optional)	3
Load balancer (optional)	3
Tokenization benefits	3
Understanding CTS Authentication and Authorization	4
What is a CTS “user?”	4
User profiles	5
CTS authentication basics	5
AD/LDAP and user authentication	6
Client certificates and user authentication	6
CTS authorization basics	7
CTS roles	7
Key-aware permissions matrix	8
Other (non-key dependent) permissions	9
Understanding CTS user groups and permissions	10
Users, user groups, and tokenization groups	10
Understanding CTS Keys	11
Key types, algorithms, and key states	11
Key input sizes	12
Creating static and dynamic keys	12
Static work flow	12
Dynamic work flow	13
Tokenization Groups, Templates, and Masks	13
What is a tokenization group?	14

Tokenization template	14
Using data masks	15
About the CTS API	16
2 Installing and Upgrading CTS	17
Overview	17
CTS basic component architecture	17
Installation Prerequisites	19
Keystore (DSM, KeySecure Classic, or CM) requirements	19
CipherTrust Tokenization Server requirements	19
Active Directory/LDAP server requirements	21
Remote logging server	23
Load balancer requirements	23
Port configuration	23
Choose an Installation Type and Deploy	24
Install CTS as a virtual machine on VMware vSphere	25
Install CTS on bare metal	25
Install CTS in the AWS cloud	26
Prerequisites	26
Create the CTS instance	26
Configuration notes	27
Install CTS in the Azure cloud	27
Create the Azure VM	27
Configuration notes	28
Install CTS on Hyper-V	28
Install CTS on KVM	29
Install CTS in Google Cloud Platform	30
Upgrading CTS	30
Notes for upgrading a cluster	31
Upgrades to AWS instances cannot use SSN key pairs	31
Upgrade steps	31
3 Configuring the CTS System	35
Accessing the CLI	35
CLI navigation cheat sheet	36
Basic Configuration Steps	37

Configure the CTS network settings	37
Set the IP address for the CTS VM	38
Configure the CTS DNS server	39
The /etc/hosts file	40
Setting the CTS hostname	40
Importing a server certificate for TLS	41
Certificates in PEM format	41
Option 1: Regenerating a self-signed server certificate	42
Option 2: Importing an authenticated third-party server certificate (recommended)	42
Registering CTS with the DSM	43
Registering CTS with Thales KeySecure or Thales CM	44
Troubleshooting and automation tips	49
Security tips	50
Configuring CTS Nodes	50
Automated cluster setup	50
Manual cluster setup	51
Manually adding nodes to a cluster	52
Example: Manually setting up a four-node CTS cluster	53
Removing a node from a cluster	54
Configuring an AD/LDAP Server (optional)	54
Enabling the Client Authentication Feature (optional)	58
Obtaining a client CA key and certificate	59
Enabling client cert authentication with or without identities	60
Sample use in code	60
Specifying the TLS protocol version (optional)	61
Redirecting log messages to a remote log server (optional)	61
Configuring OAuth	62
Configuring ADFS and LDAP for OAuth	63
Configuring the settings.ini File for LDAP	65
AD email	65
AD DN	66
Linux LDAP DN	66
Restarting the CipherTrust Tokenization Server	66
4 CTS Administration in the GUI	69

Logging into the CTS GUI	69
Troubleshooting	70
Creating and Managing Passwords	70
Managing CTS User Groups	71
Creating user groups	71
Assigning masks	71
Adding users to user groups	72
Deleting a user group	72
Managing CTS Users	73
Creating a user manually	73
Creating basic user information	73
Updating user information	74
Deleting a user manually	74
Importing users from an AD/LDAP server	75
AD/LDAP user import prerequisites	75
Unlocking user accounts	76
Specifying DSM Keys	77
Creating a key name	77
Deleting key names	78
Tokenization Setup	79
Managing tokenization groups	79
Use case example	80
Creating tokenization groups	80
Deleting tokenization groups	82
Managing character sets	82
Creating custom character sets	82
Editing character sets	83
Applying character sets	83
Deleting character sets	84
Prototyping tokenization	84
Managing tokenization templates	84
Creating tokenization templates	84
Minimum input characters with keepleft or keepright	86
Sample: Use a tokenization template in the REST API	86
Deleting tokenization templates	86
Managing data masks	87

Creating a data mask	87
Deleting a data mask	87
Applying Permissions to Users and User Groups	88
Assigning key-based permissions	88
Assigning other permissions	89
Creating and Importing Keys	90
Other permissions: tokenize/detokenize	90
Dashboard API Monitoring	91
Exporting Data	92
Administrator History Logs	92
Viewing audit logs	93
Viewing access logs	93
Downloading (export) access or audit logs	94
Showing active sessions	94
Showing locked accounts	95
System logging	95
Configuring syslog servers	95
Backing Up and Restoring the CTS	96
Backing up the CTS in the GUI	96
Managing backup tasks	97
Restoring the CTS from a backup	97
Key Rotation	98
Monitoring via SNMP	98
5 CTS CLI Reference	99
Authentication Setup Category	99
Cluster Management Category	100
ICAPI SafeNet Setup	100
Network Management Category	101
System Configuration Category	103
VAE Configuration Category	104
CTS Configuration Commands	104
6 Encryption and Key Management API Programming Notes	107
Reference Documentation	107

Encryption and Key Management API Concepts	108
TES Resource Names (TESRN)	108
Reference keys by label	108
Other default key types	108
Legacy resource names	109
Algorithms and key types	109
Opaque objects	111
Cryptographic headers	111
Versioned keys	112
Forcing refreshed key information from DSM	113
Key states	113
Valid key operations	114
Wrapping keys	115
Importing with wrapping key	115
Exporting with wrapping key	115
Importing and exporting using clear bytes	115
Authorization	116
Creating and importing keys	116
Managing keys	116
Cryptographic operations	116
Authorization operations	117
Size	118
Payload limits	118
Date handling	118
Successful responses	119
Error responses and error codes	119
Key Management Examples	120
Key creation	120
Key import	121
Key delete	122
Key update	122
Key retrieval	123
Key export	123
Cryptographic operation examples	123
Simple SHA-256 digest example	123
Encryption examples	124

AES-256 CBC encryption example	124
Simple AES-256 CTR encryption with headers example	124
Decryption examples	125
Simple AES-256 CBC encryption example	125
Simple AES-256 CTR decryption with headers example	125
Encrypting and Decrypting Large Files	126
Sign	128
Simple HMAC-SHA256 sign example	128
Verify	129
Simple HMAC-SHA256 verify example	129
Random	129
Simple random example	129
Versioned Key Management Examples	130
Versioned key creation	130
Versioned key creation example	130
Versioned key import example	130
Standard key to version key migration example	131
Version key rotation example	132
7 Tokenization REST API	133
Using the API	134
Create a Token	134
Description	134
Create token CURL example	136
Tokenizing multiple data items	137
Get Detokenize Resource Data	138
Description	138
Detokenizing multiple data items	140
Set Log Levels	141
Description	141
Tokenization Examples	143
Java client example	143
C# client example	146
Tokenization API Call Failure Error Messages and Workarounds	148
API Reference	150

8	Key Management Service REST API	151
	Requirements	151
	Base URL	151
	Authentication	151
	Retrieve a Token Request	152
	Refresh a Token Request	152
	Operations	153
	Object Format	153
	API Reference	153
	Response Codes	153
9	Admin REST API	155
	Base URL	155
	Authentication	155
	Retrieve a token request	156
	Refresh a token request	156
	Responses	156
	Response codes	156
	Pagination	157
	Operations	157
	Searching	157
	Ordering	157
	Object format	158
	Creating objects	158
	Retrieving objects	158
	Updating objects	159
	Deleting objects	159
	Backing up CTS	159
	Example cURL Requests	160
	Unauthenticated Healthcheck	160
	Description	160
	Healthcheck example	161
	API Reference	163

PREFACE

The *CipherTrust Tokenization Server (CTS) Installation, Administration, and Programming Guide* describes how to install, configure, and manage CTS. This document also provides guidelines and some REST API documentation for developers to use when implementing CTS into their applications.

The CTS API service categories are:

- Tokenization
- Encryption (Crypto)
- Key Management
- Administration



NOTE: Additional REST API documentation is provided from the in-product help.

DOCUMENTATION VERSION HISTORY

The following table describes the documentation changes made for each document version.

Table 1: Documentation Changes

Product/Document Version	Date	Changes
2.6.0 v2	12/28/20	Corrected several API endpoint URIs that were changed to use “cts” instead of “vts,” which is incorrect. These URIs use the same “vts” component as before, to retain backward compatibility prior to the product name change.

Table 1: Documentation Changes

Product/Document Version	Date	Changes
2.6.0	11/03/20	Vormetric Tokenization Server (VTS) is now named CipherTrust Tokenization Server (CTS), to conform with several other former Thales Vormetric products adopting the brand name CipherTrust. The product version number continues to increment from the previous VTS version. Note that some names and labels still retain the historical “Vormetric” name. Support for SNMP v2c (community) and v3 has been added, with a new set of CLI commands also added to the network CLI category to manage SNMP support. Multiple syslog servers are now supported. The remote log CLI category now accepts multiple servers (specified as IP : PORT arguments) to the set and remove commands. Automated CTS cluster setup is now available in addition to manual cluster setup, for greater convenience and fewer errors. Set up and use a new apiadmin user in the cluster CLI category, and use remotejoin command. CTS is no longer supported on Docker. New installations of CTS 2.6.0 can be configured to use an EC2 key pair when cliadmin logs in via SSH instead of using a password. A healthcheck API endpoint has been added to get the current status of CTS and its individual services. JWT authentication expiration can be changed at the CLI from the default 10 minutes. This command is in the auth category. Tokenization requests can return the plaintext string together with the encrypted data in the response with the returnData parameter.
2.5.0 v2	02/25/20	Add more information about KeySecure servers, including registering VTS to KeySecure. Added explicit notes recommending authenticating with client certificates <i>with identities</i>, instead of using client certificates without identities. Several miscellaneous edits and minor text fixes
2.5.0	02/11/20	VTS now supports Gemalto Next-Gen KeySecure and SafeNet Classic KeySecure key management as an alternative to using Thales CipherTrust DSM. Added capability to send email notifications to administrators when the number of failed login attempts for a user reaches five (5). This email notification service is also used when the VTS server CPU or hard disk usage reaches 90% or higher. Added procedure for installing on KVM. Added details on installing VTS on Google Cloud Platform (GCP) The VTS Administration REST API is now available at a General Availability level.
2.4.1	10/04/19	Added support for LDAP on Linux Updated content in the CLI Reference and VTS Administration GUI chapters Added new section on configuring the settings.ini File for LDAP in the “Configuring the VTS System” chapter
2.4.0	08/27/19	Added OAuth support Added support for encrypting files larger than 5MB in size Added support for RSA 3K and 4K algorithms

Table 1: Documentation Changes

Product/Document Version	Date	Changes
2.3.0 v2	05/22/19	Fixed missing table data
2.3.0 v1	04/02/19	Added a clarification on key input sizes and their relationship with encryption modes Added information about importing and exporting keys Added new chapter: Admin REST API Added GCM code samples for encrypt, decrypt, and version key creation Added information about updated performance monitoring Updated CLI Reference with summaries of new CLI hierarchy of command categories Updated Installation and Configuration chapters
2.2.2 patch	11/13/18	• Added new topic: Create Super User • Minor edits to About Logging • Updates to describe how credentials are processed for Azure
2.2.2	10/12/18	2.2.2 documentation features and fixes, and new documentation framework and format.
2.2.0 GA	05/22/18	Update to GA version and format and style clean up.
2.2 Beta	04/9/18	Addition of crypto and key management services and APIs; redesigned GUI look-and-feel; new three-way authorization schema permitting user-based permissions to tokenize, encrypt, verify, sign, etc. with specific keys; support for client-certificate user authorization; support for custom character sets, including Chinese/Japanese/Korean (CJK) use cases; additional programmer's notes and links to external REST API documentation; all-new introduction and concepts material.
2.1.2 v2	1/29/2018	Minor updates, added notes to "Create a token group," updated install requirements section
2.1 v3	2/27/2017	Corrected VTS port configuration table, removed port 7024
2.1.1 v1	6/30/2017	Added support for VTS instances in AWS and Azure.

ASSUMPTIONS

This guide addresses three primary audiences:

- **Network administrators** who install the CipherTrust Tokenization Server (CTS) package and configure it using the CTS Administrator CLI
- **CTS GUI administrators** who assign and manage users, keys, and permissions in the CTS GUI

- **CTS application developers** who integrate CTS functionality into their application code using the CTS REST APIs

HOW TO GET HELP

For support and troubleshooting issues:

- <https://supportportal.thalesgroup.com>
- (800) 545-6608

For Thales Sales:

- <https://enterprise-encryption.vormetric.com/contact-sales.html>
- CPL_Sales_AMS_TG@thalesgroup.com
- (888) 267-3732

Introduction to CTS

The CipherTrust Tokenization Server (CTS) package is a platform-independent virtual appliance that comprises three security services and their associated tools. CTS works in conjunction with:

- A key manager (or keystore), such as Thales Data Security Manager (DSM), Thales KeySecure Classic, or Thales Next Generation KeySecure (now Thales CipherTrust Manager (CM))
- AD/LDAP servers (optionally)
- Remote logging servers (optionally)
- Load balancers (optionally)

This chapter includes the following sections:

- [“What is CTS?” on page 1](#)
- [“Understanding CTS Authentication and Authorization” on page 4](#)
- [“Understanding CTS Keys” on page 11](#)
- [“Tokenization Groups, Templates, and Masks” on page 13](#)
- [“About the CTS API” on page 16](#)

What is CTS?

This section describes the core services and components that make up CTS, as well as highlighting the specific benefits of the tokenization feature.

CTS core services

CTS offers three service categories for handling sensitive data:

- **Tokenization service:** Tokenization is used for replacing sensitive data with tokenized, or encrypted, data. Tokenized data retains the format of the original data while protecting it from theft or compromise. For example, if a company accepts customer credit card numbers, those numbers can be tokenized and stored as random-looking numbers, then detokenized by an employee or application with the proper permissions.

Tokenization is frequently used for sensitive data such as credit card numbers, social security numbers, drivers licenses, or other personally identifiable information (PII). *Data masking* can be applied to any detokenized data to hide sections of the data from different groups of users. For example, less-privileged database users might view only the last four digits of a detokenized credit card number, while a more privileged user could view the entire card number. The *CipherTrust Tokenization* REST APIs are used to integrate this functionality into a developer's application.

- **Key management services:** Using the CipherTrust Data Security Manager (DSM) or Gemalto SafeNet KeySecure as the key provider and repository, *users*-- whether human or machine-- can be granted permission to create, modify, find, import, export, and/or destroy encryption keys. The *CipherTrust Key Management* REST APIs are used to integrate this functionality into a developer's application.
- **Cryptographic services:** CTS also includes data encryption, decryption, sign, and verify services which can be tied to specific keys and user permissions. They can also be integrated into a developer's application using the *CipherTrust Cryptographic* REST APIs.

Components in a CTS deployment

A CTS deployment consists of the following components:

- CipherTrust Tokenization Server
- Keystore - Keystores that CTS supports include CipherTrust DSM (Data Security Manager), Gemalto SafeNet KeySecure Classic, and Gemalto SafeNet Next Generation KeySecure
- LDAP/Active Directory server (optional)
- Remote logging server (optional)
- Load balancer (optional)

CipherTrust Tokenization Server

The CTS server is a virtual appliance that manages the tokenization tasks for the DSM.

The CipherTrust Data Security Manager (DSM) must be installed and configured before deploying CTS.

The virtual appliance handles the following tokenization tasks:

- Receive REST API calls from the three service categories and return encrypted/detokenized data to requesting client
- Tokenize/detokenize, encrypt/decrypt, sign, and verify data using keys from the DSM
- Apply dynamic data masking as appropriate
- Authenticate users via user name/password

- Authenticate users via client certificates (optional)
- Apply user or user group profiles to determine user permissions
- Associate user permissions with particular keys

The CTS includes:

- **CTS Administrative CLI** used by the overall administrator to install, configure, and maintain the CTS server(s) and cluster.
- **CTS GUI** used by a GUI administrator to manage users, groups, permissions, data masks, and DSM-based keys.
- **REST APIs** that cover tokenization, cryptography, key management, and administration.

The server can be installed as an ISO or OVA image (including on a Hyper-V server), installed as a QCOW2 image on a Kernel-based Virtual Machine (KVM), shared as an Amazon Web Services Machine Image (AMI), or instantiated from the Microsoft Azure Marketplace or the Google Cloud Platform (GCP).

Keystore

Thales Data Security Manager (DSM), Gemalto SafeNet KeySecure Classic, or Gemalto Next Generation KeySecure.



IMPORTANT: The keystore must be installed and configured before deploying CTS. It provides and retains the keys used for tokenizing and encrypting data.

LDAP/Active Directory server (optional)

An AD/LDAP can be used to import and authenticate CTS users. It is accessed via LDAP or LDAPS.

Remote logging server (optional)

The network administrator can configure the CTS to send system log messages to a remote logging server.

Load balancer (optional)

To achieve High-Availability and scalable performance, CTS can be set up with multiple servers (nodes) to form a cluster. The cluster can operate as one coherent CTS system to be served by a load balancer.

Tokenization benefits

CipherTrust tokenization provides the following benefits:

- Replaces sensitive data in databases with tokens. This reduces the number of places in which plain text credit card numbers reside, and thus reduces the scope of complying with the Payment Card Industry Data Security Standard (PCI DSS) and corporate security policies.
- Preserves the format of data in a way that reduces the operational impact associated with encryption and other obfuscation techniques. For example, you can tokenize a credit card field in a database, yet keep the tokenized information in a format that is compatible with associated applications.
- Enables outsourcing application testing and running analytics without giving access to sensitive assets because the format of the data has been preserved. To outsource, you can create a copy of the production database and give that copy to the outsourced development team.
- Creates strong separation of duties between privileged administrators and data owners. In this way, IT administrators, such as hypervisor, cloud, storage, and system administrators can perform their tasks without access to the sensitive data residing on those systems.
- Enables dynamic data masking—the ability to establish varying levels of data redaction for different database users. For example, you can enable customer service personnel to access the last four digits of a customer’s credit card number, while an accounts payable representative can access the full credit card number.
- Integrates tokenization users with existing LDAP-based identity directories. Security teams can efficiently set granular tokenization policies for specific users and groups.

Understanding CTS Authentication and Authorization

In CTS, users are authenticated, granted permissions (to tokenize, detokenize, encrypt, hash, etc.), and associated with particular keys, using a three-way permissions matrix. It is important to understand the underlying assumptions about users, user groups, permissions, and keys in order to set up your CTS project appropriately.

What is a CTS “user?”

CTS makes no official distinctions between “human user accounts,” for real people, and “service user accounts,” representing daemons, scripts, processes, applications, or groups within applications with programmatic access to CTS services. Both are referred to as “users.”

From the point of view of the CTS system, the CTS REST APIs, and the application developer who wishes to incorporate CTS services into an application, a basic user account consists of:

- **user name/identifier**

- **password or client certificate** (If authentication is done from a client certificate, the password is not used. See [“Client certificates and user authentication” on page 6](#) for details.)

These attributes form the basis of user authentication.



NOTE: As a best practice in CTS, it is recommended to collect individual users into logical user groups and assign permissions at the group level. See also [“Understanding CTS user groups and permissions” on page 10](#).

User profiles

Users can be human or servers.

Human users

Of course, some users of CTS are real people. There are, first of all, the administrators, staff, and developers who use the CTS interface and/or the CTS APIs to install, configure, integrate, and maintain V TS services.

There are also human “end-users,” such as employees in a customer service department who may be assigned to a CTS user group authorized to detokenize credit cards and to view the last four digits of the card, based on the data mask that was applied to their user group.

Service users

However, most CTS usage consists of service users, where various servers and applications pass authentication, tokenization, encryption, and key information between themselves to complete a transaction of some kind.

For example:

A company may set up a structure to assign a financial application as User1 with a certain number of keys to perform certain operations. Within that same application, User2 might be a process that uses key2 to tokenize/detokenize credit cards, and User3 might be used to encrypt/decrypt social security numbers with key3.

CTS authentication basics

As described in [“What is a CTS “user?”” on page 4](#), authentication is based on credentials—either user name and password or a client certificate.

CTS will first seek to validate whether the credentials match entries in its local database. If not, and if AD/LDAP is configured, then it will check against AD/LDAP settings. When authentication is successful, the system will also check for CTS role assignment, any key association, and any active permissions.

AD/LDAP and user authentication

Companies can choose to use Active Directory/LDAP to store both human and service user accounts for CTS, to have a centralized way to manage them. If this approach is chosen, then the implementation involves cooperation between:

- **The network administrator**, to configure the connection between AD/LDAP and CTS;
- **The AD/LDAP administrator** and other organizational decision makers who decide which users should be imported into the CTS database, how to group them, what user group names to assign them in AD/LDAP, and what types of permissions should be granted on a group and/or individual basis;
- **The CTS GUI administrator**, who must manually enter the designated user group names into the CTS GUI before the user accounts are imported. See also [“Understanding CTS user groups and permissions” on page 10.](#)



NOTE: It is also possible to create local user accounts directly in CTS. This is useful in test or demo installations of CTS, or might be used for enhanced security, as no AD/LDAP administrator would have access to the local user credentials.

Client certificates and user authentication

Client certificates can be used for authentication. The certificate itself is a credential and can be safer than a password, as it cannot be easily shared, duplicated, written on a sticky note, or otherwise carelessly exposed.

Companies might enable simple client cert authentication, to verify that a machine has a valid cert to connect with CTS. Or they may choose a more sophisticated implementation. With CTS it is possible to use the common name (CN) inside a client cert and enter it into CTS as a user name, thereby taking advantage of the ability to assign granular permissions and/or associate particular keys to a client certificate. This is known as using a client certificate *“with identities.”*



NOTE: Thales highly recommends using client certificates with identities. If you use without identities, you must pass in a username/password along with the certificate, which can be a security vulnerability that client certificates are intended to prevent. The option to use certificates without identities is provided to maintain backward compatibility with the earlier basic authentication flow.

To implement client certs with granular permissions, an organization would need to:

- Have established an in-house Certificate Authority (CA)
- Have an exportable CA cert
- Issue any number of client certs (signed by the CA) that each contain a common name (CN)

- Go to the CTS CLI to:
 - Upload the CA cert to CTS
 - Enable using client certs with ID (See [“Enabling the Client Authentication Feature \(optional\)” on page 58.](#))
- Use the CTS graphical user interface (GUI) for:
 - Creating users and entering common names as user names (or import from Active Directory for LDAP)
 - Managing user permissions (If importing from LDAP, you can use the CTS CLI setting to designate a user group as the “active” user group thereby activating all members of the group.)
 - Associating key(s) with the cert/user
 - Assigning permissions to the cert/user



NOTE: CTS requires that all client certificates be in PEM format.

CTS authorization basics

CTS employs a multi-layered security model with separation of duties and granular permissions settings built in to it. It is also flexible, adaptive to individual corporate structures and policies. This structure implies some inherent complexity, but the concepts below form the foundational assumptions.

CTS roles

When a user is created in CTS, it can be assigned one or more of the following core CTS roles:

- **Superuser:** Can perform both read and write actions in the CTS GUI.
Superusers are GUI administrators who manage users, user groups, keys, tokenization groups, tokenization templates, data masks, and character sets. Superusers associate users with key(s) and assign the crypto, key, and tokenization permissions to other users. Superusers are automatically also “Staff.”
- **Staff:** Users with read-only permission to the CTS GUI.
- **Active:** Users must be “active” in order to have encryption, key, or tokenization permissions. Active users are those authorized to use the CTS REST APIs.

The role-based permissions (Active/Staff/Superuser) are incremental. With no role assigned, the user cannot access any CTS functionality. (This is equivalent to being deleted, but the information is retained in the database for possible reactivation.)



NOTE: When users are created directly in the CTS GUI, they are automatically assigned the status “active,” which can be disabled as needed.

If users are imported from AD/LDAP, the “active” designator is handled differently. See [“Understanding CTS user groups and permissions” on page 10](#) for details.

As of CTS 2.6.0, if Active is unchecked, the user cannot log in to the GUI even if assigned Superuser/Staff status.

Key-aware permissions matrix

The GUI Administrator can assign permissions based on user (or user group), action, and key, similar to the English sentence:

“User CTSuser1 may encrypt, decrypt, tokenize and detokenize with symmetric key Sym_key1.



Figure 1: Permissions available on symmetric keys



NOTE: Permissions can be assigned at the user and the user group level, and are additive. See also [“Understanding CTS user groups and permissions” on page 10](#).



Different types of keys (symmetric, asymmetric, and opaque objects) have different permissions available. See also [“Understanding CTS Keys” on page 11](#).



Tokenize and detokenize functionality require a couple of additional steps. See also [“Tokenization Groups, Templates, and Masks” on page 13.](#)

Other (non-key dependent) permissions

Users or user groups can be authorized to dynamically create and/or import key names and key material on the DSM, under the “Other Permissions” column of the Permissions main page.



NOTE: See also [“Creating static and dynamic keys” on page 12.](#)

- **Create** key: creates the key name and key material in the DSM; the key name will be listed in the CTS GUI.
- **Import** key: used to import any key from one DSM to another, or from an external system into the DSM. The imported key name will be listed in the CTS GUI.

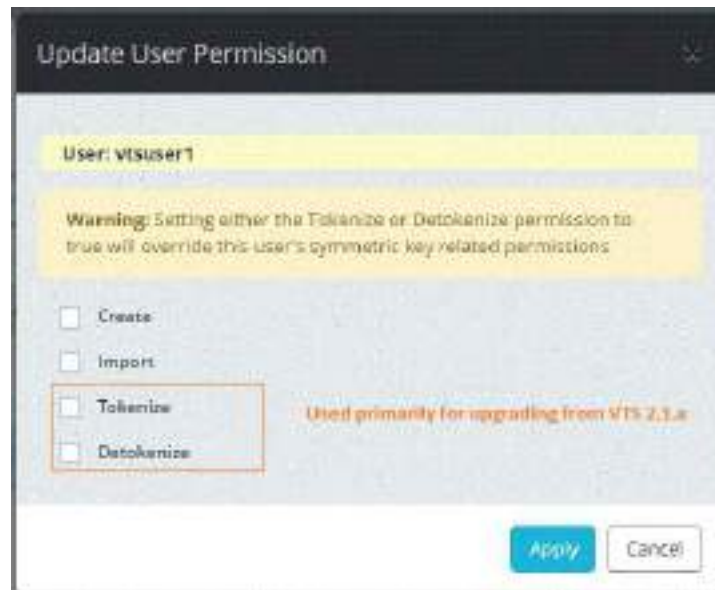


Figure 2: “Create” and “Import” keys are distinct from the legacy Tokenize/Detokenize function



NOTE: The *Other Permissions* assignment window also includes check boxes for **Tokenize** and **Detokenize**, along with a warning about overriding symmetric key assignments ([Figure 2](#)).

This Tokenize/Detokenize feature is used during upgrade from earlier versions of CTS, as it grandfathers in the global tokenize/detokenize feature that was used previously.



NOTE: See [“Other permissions: tokenize/detokenize” on page 90](#) for usage tips.

Understanding CTS user groups and permissions

- **CTS role-based user groups:**

The three core CTS roles (Superuser/Staff/Active) are also three required core CTS user groups. An organization may choose to assign different group names, but the meanings are retained (see [“CTS roles” on page 7](#) for the definitions).

CTS GUI administrators must create at least three user groups that correspond to the roles Superuser/Staff/Active.

- **Assigning users to user groups two different ways:**

If importing from AD/LDAP, users will be sorted into their role groups from the CLI.

If entering users directly in the GUI, the administrator will assign the role(s) on the user’s page and then check the appropriate group(s).

- **Adding more user groups:**

Additional groups can be added to CTS for organizational purposes.

- **Assigning group permissions:**

Key-based and non-key-dependent permissions can be assigned at the user group level or the user level, but assigning at the group level is recommended.

- **Permissions are additive:**

Individual and group permissions are additive. For example, suppose Joe is a member of the user group “Finance,” which has permission to tokenize credit card numbers, and Joe is also a team lead granted user-level permission to detokenize. In total, Joe can tokenize and detokenize.

Users, user groups, and tokenization groups

To allow for abbreviated Tokenization REST API calls and to enable division of an organization into sub-organization either by location or department, CTS adds a so-called “tokenization group” concept. So, there could be a user group “Finance” in San Francisco, and another “Finance” user group in Paris, each associated with its own tokenization group.



NOTE: See also [“Tokenization Groups, Templates, and Masks” on page 13](#).

Understanding CTS Keys

CTS supports three types of keys: **symmetric**, **asymmetric**, and **opaque objects**, each of which uses different algorithms and can be used for different actions.

User permissions (to encrypt, decrypt, sign, verify, hash, or tokenize/detokenize) must be tied to particular keys. The *CipherTrust Key Management APIs* provide an array of static key management functions, as well as the ability dynamically to create or import keys to the DSM.

Key types, algorithms, and key states

The application developer, the DSM Administrator, and the CTS GUI Administrator work together to implement key handling in CTS.

Table 1: Key Summary Table

Key Type	Description	Algorithms	State
Symmetric	Oct or octet sequence. Used to: • tokenize • detokenize • encrypt • decrypt Symmetric keys are also used for tokenization, and while the symmetric keys are typical AES keys with 128 or 256 bits of key material, the algorithms are FPE (FF3) and FF1. Note: Versioned keys may not be used for tokenization, only non-versioned keys.	A128ECB - AES 128 ECB cipher mode A256ECB - AES 256 ECB cipher mode A128CBC - AES 128 CBC cipher mode A256CBC - AES 256 CBC cipher mode A128GCM - AES 128 GCM cipher mode A256GCM - AES 256 GCM cipher mode A128CBCPAD - AES 128 CBC-PAD cipher mode A256CBCPAD - AES 256 CBC-PAD cipher mode A128CTR - AES 128 CTR (counter) cipher mode A256CTR - AES 256 CTR (counter) cipher mode	At this time, only symmetric keys support states. On create and import, only a preactive or active state may be given. On a key update, all states can be given except for preactive. The following states are available: • Preactive • Active - Default when a key is created • Suspended • Compromised • Deactivated • Destroyed
Asymmetric	RSA key type. Used for: • encrypt • decrypt • sign • verify	RSA	N/A

Table 1: Key Summary Table

Key Type	Description	Algorithms	State
Opaque Object	Octet array. This is a Thales extension (not part of JOSE RFC-7518 spec). Used for • sign • verify Does not have the option of store on server.	HS1 - HMAC-SHA-1 HS224 - HMAC-SHA-224 HS256 - HMAC-SHA-256 HS384 - HMAC-SHA-384 HS512 - HMAC-SHA-512	N/A

Key input sizes

For CBC and ECB modes of encryption, the input size should be a multiple of the block size. Note that the AES block size is 16 bytes (128 bits).

If you do not use an input length that is a multiple of the block size, use CTR or CBC-PAD encryption.

Creating static and dynamic keys

Create keys with either a static or dynamic work flow:

Static work flow

In most cases, it is recommended to implement your application using static keys — that is, keys that will change their key material rarely, and thus are rarely created or destroyed in the DSM.

The implementation tasks for static key creation are:

- Decide which actions will be performed (encrypt/decrypt, tokenize/detokenize, sign/verify), and which types of keys will be needed to support those actions.
- **DSM Administrator:** Create CTS-specific keys in the DSM, giving them meaningful names.
NOTE: it is recommended to keep tokenization keys separate from other encryption/decryption keys.
- **CTS GUI Administrator:** Enter the appropriate key names into the CTS GUI, to be stored in the local CTS database and used for assigning encryption and tokenization permissions.
- **Application Developer:** Use the CipherTrust Key Management, Cryptography, and Tokenization APIs to perform the necessary functions in the application being integrated with CTS.



Do not use the REST API `create` or `import` key functions.



IMPORTANT: It is recommended to use a static key model with tokenization.

Dynamic work flow

In some cases, an organization may decide to automate key creation, rotation, or key import in the DSM, using the appropriate *CipherTrust Key Management* APIs.



If a key is deleted, all of the ciphertext the key was instrumental in creating is rendered unreadable.

In this case, the implementation team would need to:

- Decide which actions will be performed (encrypt/decrypt, tokenize/detokenize, sign/verify) and which types of keys will be needed to support those actions.
- **Application Developer:** Use the *CipherTrust Key Management* and *Cryptography* APIs to perform the necessary functions in the application being integrated into CTS, including `create` and/or `import` key functions.
- **CTS GUI Administrator:** Designate a user (possibly a non-human user) and grant the user permission to create and/or import keys.

Tokenization Groups, Templates, and Masks

In tokenization, sensitive data—such as credit card numbers or social security numbers—are replaced with an encrypted “*token*” which retains the format of the original data. When detokenizing, different users can be authorized to view the entire detokenized/unencrypted value, or authorized to see only a part of the value, with the rest hidden by a *data mask*.

CTS includes tokenization groups, templates, and data masks as convenience features to make implementing tokenization easier.

What is a tokenization group?

A *tokenization group* is a way to compartmentalize CTS at a department level or site level. Each tokenization group has a symmetric key associated with it. While the key management and encryption services refer to keys directly, the `tokenize/detokenize` services refer to a key indirectly – via the tokenization group.



NOTE: See also [“Managing tokenization groups” on page 79](#).

Figure 3: Have at least one symmetric key created when you create a tokenization group

The screenshot shows a 'New Tokenization Group' dialog box. It has a title bar with the text 'New Tokenization Group' and a close button. The dialog contains two fields: 'Name*' with a text input field containing the text 'name', and 'Key*' with a dropdown menu showing 'Sym_key1'. At the bottom right of the dialog are two buttons: 'Create' and 'Cancel'.



NOTE: Remember that all symmetric keys used for tokenization must be unversioned keys.

Tokenization template

Tokenization templates are used to supply parameters to the tokenization/detokenization operation which otherwise would have to be supplied individually in the REST API call, thus making the REST API call both easier to use and shorter. The CTS GUI administrator collaborates with the application developer to create tokenization templates with the appropriate settings, and to give the template a name that developers will use in their code.



NOTE: See also [“Managing tokenization templates” on page 84](#).

Figure 4: Create tokenization templates after creating tokenization groups and character sets.



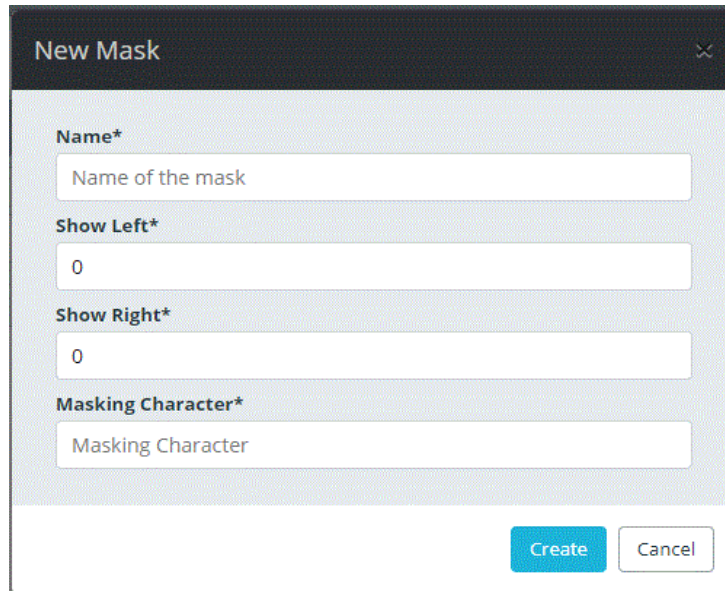
Using data masks

When a user or user group is given detokenization privileges, a data mask must be applied that defines how much of the unencrypted data they are allowed to see. To create a mask that shows the entire data string, the CTS GUI Administrator would enter very large values in the **Show Left** and **Show Right** fields (999999).



NOTE: See also [“Managing data masks” on page 87](#).

Figure 5: New data mask creation.

A screenshot of a 'New Mask' dialog box. The dialog has a dark header bar with the title 'New Mask' and a close button. Below the header, there are four labeled input fields: 'Name*' with a placeholder 'Name of the mask', 'Show Left*' with a value of '0', 'Show Right*' with a value of '0', and 'Masking Character*' with a placeholder 'Masking Character'. At the bottom right of the dialog are two buttons: 'Create' (highlighted in blue) and 'Cancel'.

About the CTS API

CTS includes the following sets of REST API services: Encryption, Key Management, Tokenization, and Administration.

Tokenization/detokenization are simple-to-use REST APIs which provide format-preserving encryption and proprietary "RANDOM" tokenization, whereas the Encryption APIs offer a plethora of traditional non-format-preserving encryption methods, as well as two format-preserving ones (FF3 and FF1)

Encryption and Key Management APIs are described in the chapter ["Encryption and Key Management API Programming Notes" on page 107](#), and in an online reference guide. The Tokenization APIs (including the reference documentation, examples, and error codes) are described in the chapter ["Tokenization REST API" on page 133](#). The Administration APIs are described in ["Admin REST API" on page 155](#).

These APIs are also described in an online reference guide, under **Documentation** in the CTS Administration GUI navigation bar.

Installing and Upgrading CTS

This chapter describes how to install CTS. It is used by the **CTS network administrator** and contains the following sections:

- [“Overview” on page 17](#)
- [“Installation Prerequisites” on page 19](#)
- [“Choose an Installation Type and Deploy” on page 24](#)
- [“Upgrading CTS” on page 30](#)

Overview

The CTS solution consists of the following required components and optional components:

- CipherTrust Tokenization Server (CTS)
- A key manager (keystore), such as Vormetric Data Security Manager (DSM), Thales KeySecure Classic, or Thales CipherTrust Manager (CM)
- Active Directory (AD)/LDAP server (optional)
- Remote Logging server (optional)
- Load Balancer (optional)



NOTE: This chapter focuses on installing the CTS, with additional sections about upgrade, backup, and recovery. Chapter 3, [“Configuring the CTS System” on page 35](#), describes the follow-up configuration steps the network administrator performs to connect all the components and establish baseline settings.

CTS basic component architecture

CTS can be installed as a virtual machine (the more typical case) or as software on bare metal.

The CipherTrust Tokenization Server does most of the tokenization work and acts as a server for the additional cryptographic and key management API calls and features. It uses the other components for

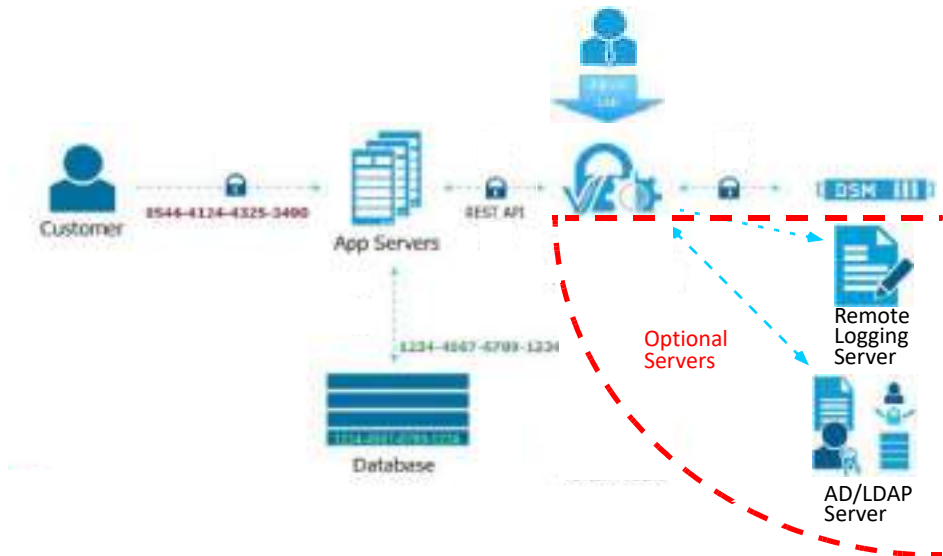
specific tasks. It obtains encryption keys from the DSM and it uses those encryption keys to encrypt sensitive data. For users created in the AD/LDAP server, the Tokenization Server uses the AD/LDAP server to validate those users when they request data to be tokenized, detokenized, encrypted, etc. Users can also be created with the CTS GUI. These users are validated by the CTS itself.

Once the system has been installed and configured, the CTS supports the RESTful API calls from applications. Log messages can optionally be sent to a remote logging server.



IMPORTANT: A minimum of two nodes is recommended. This configuration ensures that the data is retained online, and provides for optimum availability. For higher performance, use a cluster with more than two nodes. It is also recommended to use a load balancer to automatically distribute load to all nodes in the cluster.

Figure 6: CTS Component Architecture (with DSM as keystore)



Installation Prerequisites

The CTS solution requires installing multiple components. The following sections describe each component and the information you must gather before integrating them into the CipherTrust Tokenization solution. The integration steps themselves are described in [“Choose an Installation Type and Deploy” on page 24](#).

- [“Keystore \(DSM, KeySecure Classic, or CM\) requirements” on page 19](#)
- [“CipherTrust Tokenization Server requirements” on page 19](#)
- (Optional) [“Active Directory/LDAP server requirements” on page 21](#)
- (Optional) [“Remote logging server” on page 23](#)
- [“Load balancer requirements” on page 23](#)
- [“Port configuration” on page 23](#)

Keystore (DSM, KeySecure Classic, or CM) requirements

A Vormetric DSM, KeySecure Classic, or CipherTrust Manager (CM) server provides the CipherTrust Tokenization Server with encryption keys.

For specific version compatibility requirements with CTS, check the *Vormetric/CipherTrust Data Security Platform DSM | VAE | CTS | VTS | VKM | VPTD | BDT Compatibility Matrix* document.

Note that CTS does not require a dedicated keystore ; it can be shared with other VDS applications. To install a new DSM, see the *CipherTrust Data Security Manager (DSM) Installation and Configuration Guide*. Refer to the corresponding documents for installing a new Thales KeySecure Classic or Thales CipherTrust Manager (CM) server.

Gather this keystore information for system configuration:

1. DSM , KeySecure, or CM server fully qualified hostname: _____
2. DSM , KeySecure, or CM IP address (or addresses, if multiple clustered servers): _____

CipherTrust Tokenization Server requirements

CipherTrust Tokenization Server can be installed on a physical server (bare metal) or as a virtual machine.

To run as a virtual machine, CTS supports the following hypervisors:

- VMware ESXi 6.0 or higher

- Amazon AWS EC2
- Microsoft Azure
- Microsoft Hyper-V
- Linux Kernel-based Virtual Machine (KVM)
- Google Cloud Compute Engine (GCP)

CTS builds are available in the following formats:

- ISO -- for physical machines, VMware ESXi, or Hyper-V
- OVA -- for VMware ESXi
- VHD -- for Hyper-V and Azure
- AMI -- for Amazon AWS
- qcow2 -- for KVM
- GCP -- for Google Cloud Compute Engine

Minimum system requirements:

RAM:

- 16GB minimum for RANDOM mode = none
- 24GB for RANDOM mode = numeric or alphanumeric
- 32GB for for RANDOM mode = both numeric and alphanumeric

CPU:

- 4 CPUs, x86, > 2.0Ghz recommended

Disk:

If CTS is installed as a virtual machine, the disk size is automatically created and allocated when the virtual machine is created from the build image.

If CTS is installed on a physical machine, a minimum of 85GB disk is required. ISO installation will fail if the disk is less than 85GB. Installation will also fail if RAM is less than 16GB.

Networking:

CTS supports DHCP or static IP addresses. IPv6 is not supported; only IPv4 is supported at this time.

CTS supports DNS or static entries in the `/etc/hosts` file. It is recommend to use DNS if available.

Gather this CipherTrust Tokenization Server information for system configuration:

1. Tokenization Server .ova image, .iso image, Amazon AWS ID, or Azure ID:
-

2. Tokenization Server system hostname:
-
3. Tokenization Server IP address:
-
4. Tokenization Server subnet prefix length:
-
5. Tokenization Server default gateway:
-
6. DNS Server IP address:
-



NOTE: If you don't use DNS to resolve hostnames, you can modify the /etc/hosts files. See [“The /etc/hosts file” on page 40](#).

Active Directory/LDAP server requirements

An optional AD/LDAP server can be used to generate users and authenticate their credentials when they request services.

See [“Understanding CTS Authentication and Authorization” on page 4](#) and [“AD/LDAP and user authentication” on page 6](#) for a conceptual basis on how AD/LDAP is used.

Requirements:

- Access to the AD/LDAP server containing users and groups that will have CTS privileges.
- For LDAPS, the LDAP server's CA certificate must be imported into the Tokenization Server.
- If you use the three example Active Directory groups below (ctsUsers, ctsManager, and ctsSuperuser), create these groups before completing the steps described in [“Configuring an AD/LDAP Server \(optional\)” on page 54](#).

Gather this AD/LDAP information for system configuration:

Table 2: AD/LDAP Configuration Information

Name	Description	Your Site Information
LDAP Server URI	Hostname, port and protocol of LDAP server. Example: ldaps://ads.example.com:636	

Table 2: AD/LDAP Configuration Information

Name	Description	Your Site Information
binding DN	FQDN of user that has access to the LDAP server Example: cts@ads.example.com	
binding password	The credential of above user Example: H838%i2hz9*%^86&^*	
user search scope	Search criteria of user object in AD/LDAP server Example: dc=corp,dc=example,dc=com	
user group search scope	Search criteria of user group inside AD/LDAP Example: cn=Users,dc=corp,dc=example,dc=com	
user search filter	Name of user field in AD/LDAP Example: sAMAccountName for AD, uid for LDAP	
Tokenization Server active user group	Group that user can access Tokenization Server RESTful API Example: cn=ctsUsers,cn=User,dc=corp,dc=example, dc=com	
Tokenization Server super user group	Group that can access the GUI and REST APIs. Example: cn=ctsSuperuser,cn=User,dc=corp,dc=example, dc=com	
group member selector	<u>UNIX/Linux LDAP only.</u> The LDAP attribute which designates a group member within a group. Example: member or memberUID	
group object class	<u>UNIX/Linux LDAP only.</u> The LDAP attribute that identifies a group of users. Example: posixGroup or groupOfNames	
user prefix	<u>UNIX/Linux LDAP only.</u> The LDAP attribute to be used to prefix the username when searching for username in the LDAP database Example: uid, cn, sn	
CA Certificate	For LDAPS only: Access to root and intermediate CA certificates for the LDAPS server in .pem file format. May require a Certificate Management tool	

Remote logging server

Optionally, you can redirect log messages to a remote logging server.

Gather the following remote logging server information for system configuration:

1. Remote logging server name or IP address:_____
2. Remote logging server port (TCP):_____

Load balancer requirements

A cluster of CTS nodes can be installed and run as one coherent CTS system using a load balancer. The load balancer acts as the main entry point for tokenization and crypto services to distribute load across all nodes in the cluster. This provides scalability and reliability of the system.

CTS supports the following methods of load balancing:

- Round-robin -- requests are distributed in a round-robin fashion.
- Least-connected -- allows the load balancer to control the load on some CTS nodes.
- IP-hash -- the client IP address is used to determine what server in a server group should be selected for the client request. This can be used for client's session sticky or persistent.

The load balancer must be able to use TLS1.1 and higher.

Port configuration

If the CipherTrust Tokenization Server must communicate with other components through a firewall, open the ports in the firewall as shown in the following tables. [Table 3](#) shows incoming traffic, and [Table 4](#) shows outgoing traffic.



NOTE: For a complete list of ports required for the DSM, see the *DSM Installation and Configuration Guide*.

[Table 3](#) describes each port you must open for incoming communication to the CTS.

Table 3: Incoming Ports to Configure for CipherTrust Tokenization Server

22	TCP	Management Console →CTS	Administration CLI SSH access.

Table 3: Incoming Ports to Configure for CipherTrust Tokenization Server

443	TCP	Browser → CTS Requester → CTS CLI → CTS	HTTPS access for Administration GUI and REST/JSON API. Download the zip file to perform upgrade.
5432	TCP	CTS ↔ CTS	This port is required if a cluster of more than one node is used. This port is used exclusively for communication between the CTS nodes in a cluster. If the cluster of nodes is inside a VPC or subnet, access to port 5432 should only be confined within the subnet or VPC. It is recommended that this 5432 port is firewalled off from anything other than the cluster nodes.

[Table 4](#) describes each CTS port you must open for outgoing communication from the CTS.

Table 4: Outgoing Ports to Configure for CipherTrust Tokenization Server

User-defined	TCP	CTS → Log Server	Remote logging
123	UDP	CTS ↔ NTP Server	Optional network time synchronization
389	TCP	CTS → LDAP Server	Optional LDAP connection
636	TCP	CTS → LDAPS Server	Optional LDAPS (LDAP over TLS) connection
8080	TCP	CTS → DSM	Default TCP/IP port for HTTP that is used once to perform the initial certificate exchange between a CTS host and DSM.
8443	TCP	CTS → DSM	TCP/IP port through which the CTS communicates with the DSM to exchange configuration. The CTS establishes a secure connection to the DSM via certificate exchange using this port.
8444	TCP	CTS → DSM	Upload CTS log messages to DSM
8446	TCP	CTS → DSM	Configuration exchange (policy/key pull) using Elliptic Curve Cryptography (Suite B)
8447	TCP	CTS → DSM	CTS uploads log messages to DSM using Elliptic Curve Cryptography

Choose an Installation Type and Deploy

There are several options for the network administrator to install the CipherTrust Tokenization Server:

- [“Install CTS as a virtual machine on VMware vSphere” on page 25](#)
- [“Install CTS on bare metal” on page 25](#)
- [“Install CTS in the AWS cloud” on page 26](#)

- [“Install CTS in the Azure cloud” on page 27](#)
- [“Install CTS on Hyper-V” on page 28](#)
- [“Install CTS on KVM” on page 29](#)
- [“Install CTS in Google Cloud Platform” on page 30](#)

Install CTS as a virtual machine on VMware vSphere

CTS can use a VMware ESXi hypervisor to create a virtual machine using the ISO or OVA build image. CTS supports ESXi version 6.0 or higher.

Refer to VMware ESXi documentation on how to create a VM using an ISO or OVA image.

It is recommended to use Thick Provisioning for hard disks.

For a multi-node setup, you can repeat this installation process for additional machines. You would then configure each node separately and join them in a cluster as described in [“Configuring CTS Nodes” on page 50](#).

To begin configuration, click the **Console** tab of the vSphere Client to log in to the CTS CLI as `cliadmin`, change the default password, accept the Service Level Agreement, and begin configuring CTS settings using the CTS CLI (see [“Configuring the CTS System” on page 35](#)).

Install CTS on bare metal

Although typically installed as a virtual machine, the CipherTrust Tokenization Server is also provided as an ISO image that can be installed on a bare metal host.

A bare metal host is a computer that does not run hypervisor software. Do not install the operating system on the host before installing CTS software, because the OS is included along with the CTS software in the installation package. When selecting the operating system, specify 64-bit CentOS version 7 or higher.

Make sure the computer meets the following minimum system requirements:

- Number of CPU cores: 4
- RAM: 15 GB minimum if RANDOM mode is none. 24 GB recommended if CTS will be used to tokenize credit card numbers longer than 16 digits. 24 GB if using numeric RANDOM mode or alphanumeric RANDOM mode, or 32 GB if using both modes.
- Hard disk space: 85 GB

To do a bare metal installation:

1. Obtain the installation file `cts-<release>.iso`.
2. Refer to the machine's documentation on how to boot and install from an ISO image.

Install CTS in the AWS cloud

Prerequisites

The following components are necessary for running a CTS in the cloud on Amazon Web Services.

- A VPC (Amazon's *Virtual Private Cloud*). If you plan to create a cluster of CTS machines, it is recommended that your cluster nodes use a VPC so that they communicate only with each other within the private cloud. Cluster communication does not need to be, and should not be, exposed on the Internet.
- A Subnet. If you choose to use a VPC, you will need to create a subnet and associate that subnet to that VPC. This subnet is where your CTS cluster nodes will get the IP addresses.
- A Security Group. Set your security group with the following minimum settings:

Incoming port 443 - Web and REST API access

Incoming port 22 - CLI access

Incoming port 5432 - Optional. Set this only if the machines in your cluster are not using a VPC to communicate with each other.

If you will be accessing your CTS strictly inside the VPC, you will still need a security group, but you can block all incoming ports. If your DSM is located inside the same VPC, you can block both incoming and outgoing ports.

- For best security, use SSH key pairs instead of passwords for SSH logins. If you choose to use an EC2 key pair when creating a CTS instance, the `cliadmin` will use that key (a private key) to log in by SSH. Password access will be disabled.

If you choose to not to use an EC2 key pair, the `cliadmin` will use a password to log in by SSH.

Upgrading an EC2 instance from VTS v2.5 or earlier retains the use of passwords for the `cliadmin`. Only newly-created instances from CTS v2.6 and above can use the SSH key pair in AWS.

Create the CTS instance

Follow this procedure to create the CTS.

1. Choose your AWS region.
2. Locate the AMI of the CTS image.

3. Select the AMI image and launch an instance.
4. Select Instance type: m4.xlarge - with recommended minimums of 4 vCPUs and 16GB memory.
5. Select the VPC to use for the installation.
6. Select the subnet.
7. Select **Auto-assign public IP**, if applicable.
8. Set the primary IP address. This will be the IP address of the VPC subnet, and it will be used by the VM for its cluster node IP address.
9. In the **Add Storage** menu, select delete on termination, if applicable. Do not change the size of the root volume.
10. Choose whether to have cliadmin log in via SSH with a password or with an EC2 key pair.

If you choose not to use an EC2 key pair, the cliadmin will use a password to log in by SSH.

If you choose an EC2 key pair, the cliadmin will use that key (private key) to log in by SSH. Password access will be disabled. You are prompted to select an existing key pair. Select the private key file (in PEM format), and make sure to confirm your selection before launching the new instance. Note that if an instance is configured to use an EC2 key pair, password access is blocked, and all password-related CLI commands are disabled.

Configuration notes

To begin configuration, ssh into the instance as cliadmin user, using the public IP address or the VPC IP address, and follow the steps in [“Configuring the CTS System” on page 35](#).

If you will be creating a cluster of CTS machines, create an instance for each node in the cluster. You would configure each node separately and join them in a cluster as described in [“Configuring CTS Nodes” on page 50](#). Note that when configuring a multi-node cluster, you will use the **VPC IP** address for the node_IP_address.

Install CTS in the Azure cloud

Create the Azure VM

1. Log in to the Azure web portal at <https://portal.azure.com>.
2. Go to the *Marketplace* screen and search for **CipherTrust Tokenization Server**. Select the desired version and click the **Create** button.
3. Supply the following information to create a virtual machine:
 - **VM name:** name of the VM you want to use, for example, my-CTS-2.3.0-azure-vm
 - **VM disk type:** SSD or HDD.

- **User name:** Not required by CTS but Azure needs this data to create a VM. This user name entry is not retained by CTS, and is not used as a CTS credential.
- **Authentication type:** Also not required by CTS but required for Azure to create a VM. This authentication type entry is not retained by CTS, and does affect the CTS configuration.
- **Resource group:** You may create a new resource group or use an existing one. If you plan to create a cluster of CTS machines, it is recommended that all cluster nodes use the same resource group, so that they can communicate with one another without having to use a public static IP address or enable an incoming port in the firewall.
- **Location:** For example, “West-US”
- **Subnet and subnet address range**
- **Static or dynamic public IP address:** Use a static IP address if you expect your cluster to span multiple resource groups

Configuration notes

To begin configuration, ssh into the instance as the `cliadmin` user using the public IP address, and follow the steps in [“Configuring the CTS System” on page 35](#).

Note that on Azure, you will not [“Configure the CTS network settings” on page 37](#). Azure automatically configures these parameters for the CTS VM:

- Internal and subnet IP addresses are provided to the VM through DHCP.
- For CTS clusters, create an instance for each node in the cluster. Then configure each node separately and join the nodes in a cluster.
- When creating or joining a cluster, if you have all the cluster nodes in the resource group, you must use the same **internal or subnet IP** address used by the VM.
- When adding a node to the cluster, you must always use the internal or subnet IP address of the new CTS VM node.

The network security group defines the firewall rules of the CTS virtual machine. At a minimum, incoming ports must allow port 22 (SSH) and port 443 (HTTPS).

If your cluster nodes are located in different resource groups, you also need to add port 5432 to the allowed incoming ports. However, if all your cluster nodes are in the same resource group, do not add port 5432, because nodes inside a resource group do not need any firewall permissions to communicate with one another.

Install CTS on Hyper-V

Install the CTS on a Hyper-V server using an `.iso` or VHD file.

Download the `.iso` or VHD file to the installation server.

Use the following steps to install the CTS on a Hyper-V server using an .iso or VHD image file.

1. Open Hyper-V Manager.
2. Select **Action > New > Virtual Machine**.
3. Choose **Generation 1**.



NOTE: Generation 2 is not supported.

4. Set the start up memory to 16GB.
5. Choose a network adapter.
6. For the VM, choose one of the following based on the type of file used:
 - .iso image - Use an existing hard disk: Select CTS VHD image file you've downloaded or copied.
 - VHD file - Install an operating system from a bootable CD/DVD-ROM: Navigate to and select the downloaded .iso file.
7. Select **Finish** to create the Virtual Machine
8. For .iso only: Select and start the Virtual Machine. This VM boots the mounted ISO image file and the CTS installation begins.

Install CTS on KVM

To install CTS on a Linux Kernel-based Virtual Machine (KVM):

1. Download the CTS image in qcow2 format. Place this image file into your KVM images directory.
2. Create a KVM virtual machine by importing the CTS qcow2 image. Use the following KVM parameters:
 - virt-type: kvm
 - os-type: linux
 - os-variant: centos7.0
 - vcpus: 4
 - ram: 16000
 - controller: type=scsi,model=virtio-scsi
 - disk: /var/lib/libvirt/images/cts-2.x.n.nnn.qcow2,bus=scsi
3. Connect to the console and log in as user cliadmin.

Install CTS in Google Cloud Platform

To install the CTS on the Google Cloud Platform (GCP):

1. Contact Thales Engineering Support and provide your GCP account for access to CTS images.
2. Log in to the GCP console and navigate to **Compute Engine > VM Instances > Create Instance**.
3. Set a name, region, and zone for your instance.
4. Select your machine configuration. Recommended settings:
 - General-purpose
 - N1 Series
 - 4 CPUs, 15GB memory
5. Select **Boot disk** and navigate to **Change -> Custom Images**.
6. In the **Show Images** tab, look for the project name that Thales Support has provided. Select that project.
7. Select the **Image** tab and look for the latest CTS GA release and select it. For example, **cts-240-216**.
8. Choose a boot disk type, for example, **Standard** or **SSD**.
NOTE: Do not change the disk size.
9. Close the *Boot disk* page by clicking **Select**.
10. In the **Firewall** setting, select **Allow HTTPS traffic**.
NOTE: Do not use SSH keys.
11. Click **Create** to begin creating the CTS instance.
12. Connect to the instance using SSH to begin the CTS setup.

Upgrading CTS

You can upgrade with the standard upgrade ZIP file only if you are upgrading from CTS v2.3 or higher. If you are upgrading from any CTS version earlier than v2.3, refer to the release notes for specific details based on the current version you are upgrading from.

If you are upgrading from CTS version 2.2.x or earlier, you must perform a backup and restore procedure as part of your upgrade. The backup includes users and groups, permissions, and settings. You can perform this backup either through the CTS GUI or REST APIs. Refer to [“Backing Up and Restoring the CTS” on page 96](#) for instructions on this backup step, and on the subsequent restore step.

Notes for upgrading a cluster

- All nodes in the cluster must be upgraded to the same CTS version. A CTS cluster must not operate nodes using different CTS versions.
- When upgrading nodes in a cluster, all CTS operations must be suspended. If this is not possible, you must perform a rolling upgrade. The nodes to be upgraded must be taken offline -- for example taken out of the load balancer -- before an upgrade is performed. When the load balancer is ready to use the upgraded nodes, all other nodes not yet upgraded must be taken offline. At any one time, the load balancer must not serve nodes running different CTS versions.
- While a node upgrade is in progress, it must not be interrupted. If it is interrupted or does not complete successfully, you must discard the node by removing it from the cluster. Create a new VM, upgrade it to the desired version, and rejoin it to the cluster. When joining the node to the cluster, it must have the same version as all other nodes in the cluster.
- All nodes in the cluster must be upgraded and must complete the upgrade before resuming operations.

Upgrades to AWS instances cannot use SSN key pairs

Upgrading an EC2 instance from VTS v2.5 or earlier retains the use of passwords for the `cliadmin` instead of SSH key pairs. Only newly-created instances from CTS v2.6 and above can use the SSH key pair in AWS.

Upgrade steps

If you have multiple nodes in a cluster, you can upgrade all nodes at the same time.

1. Obtain the upgrade ZIP file from your Thales sales or support representative, and place the file on a server that is accessible from CTS over either HTTP or FTP.
2. Although the upgrade ZIP file is encrypted and digitally signed, the HTTP server can use SSL (HTTPS) and basic authentication. The HTTPS server can also use a self-signed certificate.

3. Log in as cliadmin to the CTS virtual machine that you want to upgrade. In the following example command, substitute your own CTS hostname.

```
$ ssh -l cliadmin <Tokenization Server hostname>
cliadmin@<Tokenization_Server_IP_address> password: <Enter the cliadmin
password>
[...]
CipherTrust Tokenization Server Main Menu
main> ?

      Command          Description
      =====
auth          Authentication Setup
cluster       Cluster Setup
icapi         SafeNet ProtectApp Setup
network       Network Setup
system        System Setup
vae           VAE Configuration
vts           Tokenization Setup
quit|q|up|<ctrl-d>  Quit or return to previous menu
exit          Exit application

      Enter <command> to display usage
main>
```

4. At the main menu, enter:

```
main> vts upgrade --url <upgrade_zip_url>
```

Another option is to navigate to the **vts** submenu and enter the upgrade command directly:

```
main> vts
vts> upgrade <upgrade_zip_url>
```

5. Upgrade the CTS software with the upgrade command. In the example command below, substitute the location of your own upgrade ZIP file.

```
vts> upgrade <Upgrade Package URL>
```

Examples:

```
upgrade https://example.com/cts-upgrade-<version>.zip
upgrade https://username:password@example.com/cts-upgrade-<version>.zip
upgrade ftp://example.com/cts-upgrade-<version>.zip
```

Watch the output of the upgrade command while the upgrade is underway. After several progress messages, you should see a success message like the following:

```
Upgrade complete.
```

Depending on the type of upgrade, the system will be rebooted at the end for changes to take effect. You are prompted at the beginning of the upgrade for confirmation.

There are other options to perform an upgrade. Run `upgrade --help` for details:



Configuring the CTS System

This chapter describes how the network administrator uses the CTS CLI to configure the CTS environment, establish connections between the various components, and other system-level tasks. It contains the following sections:

- “Accessing the CLI”
- “Basic Configuration Steps”
- “Configuring CTS Nodes”
- “Configuring an AD/LDAP Server (optional)”
- “Configuring ADFS and LDAP for OAuth”
- “Configuring the settings.ini File for LDAP”
- “Restarting the CipherTrust Tokenization Server”

Accessing the CLI

The network administrator uses the CTS CLI to configure the CTS environment.

1. Access the CLI through a terminal emulator like PuTTY.
2. On first access, enter the default login and password:
login: cliadmin
Password: cliadmin123
3. On first access, you will be prompted to change the password.
Save and store your new password securely.
4. On first access, you will be prompted to review and sign the License Agreement.

CLI navigation cheat sheet

Table 5: CLI Navigation Summary and Guidelines

Navigation	Command	Example
Starting prompt	main>	
List CLI categories or the commands within categories	?	<pre>main> ? auth Authentication setup cluster Cluster Setup icapi SafeNet ProtectApp Setup network Network Setup system System Setup vae VAE Configuration vts Tokenization Setup quit q up <ctrl-d> Quit or return to previous menu exit Exit application</pre>
Open a category and display its commands	<category name> ?	<pre>main> cluster cluster> ? add Register a new node to a cluster apiadmin API admin user setup create Create New Cluster join Join node to a cluster remotejoin Remotely join nodes to a cluster remove Remove node from cluster show Show cluster status quit q up <ctrl-d> Quit or return to previous menu exit Exit application</pre>
Get help: See command usage and example output	<enter the command without a value>	<pre>network> checkport usage: checkport [-h] [--host HOST] [--port PORT [PORT ...]] Check port connection status of a host Optional arguments: -h, --help show this help message and exit --host HOST hostname or IP address of host --port PORT [PORT ...] port numbers to scan</pre>
Have CLI complete a category name, command, or argument when typing	Press <Tab>	Enter enough characters to uniquely identify a category, command, or argument, and then press the <Tab> key. The CTS CLI will complete it for you.
Return to the main level	up	<pre>main> cluster cluster> up main></pre>

Table 5: CLI Navigation Summary and Guidelines

Navigation	Command	Example
End the CLI session	<code>exit</code>	
NOTE: You must enter a category to execute any of its commands. For example, the reboot command is in the system category, so you must first enter system, then enter reboot.		

Basic Configuration Steps

The configuration steps in the following sections employ a subset of the CLI commands. See [“CTS CLI Reference” on page 99](#) for a summary of CLI commands. For complete usage details on any CLI command, run the command with the “-h” option.



NOTE: For multi-node installations, the basic configuration steps in this section must be completed on each node before you follow [“Configuring CTS Nodes”](#) as described on page 50.

Configure the CTS network settings

Configure the network setting for VMware and bare metal installations with a static IP implementation.



IMPORTANT: Network settings are required only for VMware and bare metal installations, with a static IP implementation. Do not attempt to set the IP address and gateway of the network interface if you are using CTS in an Azure, Amazon, or Google cloud. Cloud providers assign the IP addresses to the VM through DHCP.

To configure the CTS network settings, complete the following tasks:

- “Set the IP address for the CTS VM”
- “Configure the CTS DNS server” or use “The /etc/hosts file.”

Check the annotations you made under [“CipherTrust Tokenization Server requirements” on page 19](#) for the IP address, gateway address, and DNS values you will need.

Set the IP address for the CTS VM

1. Using the CTS CLI, navigate to the *network* commands menu.
2. There are two ways to set the IP address for the CTS VM: Interactive (with the *setup* command) or Non-interactive (with the *set* command).

Interactive

To set the IP address using the interactive mode, use the *setup* command followed by the network device name:

```
network> setup eth0
```

Using the *setup* command prompts you for the IP address, as well as other information required to set up the specified network device, such as subnet mask, gateway, and so on.

Non-interactive

To set up the IP address with the non-interactive mode, use the *set* command with the *--ipaddr* IPADDR option and argument value. With the same *set* command invocation you can also configure other network and device parameters with additional options:

```
vormetric> network set --help
usage: set [-h] [--onboot {yes,no}] [--dhcp] [--ipaddr IPADDR]
          [--netmask NETMASK] [--gateway GATEWAY] [--defroute {yes,no}]
          [--mtu MTU] [-y]
          device
Configure Network Interface
[. . .]
```

Table 6: Options and Arguments to the Non-interactive *set* Command

Option	Argument	Description
--onboot	yes no	Enable this device at boot time
--dhcp		Use DHCP
--ipaddr	IPADDR	Set the IPv4 address of this device to <i>IPADDR</i>
--netmask	NETMASK	Sets the netmask of this device to <i>NETMASK</i> (for example, 255.255.0.0)
--gateway	GATEWAY	Specify the gateway IP address as <i>GATEWAY</i>

Table 6: Options and Arguments to the Non-interactive set Command

Option	Argument	Description
--defroute	yes no	Set to yes to use this device as the default route. Default value is no . NOTE: Set only one device as the default route.
--mtu	MTU	Set the maximum transmission unit (MTU) value to <i>MTU</i>
-y		Do not prompt for confirmation

3. Verify the IP address configuration using the show command followed by the device name:

```
network> show eth0
```

Configure the CTS DNS server

If your network environment uses DNS, use the dns command to configure the nameservers:

```
network> dns --add IP [IP ...]
```

You can configure a maximum of three nameservers.

Show the DNS settings with the --show option:

```
network> dns --show
DNS resolver is using DHCP
Generated by DHCP:
  nameserver 192.168.100.199
  nameserver 192.0.0.299
```



NOTE: If you are using DHCP, using the dns command overrides the nameservers provided by the DHCP.

The /etc/hosts file

If you have hostnames that do not use DNS, you can use the local /etc/hosts file to resolve IP addresses. To update this file, run the CLI "system hosts" command, which includes options to add or remove IP addresses, and to show the hosts file contents.

```
main> system hosts
usage: hosts [-h] [--show] [--add IP [HOSTNAME ...]] [--remove IP]
Update hosts file
optional arguments:
  -h, --help            show this help message and exit
  --show                Show hosts file entries
  --add IP [HOSTNAME ...] Add/update hosts entry
  --remove IP           Remove hosts entry
```

For example:

```
main> system hosts --add 192.168.99.99 rgrimes
192.168.99.99      rgrimes
10.3.110.187      cts-builder2

main> system hosts --show
192.168.99.99      rgrimes
10.3.110.187      cts-builder2
```



NOTE: You must do this on *each* CTS, since entries in the hosts file are not replicated across CipherTrust Tokenization Servers.

Setting the CTS hostname

You must set a hostname for the CTS VM.

This hostname must match the hostname created on the DSM or other Thales keystore. The CTS will not register to the keystore if these names do not match.

To set the hostname of the CTS VM, run the system hostname command with the --set option in the CTS CLI:

```
main> system hostname --set your_cts_hostname
```

Importing a server certificate for TLS

Use the `vts server_certificate` command in the CTS CLI to import a server certificate. Enter the command without any options to see a detailed help message that explains all arguments to the command (some of which are deleted from the example shown below to save space):

```
main> vts server_certificate
usage: server_certificate [-h] [--import {certificate,key}]
                        [--create {csr,selfsigned,key}] [--size SIZE]
                        [--show {key,certificate}] [-y] [--country C]
                        [--state ST] [--locality L] [--organization O]
                        [--organizational_unit OU] [--common_name
HOSTNAME]
                        [--email EMAIL] [--dns [DNS [DNS ...]]]
                        [--days DAYS]

Server Certificate Utility

optional arguments:
  -h, --help                show this help message and exit
  --import {certificate,key}
                           import a certificate or private key
  --create {csr,selfsigned,key}
                           create a certificate signing request, a self-
signed certificate, or a new private key.
                           a certificate signing request is created using
[ . . . ]
```

Certificates in PEM format

All certificates used in CTS must be in PEM format. The following is an example of a certificate in PEM format:

```
-----BEGIN CERTIFICATE-----
MIIDrTCCAxagAwIBAgIBADANBgkqhkiG9w0BAQQFADCBnDEbMBkGA1UEChMSVGhl
IFNhbXBsZSBDb21wYW55MRQwEgYDVQQLewtDQSBEdXZpc2lvb2lvcjEwMDU0MgU0
DQEJARYNY2FAC2FtcGxLLmNvbTETMBEGA1UEBXMKTWV0cm9wb2xpczERMA8GA1UE
CBMtM3V3IFlvcmsxCzAJBgNVBAYTAlVTMRQwEgYDVQDEwtUU0MgU09vdCBDQTAE
Fw0wMTEyMDgwNDI3MDVaFw0wMjE5MDgwNDI3MDVaMIGCMRswGQYDVQQKEsJUaGUg
U2FtcGxLIENvbXBhbnkxFDASBgNVBASTC0NBIEpdmLzaw9uMRwwGgYJKoZIhvcN
AQkBFg1jYUBzYW1wbGUuY29tMRMwEQYDVQQHEwpNZXRyb3BvbGlzMREwDwYDVQI
Ewh0ZXcgW9yazELMAkGA1UEBhMCVVMxFDASBgNVBAMTC1RTQyBSb290IENBMIGf
```

```
MA0GCSqGSIB3DQEBAQUAA4GNADCBiQKBgQDaiAwfKB6ZBtNTRTio6ddomt0S9ec0
NcuvtJogt0s9dXpHowh98FCDjnLtCi8du6LDTZluhl0tTFARPlV/LVnpsbyMCXMs
G2qpdjJop+XIBdvoCz2HpGXjUym8WLqt+c0WwJqUSwiEba74JG93v7TU+Xcvc00
5MwnxmKZZD/R3QIDAQABo4H8MIH5MAwGA1UdEwQFMAMBAf8wHQYDVR00BBYEFg/v
yytrBtEquMX2dreysix/MLPMMIHJBgNVHSMGcEwgb6AFG/vyytrBtEquMX2drey
six/MLPMoYGipIGfMIGcMRswGQYDVQQKEsJUAUGUgU2FtcGxLIENvbXBhbnkxZDAS
BgNVBAsTC0NBIERpdmlzaW9uMRwwGgYJKoZIhvcNAQkBFg1jYUBzYW1wbGUuY29t
MRMwEQYDVQQHEwpNZXRyb3BvbGlzMREwDwYDVQQIEWh0ZXcgWW9yazELMAkGA1UE
BhMCMVVMxZDASBgNVBAMTC1RTQyBSb290IENBggEAMA0GCSqGSIB3DQEBAUAA4GB
ABcLymJfsPOUazNQ08aIaxwVbXWS+8AFEkMMRx6068ICAMubQBvs8Buz3ALXhqYe
FS5G13pw2ZnAlSdTkSTKkE5wGZ1RYSfyiEKXb+u0KhDN9LnajDzaMPkNDU2NDXDz
SqHk9ZiE1boQaMzjNLU+KabTLpmL9uXvFA/i+gdenFHv
-----END CERTIFICATE-----
```

Option 1: Regenerating a self-signed server certificate

To generate a self-signed certificate run the `vts server_certificate` command with the `--create selfsigned` option:

```
main> vts server_certificate --create selfsigned
```

Enter answers to the informational questions and restart the web server when asked.

If your server requires a particular TLS protocol, you can set it by following the instructions in [“Specifying the TLS protocol version \(optional\)” on page 61](#).

Option 2: Importing an authenticated third-party server certificate (recommended)

To import an authenticated third-party CA certificate to the CTS, follow the steps below:

1. In the CTS CLI, go to the `vts` category, and enter `server_certificate --create csr` to generate a certificate request:

```
main> vts
vts> server_certificate --create csr
```

2. Follow the system prompts. A certificate request displays at the command line:

```
----- BEGIN CERTIFICATE REQUEST -----
. . .
----- END CERTIFICATE REQUEST -----
```

3. Copy and paste the certificate request including the "REQUEST" lines onto a third-party certificate request form, such as GoDaddy, Thawte, or Verisign.

4. After receiving the certificate, at the vts command category, run `server_certificate --import certificate` at the command line:

```
vt> server_certificate --import certificate
```

5. When prompted to continue, enter yes.
6. Note that the certificate must be in PEM format. Copy and paste the entire certificate including the CERTIFICATE lines, as shown in the example below:

```
-----BEGIN CERTIFICATE-----
MIID1TCCAr2gAwIBAgIJAMf9x1/DhPfPMA0GCSqGSIb3DQEBCwUAMIGAMQswCQYD
VQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcn5pYTERMA8GA1UEBwwIU2FuIEpvc2Ux
. . .
9hZV7uAAnRzqaodIPQYABwj58/eo8hMw4JfLfJwTavZuFt/hpnbxEECynd+s4
cLSgBd7x4aeDKt0vinSvBv2niyZyLOv1eg==
-----END CERTIFICATE-----
```

7. Enter `Ctrl-d` to end the input.
8. The CTS imports the certificate and tries to verify that the imported certificate matches the private key generated at the time the certificate was generated.
9. The system prompts you to restart the web server. Type yes.
10. The web server component of the CTS is restarted, and the new certificate takes effect. The system is set up using the third-party server certificate for TLS connections.

Registering CTS with the DSM

CTS can use as its keystore the Vormetric DSM, or either Thales KeySecure Classic or Thales CipherTrust Manager (CM) (formerly Next Generation KeySecure). For KeySecure or CM instructions, see [“Registering CTS with Thales KeySecure or Thales CM” on page 44](#).

See [“Keystore \(DSM, KeySecure Classic, or CM\) requirements” on page 19](#) for a list of the DSM prerequisites before configuring.

1. If your DSM hostname is not registered to the DNS, add the hostname to the `/etc/hosts` file. Go to the system submenu in the CTS CLI and enter: `hosts --add <dsm IP Address> <dsm hostname>`

```
system> hosts --add 192.168.34.12 dsm1
```

2. If you have not done so already, set the desired hostname of your CTS, using the hostname command under the system category menu.

```
system> hostname --set cts1
```

3. Ask the DSM Administrator to add the CTS VM hostname to DSM database as follows.

On the DSM Management Console click **Hosts > Add**. Enter:

Host Name: See [“Setting the CTS hostname”](#).

Agent type: Key

License type: Set as per your DSM licensing agreement

Communication Enabled: Checked

All other parameters can be left at the default. Refer to the DSM manual for details.



IMPORTANT: The hostname added to the DSM must match the hostname set on the CTS VM. If they do not match, registration will fail.

4. Return to the CTS CLI and register the CTS with the DSM.
 - a. Obtain the DSM hostname from the *DSM Management Console*.
 - b. Go to the vae category of the CLI and run the register command, supplying the hostname of the DSM as a parameter. For example:

```
main> vae
vae> register jw-dsm-1150.i.vormetric.com
Welcome to the CipherTrust Key Agent Registration Program
Agent Type: CipherTrust Key Agent
Agent Version: 6.2.0.53...
```

A successful registration results in the following message being displayed:

```
Successfully registered the CipherTrust Key Agent with the primary
CipherTrust Data Security Server on jw-dsm-1150.i.vormetric.com
```

5. Log into the DSM and navigate to the hostname of the CTS VM. A successful registration shows a fingerprint hash. This should match the fingerprint message in the output when the register command was run.

Registering CTS with Thales KeySecure or Thales CM

As an alternative to using a Vormetric DSM, you can register CTS with either a Thales KeySecure Classic or a Thales Next Generation KeySecure (or Thales CipherTrust Manager) server. To configure with DSM, see [“Registering CTS with the DSM” on page 43](#).

Complete the following procedure to register CTS with either a Thales KeySecure Classic or CM server. For easier reference throughout this procedure, the term “*KeySecure*” is used to refer to either Thales keystore server product, unless otherwise specified individually.

1. Create a user and set a password for it on the KeySecure server.
Besides client certificate authentication, KeySecure servers require user/password authentication for key access.
2. Optional: Create a user-group on the KeySecure server
In addition to granting key access to a user, key access can be granted to a group. Users belonging to that group have access to the group keys.
3. Create symmetric AES keys for CTS use on the KeySecure server using the properties below:
 - Key type: AES
 - Key size: 256 bits
 - Exportable: yes
 - Prevent keys from being deleted: yes
 - Key properties: Encrypt and Decrypt
 - Owner: CTS user in Step #1 above or
 - Group: group in Step #2 above
4. If not already created, create a local CA on the KeySecure server.
This will be used by CTS for client certificate authentication.
5. Create an NAE interface on the KeySecure server using the following properties:
 - Type: NAE
 - Port number: 9000
 - Mode: TLS; user must supply password; verify client certificate
 - Username Location in Certificate: CN
 - Local CA: (select)
 - Network interface: (select)
6. Set up CTS with IP(s), port, and user credentials to the KeySecure server.
Log in to the CTS CLI and run the **icapi --set** command:

```
main> icapi set --IP <keysecure_ip> --port 9000 --user <ctsuser>
```

When prompted, enter the password of the ctsuser on the KeySecure server.

If the KeySecure is a cluster of servers, supply all the IP addresses of the servers in the cluster. For example:

```
main> icapi set --IP ip1 ip2 ip3 --port 9000 --user <ctsuser>
```

Run the **icapi show --properties** command to inspect the settings made. For example:

```
main> icapi show --properties

Client_Cert_Authentication    yes
KEYSTORE_PASSWORD            *****
KEYSTORE_USER                 ctsuser
NAE_IP                        ['10.3.211.79']
NAE_Port                      9000
Protocol                      ssl
Version                       3.1
```

7. Register CTS to the KeySecure server, with either an automated registration or a manual registration.

Automated registration is available only with Next Generation KeySecure. It is currently not supported with KeySecure Classic. To register CTS to a KeySecure Classic server, skip the following sub-procedure (“To perform automated KeySecure registration”) and resume the procedure at “To perform manual KeySecure registration” on page 48 below.

- **To perform automated KeySecure registration:**

Log in to the CTS CLI and run the **icapi register** command. For example:

```
main> icapi register --host 10.3.211.79 --user <adminuser>
```

This command requires the user credentials on the KeySecure server that has admin rights to the local CA. It creates a client certificate on the CTS that is signed by the KeySecure server. This serves as the client certificate authentication whenever CTS connects to the KeySecure server.

This command prompts for the password of the KeySecure admin user, and the information to create the certificate.

For example:

```
main> icapi register --host 10.3.211.79 --user admin
password:

Running registration will delete any existing CA and client certificates.
Do you wish to continue? <y/n> [n] y
Country Name (2 letter code) [US]:
State or Province Name [California]:
Locality Name (e.g., city) [San Francisco]:
Organization Name (e.g., company):
Organizational Unit Name:
Common Name (eg, server's hostname) [cts-250-263-202]:
Created Private-Key: (2048 bit)
CA certificate updated
Client certificate updated
Registration complete
Keysecure signed certificate:
id                : f22e722f-4135-44c7-9e78-c03b1acc83da
ca                : kylo:kylo:naboo:localca:8d2889d6-45a4-4981-95e8-d1589238deb8
issuer            : /C=US/ST=MD/L=Belcamp/O=Gemalto/CN=KeySecure Root CA
notAfter          : 2030-02-09T21:05:22Z
serialNumber      : 195829222921448800841697297203339489128
sha1Fingerprint   : 4C5B96B0F0C3D6FCF06C905E49965E4BCF370824
sha256Fingerprint :
EB7E39AD001EF9B63F9DF1370569869813869B1D592F0130CF6CADB501E625F1
subject           : /C=US/ST=California/L=San Francisco/CN=cts-250-263-202
```

A successful registration will display the client certificate information and its fingerprint.
Log in to the KeySecure server and verify that the certificate information and fingerprint match.

Notes:

- If the KeySecure server has more than one Local CA, the automated registration uses the first Local CA found. If the desired Local CA is not the first on the list, use the **--caid** option to supply the ID of the desired Local CA. You can use the **show --LocalCAs** command to view the ID of all the local CAs on the KeySecure server:

```
main> icapi show --LocalCAs --user <adminuser>
```

- By default, the certificate is set to expire in 10 years. Use the **--days** option to set the desired expiration.

- The registration command can be run in non-interactive mode by supplying all parameters to the command. Run the **icapi register --help** command for details.

- **To perform manual KeySecure registration:**

Use the following procedure to register CTS to a KeySecure Classic server. You can also use this manual procedure with a Next Generation KeySecure server if you do not want to automatically register, or for some other reason cannot automatically register.

Some reasons for using manual registration include:

- Network connectivity from the CTS to KeySecure is not yet in place.
- The user performing auto-registration does not have administration credentials to the KeySecure server. The user credentials to perform registration must include permission to run REST APIs on the server, permission to download the Local CA, and permission to sign a certificate request.
- Access to the REST API (port 443) of the KeySecure server is blocked.
- An imported private key is desired instead of letting CTS generate a random key internally.
- A common client certificate is desired for several CTS servers, for example, with CTS servers in a cluster.

a. Create a certificate private key on the CTS:

```
main> icapi create --client_key
```

You may import a private key instead of letting CTS create one. Use the **icapi upload** command to import a private key.

b. Create a certificate signing request:

```
main> icapi create --csr
```

This command creates a certificate signing request using the private key you have created or imported.

c. Using the CSR you now generated, log in to the KeySecure server and locate the Local CA to sign the CSR.

The signed CSR produces a certificate in PEM format.

d. Import the signed certificate into the CTS using the **upload** command:

```
main> icapi upload --client_certificate
```

- e. Log in to the KeySecure server and download the Local CA. Import this certificate into the CTS using the **"icapi upload --CA_certificate"** command:

```
main> icapi upload --CA_certificate
```

8. Enable ICAP.

Run **"icapi enable yes"** command to enable CTS access to the KeySecure server. For example:

```
main> icapi enable yes
Restarting cts (via systemctl): [ OK ]
ICAPI properties updated
```

9. Test connectivity to the KeySecure server.

Use the **"icapi test --server_connection"** command to test server connectivity to the KeySecure server. For example:

```
main> icapi test --server_connection
Test server connection using user: cts-user1
10.3.211.79: OK
```

This command tests the connection and authentication to the KeySecure server. A successful authentication confirms correct KeySecure user credentials and a valid client certificate.

10. Test access to an AES key.

Use the **"icapi test --key"** command to test access to an AES key on the KeySecure server. For example, to test access to a KeySecure key named cts-key:

```
main> icapi test --key cts-key
cts-key: OK
```

This command tests if the saved user credentials provide proper access to the specified key. This test requires correct user and client certificate authentication to the KeySecure server.

Troubleshooting and automation tips

- Make sure the **"icapi test"** CLI commands pass before setting up CTS for tokenization use.
- Use the **"icapi show"** CLI command to inspect if the icapi agent is enabled or not, if certificates are created or uploaded, and if server settings are correct.
- All CLI commands can be run in non-interactive mode. This mode is ideal for automating setup.

Security tips

Consider the following guidelines when configuring CTS to ensure maximum security:

- When setting up the NAE interface, it is highly recommended to use the following mode for best security:
 - TLS, user must supply password, verify client cert

Although other modes are possible, anything other than the above would be less secure.

- When creating a user for CTS to use on the KeySecure server, do not use a user with admin privileges or access to any keys that CTS will not use. The Keystore user should only have 1) login access, 2) key access privileges, and 3) access to the keys that CTS will use.

Do not add the Keystore user to the "Key Users" group as this may allow the user access to keys that CTS does not use. Preferably, create a group for CTS, assign to that group all the keys that CTS will use, and then add the Keystore user to that group.

Configuring CTS Nodes

To set up a CTS system, a single VM or multiple VMs can be used to form a cluster of nodes operating as one coherent CTS system.

A cluster of nodes is useful to achieve scalability and redundancy of the system. You can set up a CTS cluster manually, or with an automated process.

It is highly recommended to use the automated cluster setup process, which is easier to implement and is much less error-prone, especially when the cluster contains more than two nodes.

A CTS system is set up by creating the first node in the cluster. At a minimum, a CTS system can run with just one node. It is still considered a cluster, but with only one node. More nodes can be subsequently added to the cluster.



NOTE: CTS supports up to ten nodes in a cluster.

Automated cluster setup

To automatically set up a cluster:

1. Log into the CLI and enable the apiadmin user on all nodes:

```
main> cluster apiadmin [--enable] [--setpasswd PASSWORD]
```

This password must be set on all nodes that will use the automated cluster feature. The password must be the same on all nodes.

2. Choose a node to start your cluster:

```
main> cluster create
```

3. On the same node, start joining other nodes to the cluster, by specifying the IP addresses of the other nodes:

```
main> cluster remotejoin NODE_IP [NODE_IP ...]
```

4. It is good practice to disable the apiadmin user after you have finished the automated cluster setup process with the --disable command:

```
main> cluster apiadmin --disable
```

5. To add a new node to an existing cluster at a later time, start by enabling the apiadmin user with the --enable command, and from a node in the cluster run the remotejoin command as before to add one or more nodes to that cluster.

Manual cluster setup

To manually create the first node and cluster, run the `create` command under the `cluster` category submenu. The command requires the following parameters:

- **IP** - IP address of this node. It is highly recommended to use the internal IP, for example, the VPC IP address of this node. The IP address is optional if creating a cluster node using existing data (with the --preserve option).
- **USERNAME** - WebUI Admin username.
- **PASSWORD** - WebUI Admin password.
- **EMAIL** - WebUI Admin email.

The `cluster create` command supports both interactive and non-interactive modes. See the `--help` option output for details on how to use:

```
main> cluster create --help
usage: create [-h] [-u USERNAME] [-p PASSWORD] [-e EMAIL] [--preserve] [-y]
              [IP]

Create new cluster with this node as first node

[. . .]
```

Once this command completes, you can log into the CTS Web UI using the credentials entered above. Load the Web UI by loading the node IP address or the external IP address of the VM:

- `https://node_IP`
- `https://external_IP`

Manually adding nodes to a cluster

The first node in a manually-created cluster (created in [“Configuring CTS Nodes” on page 50](#)) creates a CTS cluster but with only one node. To add more nodes to the cluster, use the `join` command under the `cluster` category menu.



IMPORTANT: Before running the `join` command, the `add` command must be run on the first node. This is a way of telling the first node that an additional node will be joining the cluster. Simply running the `join` command on the new node will not work. The first node must know about the new node by running the `add` command.

To add a second node to a cluster:

1. Log in to the CLI of the first node and run the `cluster add` command:

```
main> cluster add NODE2_IP
```

where `NODE2_IP` is the IP address of the new (second) node joining the cluster.

2. Log in to the CLI of the second node. Run the `add` command for the first node, and then the `join` command:

```
main> cluster add NODE1_IP
main> cluster join NODE2_IP NODE1_IP
```

The `join` command requires two parameters: `NODE2_IP` (the current, new node IP) and `NODE1_IP` (the existing cluster node IP).

The `add` command is optional and not necessary since the `join` command will take care of it. However, if you are joining a cluster already having more than one node in the cluster, you must run the `add` command for all nodes already in the cluster.

The `join` command only lets you select one `node_ip` to join for connectivity purposes, but you will need to tell the system all the other nodes in the cluster you are joining and that is accomplished by running the `add` command.

Example: Manually setting up a four-node CTS cluster

The following procedure shows how to manually set up a CTS system with four nodes, with IPs `NODE1_IP`, `NODE2_IP`, `NODE3_IP`, and `NODE4_IP`.

1. On node1, run the `create` command:

```
main> cluster create NODE1_IP
```

2. On node1, run the `add` command for all nodes:

```
main> cluster add NODE1_IP  
main> cluster add NODE2_IP  
main> cluster add NODE3_IP  
main> cluster add NODE4_IP
```

Note: The `add NODE1_IP` command above is redundant since `NODE1` is already joined to the cluster. But for clarity, this repetition shows that the system must be told of all nodes joining the cluster.

3. Repeat the same `add` commands shown in Step 2 on nodes `NODE2`, `NODE3`, and `NODE4`.
4. Log in to `NODE2` and run the `join` command:

```
main> cluster join NODE2_IP NODE1_IP
```

5. Log in to `NODE3` and run the `join` command:

```
main> cluster join NODE3_IP NODE1_IP
```


NODE3 can also be joined to the cluster using NODE2_IP as the second parameter instead of NODE1_IP. This is because NODE2 is already joined to the cluster. Any node can be used as the second parameter as long as that node is already joined to the cluster.

For example, the following is equivalent to the preceding example above:

```
main> cluster join NODE3_IP NODE2_IP
```

6. Repeat the join command for NODE4.

```
main> cluster join NODE4_IP NODE2_IP
```

Log into the GUI on all nodes and verify that they are all replicating. Changing a value in the GUI of one node should be reflected in the same changes on all other nodes.



IMPORTANT: To add a new node to a cluster, all nodes in the cluster must be online. No new nodes can be added if not all nodes in the cluster are online.

Removing a node from a cluster

The process of removing a cluster node is the same for a manually-created cluster as it is for an automatically-created one.

To remove a node from a cluster, run the `remove` command from one node -- any node except the node that will be removed. You only need to run the `remove` command on one node. All other nodes see this command and remove the node from their system.

As with adding a node to a cluster, to remove a node from the cluster, all nodes must be online so that they will know that a node is being removed.

Configuring an AD/LDAP Server (optional)

The Active Directory/LDAP server is optional. However, you can use it to import CTS users and CTS GUI Administrators. See [“Active Directory/LDAP server requirements” on page 21](#) before configuring. You must configure the AD/LDAP server on every node in your CTS HA cluster.

1. Go to the auth command category in the CTS CLI to see the ldap command.

```
main> auth
auth$ ?
```

Command	Description
=====	=====
ldap	LDAP Setup
oauth2	OAuth2 Setup

2. Like all CLI commands, you can type the command without a parameter and a usage example displays:

```
auth> ldap
usage: ldap [-h] [--show] [--setup] [--enable {true,false}]
           [--server SERVER_URI] [--bind_dn BIND_DN]
           [--bind_password PASSWORD] [--user_prefix PREFIX]
           [--user_search_filter FILTER]
           [--user_search_scope SCOPE] [--group_search_scope SCOPE]
           [--active_user_group GROUP] [--staff_user_group GROUP]
           [--super_user_group GROUP]
           [--ca_cert {import,remove,show}] [--test {bind_dn}]
           [--query_user NAME] [-y]

LDAP Setup

optional arguments:
-h, --help            show this help message and exit
--show                show LDAP settings
[ . . . ]
```

3. Disable, then enable LDAP support in the Tokenization Server:

```
auth> ldap --enable false
LDAP settings have been saved
auth> ldap --enable true
Restarting cts (via systemctl):          [ OK ]
LDAP settings have been saved
auth>
```



NOTE: If you disable LDAP support, then you can use only the CTS GUI to create users and groups.

4. Set the AD/LDAP server URI with the --server command (where the default port number for an LPDAS server is 636):

```
auth> ldap --server ldaps://<hostname_or_IP_address>:<Port>
```

or (where the default port number for an LDAP server is 389):

```
auth> ldap --server ldap://<hostname_or_IP_address>:<Port>
```

Example:

```
auth> ldap --server ldap://192.168.118.99:389
LDAP settings have been saved
auth>
```

5. Set the binding DN of the user who has access to the AD/LDAP server with password. Example:

```
ldap --bind_dn <FQDN of user with access to LDAP Server>
```

Example:

```
auth> ldap --bind_dn cts0@cts.vormetric.com
Enter Password :
Enter Password again :
Set LDAP Bind User DN SUCCESS
auth>
```

6. Set user search scope using user_search_scope command:

```
ldap --user_search_scope <User search scope>
```

Example:

```
auth> ldap --user_scope dc=cts,dc=vormetric,dc=com
LDAP settings have been saved
auth>
```

7. Set group search scope using the group_search_scope command:

```
ldap --group_search_scope <Group search scope>
```

Example:

```
auth> ldap --group_search_scope CN=Users, dc=cts,dc=vormetric,dc=com
LDAP settings have been saved
auth>
```

8. Set attribute field for user search filter using the user_search_filter command:

ldap --user_search_filter <Name of user field in AD/LDAP>

Example:

```
auth> ldap --user_search_filter sAMAccountName
LDAP settings have been saved
auth>
```

9. Set user prefix name using `user_prefix` command. The user prefix is the attribute that should be used to prefix the user name when searching for the user name in the LDAP database.

ldap --user_prefix <User prefix name. Example: uid or cn or sn>

Example:

```
auth$ user_prefix uid
Set LDAP USER PREFIX SUCCESS
```

10. Set the Tokenization Server active user group using the `active_user_group` command:

```
auth> ldap --active_user_group <Group whose members can access the
RESTful API>
```

In this example, any user who wants access to the CTS RESTful API must belong to the AD group `ctsUsers` within `Users` of the domain `cts.vormetric.com`:

```
auth> ldap --active_user_group cn=ctsUsers,cn=Users,dc=cts,dc=vormetric,dc=com
LDAP settings have been saved
auth>
```

11. Set the CTS superuser group using the `--super_user_group` command:

ldap --super_user_group <Group that can access the Tokenization Server GUI & APIs>

In this example, any user who requires access to the CTS GUI and REST APIs must belong to the AD group `ctsSuperuser` within `Users` of domain `cts.vormetric.com`:

```
auth> ldap --super_user_group cn=ctsSuperuser,cn=Users,dc=cts,dc=vormetric,dc=com
LDAP settings have been saved
auth>
```

12. *For LDAPS support only:* Import the CA certificate chain of the LDAP server to the built-in LDAPS client of the CTS. A Certificate Management tool like DigiCert or Certificate Manager tool is recommended to request the certificate, after which use the `--ca_cert` option to import the certificate into the CTS.



NOTE: The `--ca_cert` option uses an underscore (`_`) instead of a hyphen (`-`).

- a. Using a Certificate Management tool like DigiCert or Certificate Manager tool, download your intermediate and root certificates.



CTS requires all certificates to be in PEM format.

- b. Run `--ca_cert import` to import the certificate trust chain. Paste the entire body of each certificate in the order shown in the example, then terminate with `Ctrl-d` when finished.

Example:

```
auth> ldap --ca_cert import
Cut-and-paste below a CA Certificate in pem format. Press Ctrl-d when
done.
-----BEGIN CERTIFICATE-----
(Your Intermediate certificate: DigiCertCA.crt)
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
(Your Root certificate: TrustedRoot.crt)
-----END CERTIFICATE-----
Restarting cts (via systemctl):                [ OK ]
LDAP CA Certificate has been uploaded.
auth>
```

13. Execute the `--show` option to list the LDAP configuration:

```
auth> ldap --show
ENABLE                true
SERVER_URI            ldap://192.168.118.99:389
BIND_DN               cts0@cts.vormetric.com
BIND_PASSWORD         *****
USER_SEARCH_SCOPE     dc=cts,dc=vormetric,dc=com
GROUP_SEARCH_SCOPE    CN=Users,dc=cts,dc=vormetric,dc=com
USER_SEARCH_FILTER     SAMAccountName
ACTIVE_USER_GROUP     cn=ctsUsers,cn=Users,dc=cts,dc=vormetric,dc=com
SUPER_USER_GROUP
auth>
```

Enabling the Client Authentication Feature (optional)

See also: [“Client certificates and user authentication” on page 6.](#)

In addition to Basic Auth (username and password), the CTS server can authenticate clients using client certificates.



NOTE: Client certificates must be in PEM format.

To enable, client-certificate authentication must be enabled on every node of the CTS cluster. Perform the procedure below on each node.

Obtaining a client CA key and certificate

1. Log in to the CTS CLI.
2. Go to the vts menu and run: `client_certificate --upload`



NOTE: The `client_certificate` command uses an underscore (_) instead of a hyphen (-).

3. At the prompt, paste the text of the certificate. For example:

```
main> cts
vts> client_certificate --upload
Copy-and-paste a CA Certificate below in pem format. Press Ctrl-d when
done.
```

-----BEGIN CERTIFICATE-----

```
MIIDrTCCAxagAwIBAgIBADANBgkqhkiG9w0BAQQFADCBnDEbMBkGA1UEChMVSVGHl
IFNhbXBsZSBDb21wYW55MRQwEgYDVQQLEwtDQSBEdXZpc2lvdjEcmBo6CSqGSib3
DQEJARYNY2Fac2FtcGxllMnVbTETMBEGA1UEBxMKTWV0cm9wb2xpczERMA8GA1UE
CBMITmV3IFlvcmsxCzAJBgNVBAYTAlVTMRQwEgYDVQQDEwtUU0MgUm9vdCBDQTAe
Fw0wMTEyMDgwNDI3MDVaFw0wMjE2MDgwNDI3MDVaMIGcMRswGQYDVQQKExJUaGUg
U2FtcGxliENvbXBhbnkxFDASBgNVBAsTC0NBIERpdmlzaW9uMRwwGgYJKoZIhvcN
AQkBFg1jYUBzYW1wbGUuY29tMRMwEQYDVQQHEwpnZXRYb3BvbGZlMREwDwYDVQI
EwhOZXcgW9yazELMAkGA1UEBhMCVVMxZDASBgNVBAMTC1RTQyBSb290IENBMIGf
MA0GCSqGSib3DQEBAAUAA4GNADCBiQKBgQDaiAwfKB6ZBtNTRTio6ddomt0S9ec0
NcuvtJogt0s9dXpHowh98FCDjnLtCi8du6LDTZluhlotTFARPlV/LVnpsbyMCXMs
G2qpdjJop+XIBdvoCz2HpGXjUym8WLqt+coWwJqUSwiEba74JG93v7TU+Xcvc00
5MwnxmKZzD/R3QIDAQABo4H8MIH5MAwGA1UdEwQFMAMBAf8wHQYDVR00BBYEFg/v
yytrBtEquMX2dreysix/MLPMIHIJBgNVHSMGcgEwgb6AFG/vyytrBtEquMX2drey
six/MLPMoYGipIGfMIGcMRswGQYDVQQKExJUaGUgU2FtcGxliENvbXBhbnkxFDAS
BgNVBAsTC0NBIERpdmlzaW9uMRwwGgYJKoZIhvcNAQkBFg1jYUBzYW1wbGUuY29t
MRMwEQYDVQQHEwpnZXRYb3BvbGZlMREwDwYDVQIENwOZXcgW9yazELMAkGA1UE
BhMCVVMxZDASBgNVBAMTC1RTQyBSb290IENBggEAMA0GCSqGSib3DQEBBAUAA4GB
AbClymJfsPOUazNQ08aIaxwVbXWS+8AFEkMMRx6068ICAMubQBvs8Buz3ALXhqYe
```

```
FS5G13pW2ZnAlSdTkSTKkE5wGZ1RYSfyiEKXb+uOKhDN9LnajDzaMPkNDU2NDXDz  
SqHk9ZiE1boQaMzjNLU+KabTLpmL9uXvFA/i+gdenFHv
```

```
-----END CERTIFICATE-----
```

4. Verify the uploaded certificate by using the `client_certificate --show` command.

```
vts> client_certificate --show
```

Enabling client cert authentication with or without identities

Client certificate authentication has two modes:

- Enable without identities
- Enable with identities



NOTE: Thales strongly recommends that you use client certificates **with identities**. If you use without identities, you must pass in a username/password along with the certificate, which can be a security vulnerability that client certificates are intended to prevent. The option to use certificates without identities is provided to maintain backward compatability with the earlier basic authentication flow.

To enable without identities, run:

```
vts> client_certificate --enable-without-id
```

To enable with identities (recommended), run:

```
vts> client_certificate --enable-with-id
```

Note: The CTS GUI administrator must enter the common name (CN) from the certificate as a user name. See [“Managing CTS Users” on page 73](#).

To disable client certificate authentication, run:

```
vts> client_certificate --disable
```

To check if client certificate authentication is enabled or not, run:

```
vts> client_certificate --status
```

Sample use in code

To use client-certificate authentication, include the client key and CA certificate when running the REST API.

Example without identity:

```
$ curl --tlsv1.2 -k --key client.key --cert client.crt \  
-X POST -u username:password \  
-d '{"tokengroup" : "t1" , "data" : "9453677629008564",  
"tokentemplate" : "Credit Card" }' \  

```

```
https://192.168.12.88/vts/rest/v2.0/tokenize  
Example with identity (recommended):  
$ curl --tlsv1.2 -k --key client.key --cert client.crt \  
-X POST \  
-d '{"tokengroup" : "t1" , "data" : "9453677629008564",  
"tokentemplate" : "Credit Card" }' \  
https://192.168.12.88/vts/rest/v2.0/tokenize
```

Specifying the TLS protocol version (optional)

Clients may require different versions of TLS protocol. You may specify what version of SSL/TLS is supported by CTS. If you opt in to this feature, you must specify the TLS version on every node in your CTS HA cluster.

To do this, log in to the CTS CLI and run:

```
vt> ssl_protocols --set [ TLSv1.2 | TLSv1.1 | TLSv1 v
```

where you specify the lowest protocol version allowed. TLS versions above the specified version are also allowed:

TLSv1 supports tlsv1, tlsv1.1, and tlsv1.2

TLSv1.1 supports tlsv1.1, and tlsv1.2

TLSv1.2 supports tlsv1.2



NOTE: The `ssl_protocols` command uses an underscore (_) instead of a hyphen (-).

Example:

```
main> vts  
vt> ssl_protocols --set TLSv1.1  
vt> ssl_protocols --status  
SSL Protocols: TLSv1.1 TLSv1.2  
vt>
```

Redirecting log messages to a remote log server (optional)

Optionally, you can redirect log messages to a remote logging server. If you opt in to this feature, it is recommended that you redirect log messages off every node in your CTS HA cluster to the remote log server. You can also add SIEM software (example: Splunk, QRadar) for logging. See the *VDS Security Intelligence User Guide*.

1. Go to the vts category in the CTS CLI and use the remotelog command:

```
main> vts
vts> remotelog
usage: remotelog [h] [--show] [--set HOST:PORT] [--enable] [--disable]
        [--check]
Set Remote Logging
[...]
```

2. Show the current remote log settings:

```
vts> remotelog --show
remote-host:514, disabled
vts>
```

3. To enable or disable the remote logging re-direct, use the --enable or --disable options, respectively:

```
vts> remotelog --enable
remote-host:514, enabled
vts>
```

4. To set the remote log server hostname and port, use the --set option with the *HOST:PORT* argument:

```
vts> remotelog --set new-remote-host:1099
new-remote-host:1099, enabled
vts>
```

You do not need to restart the CTS after changing the remote log server configuration.



NOTE: The current log level is set to 'INFO' level for operations and 'DEBUG' level for other packages. Use the `loglevel` command to show or set the log level.

Configuring OAuth

You need to save the following attributes in the `settings.ini` relative to the OAuth2 configuration:

- Method (basic/oauth)
- Redirect URI
- Provider name

These settings will enable the GUI to show the correct login page (Redirect button or basic username/password form).

Use the CLI to configure OAuth2. Log into the CLI and run the **oauth2** command under the auth menu.

```
main> auth oauth2
```

Use the **--name** option to set the provider name (or to disable entirely), and the **--uri** option to set the URI to the OAuth2 client. Use the **--import** option to import the public key or certificate (in PEM format).

Configuring ADFS and LDAP for OAuth

Follow these procedures to configure Active Directory Federated Services (ADFS) and LDAP for an OAuth authentication option.

To export a certificate for CTS use:

1. Open Server Manager.
2. Click **Tools** and select **AD FS Management**
3. Click **Certificates**.
4. Double-click on **Token-signing certificate**.
5. In the *certificate* window select **Details** tab.
6. Click **Copy to File...**
7. Click **Next**.
8. Select **Base64 format** and **Next**.
9. Enter file name.

To add an application group:

1. Open Server Manager.
2. Click **Tools** and select **AD FS Management**.
3. Right click **Application Groups** and select **Add Application Group** wizard.
4. Enter name and select **Native application accessing a web API**.
5. Copy **Client Identifier** to a template for later use.
6. Enter your CTS login URL in **Redirect URI** field (`https://<cts uri>/login`) and click **Add**.

7. Click **Next**.
8. Enter the same CTS URI as entered in previous step in the **Identifier** field and click **Add**.
9. Click **Next**.
10. Use default (**Permit everyone**) and click **Next**.
11. Click **Next** on the review page.
12. Click **Close**.

To add properties to a new group:

1. Double-click on the desired new group.
2. Select **<new group> - Web API** and click **Edit**.
3. Select **Issurance Transform Rules** and click **Add Rules**.
4. Select **Claim rule template: Send LDAP Attributes as Claims**.
5. Click **Next**.
6. Enter a rule name.
7. Add the following rules:
 - Token-Groups - Unqualified Names : Group
 - User-Principal-Name : UPN
 - SAM-Account-Name : Name
 - Surname : Surname
 - Given-Name : Given Name
8. Click **Finish**.
9. Click **OK**.

To test ADFS authentication:

```
curl -k -X POST https://<adfs server>/adfs/oauth2/token -d  
'grant_type=password&username=<user>&password=<password>&client_id=<client id>5&resource=https%3A%2F%2Fmickeymouse'
```

The expected response is:

```
{"access_token":"eyJ0eXA... mVSpzmW-FBDlg"}
```

Configuring the settings.ini File for LDAP

Set up the settings.ini file according to which LDAP scenario you implement:

- [“AD email” on page 65](#)
- [“AD DN” on page 66](#)
- [“Linux LDAP DN” on page 66](#)

To test your settings, use the CLI option **--query_user** to the **ldap** command:

```
auth> ldap --query_user NAME
```

AD email

The following is an example settings.ini configuration for an LDAP setup with AD email:

```
auth> ldap --show
SERVER_URI          ldap://10.3.118.98:389
BIND_DN             cts0@cts.vormetric.com
BIND_PASSWORD       *****
USER_PREFIX
USER_SEARCH_FILTER  sAMAccountName
USER_SEARCH_SCOPE    CN=Users,DC=cts,DC=vormetric,DC=com
GROUP_SEARCH_SCOPE  CN=Users,DC=cts,DC=vormetric,DC=com
ACTIVE_USER_GROUP    CN=ctsUsers,CN=Users,DC=cts,DC=vormetric,DC=com
STAFF_USER_GROUP     CN=ctsManager,CN=Users,DC=cts,DC=vormetric,DC=com
SUPER_USER_GROUP     CN=ctsAdmin,CN=Users,DC=cts,DC=vormetric,DC=com
ENABLE              true
```

AD DN

The following is an example settings.ini configuration for an LDAP setup with AD DN:

```
auth> ldap --show
SERVER_URI          ldap://10.3.118.98:389
BIND_DN             CN=cts User0,CN=Users,DC=cts,DC=vormetric,DC=com
BIND_PASSWORD       *****
USER_PREFIX         CN
USER_SEARCH_FILTER   cn
USER_SEARCH_SCOPE    CN=Users,DC=cts,DC=vormetric,DC=com
GROUP_SEARCH_SCOPE   CN=Users,DC=cts,DC=vormetric,DC=com
ACTIVE_USER_GROUP    CN=ctsUsers,CN=Users,DC=cts,DC=vormetric,DC=com
STAFF_USER_GROUP     CN=ctsManager,CN=Users,DC=cts,DC=vormetric,DC=com
SUPER_USER_GROUP     CN=ctsAdmin,CN=Users,DC=cts,DC=vormetric,DC=com
ENABLE              true
```

Linux LDAP DN

The following is an example settings.ini configuration for setup with LDAP DN on Linux:

```
auth> ldap --show
SERVER_URI          ldap://10.10.62.132:389
BIND_DN             uid=tuser1,ou=People,dc=i,dc=vormetric,dc=com
BIND_PASSWORD       *****
USER_PREFIX         uid
USER_SEARCH_FILTER   uid
USER_SEARCH_SCOPE    ou=People,dc=i,dc=vormetric,dc=com
GROUP_SEARCH_SCOPE    ou=Groups,dc=i,dc=vormetric,dc=com
ACTIVE_USER_GROUP    cn=ctsUsers,ou=Groups,dc=i,dc=vormetric,dc=com
STAFF_USER_GROUP     cn=ctsManager,ou=Groups,dc=i,dc=vormetric,dc=com
SUPER_USER_GROUP     cn=ctsAdmin,ou=Groups,dc=i,dc=vormetric,dc=com
ENABLE              true
```

Restarting the CipherTrust Tokenization Server

After all CLI configuration is completed:

1. Navigate to the vts commands menu:

```
main> vts  
vts>
```

2. Restart the web server component of the CTS:

```
vts> service --restart  
Restarting cts (via systemctl) [OK]  
vts>
```



CTS Administration in the GUI

This chapter contains the following sections:

- [“Logging into the CTS GUI” on page 69](#)
- [“Creating and Managing Passwords” on page 70](#)
- [“Managing CTS User Groups” on page 71](#)
- [“Managing CTS Users” on page 73](#)
- [“Specifying DSM Keys” on page 77](#)
- [“Tokenization Setup” on page 79](#)
- [“Applying Permissions to Users and User Groups” on page 88](#)
- [“Creating and Importing Keys” on page 90](#)
- [“Dashboard API Monitoring” on page 91](#)
- [“Administrator History Logs” on page 92](#)
- [“Backing Up and Restoring the CTS” on page 96](#)
- [“Key Rotation” on page 98](#)
- [“Monitoring via SNMP” on page 98](#)

Logging into the CTS GUI

Use the CTS administration GUI to

- create tokenization users and groups
- specify permissions for users and groups
- create token templates
- create token groups

To log into the CTS administration GUI:

1. Point your browser to the Tokenization Server:

```
https://<CTS name or IP address>/
```

2. Log in using the Admin user name and password you defined when you created the CTS cluster.
3. If you have implemented OAuth authentication, log in with those credentials.
4. The CTS GUI Home page is displayed.

Troubleshooting

A Tokenization Server Administration GUI login failure is indicated by the following error message:

```
nginx 504 Gateway Time-out error
```

In this scenario, consider the following problem/solution option:

- Problem: The Tokenization Server is not connected to the LDAP server.
- Solution: Restart the CTS service. Open the CTS CLI, and run the following command:

```
main> vts service --restart
```

Creating and Managing Passwords

The CTS uses a dynamic password policy control mechanism to ensure system security.

The CTS leverages a password checker library. If you try to create a password that does not pass the proactive password checking provided by this library, you are prompted to try another string. The policy can change dynamically based on several factors. The dynamic nature of the checking algorithm provides the most secure possible password policy options. When creating a system password, use the following guidelines:

- Use at least 1 digit, 1 uppercase letter, 1 lowercase letter, 1 special character, and use at least 8 characters in total.
- Do not use your name, the company name, or other familiar names in the password string.
- Do not use your birthday or any other personally or professionally meaningful dates in the password string.
- Include a combination of the following character types in the password string:

- Lowercase letters
- Uppercase letters
- Numbers
- Special characters

Managing CTS User Groups

User groups enable you to assign the same permissions to multiple users.

Review the following sections to familiarize yourself with user groups in CTS:

- [“CTS roles” on page 7](#)
- [“Understanding CTS user groups and permissions” on page 10](#)

If using AD/LDAP users, create at least three user groups, to correspond to the CTS roles:

- Superuser
- Staff
- Active

User groups must be defined on the LDAP system. The user group names must be entered into the CTS GUI before importing users.

Creating user groups

To create a user group:

1. Log into the CTS GUI and select **User Groups** from the navigation bar.
The All User Groups list is displayed.
2. Select **New User Group** and enter a name on the resulting page. Click **Create**.
The new group name appears in the **All User Groups** list, and is selectable from user pages and the *Permissions* page.

Assigning masks

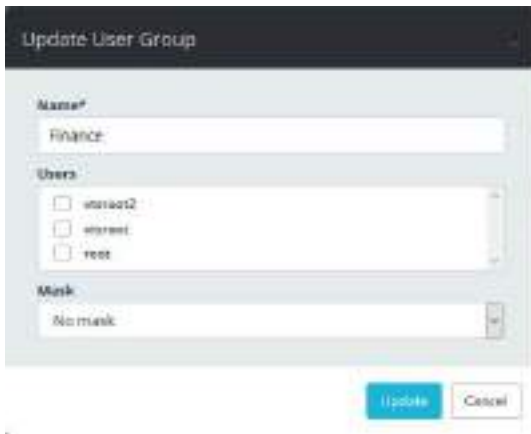
You can assign a mask to a group when initially creating the user group (at the *New User Group* dialog), or any time later to an existing user group, by clicking the user group and adding a mask assignment at the *Update User Group* dialog.

Adding users to user groups

To add a user to an existing user group:

1. From the CTS GUI select **User Groups**.
2. Click on a group from the list.
3. Select users to add to the group, and click **Update**.

Figure 7: Update User Group Dialog



Deleting a user group

To delete a user group:

1. Log into the CTS GUI and select **User Groups** from the navigation bar.
2. Select the check box by a group
3. Click **Delete**

A dialog asks you to confirm the **Delete**.



IMPORTANT: If you delete a user group to which users and/or permissions have been associated, the user group is removed from any user belonging to it, and will delete any permissions assigned to it.

Managing CTS Users

Review the following sections to familiarize yourself with users in CTS:

- “What is a CTS “user?”” on page 4
- “CTS authentication basics” on page 5
- “CTS authorization basics” on page 7

The first step is to create or import users, as described in:

- “Creating a user manually”
- “Importing users from an AD/LDAP server”



NOTE: Thales recommends creating at least one user in the GUI before attempting the task manually.

Creating a user manually

The following subsections describe how to create a user manually:

1. “Creating basic user information” on page 73
2. “Updating user information” on page 74

Creating basic user information

1. Log into the CTS GUI and select **Users** from the navigation bar.
The list of All Users is displayed.
2. Select **New User**.
In the *New User* dialog, fill in the required information and click **Create**.



IMPORTANT: The User Name must contain only English letters, numbers, or the following special characters: “@” (at-sign), “.” (period), “+” (plus-sign), “-” (hyphen), or “_” (underscore).



NOTE: User names and passwords are case-sensitive. Application code that calls the CTS RESTful APIs must pass user names and passwords in the same case that are defined in the CTS GUI.

Updating user information

After you complete the steps described in [“Creating basic user information”](#), you can add more attributes to the basic user, including changing the CTS role and assigning to a user group.

1. Log into the CTS GUI, and select **Users** from the navigation bar.
2. Click on the User Name of interest.

3. Change CTS role, User Groups assignment, Mask, or Email, as needed.
4. Click **Change password here** to update the user's password.
5. Click **Update**.

Deleting a user manually

If you delete a user to which permissions have been associated, all permissions linked to the user are also deleted.



NOTE: It is generally recommended to uncheck the “Active” status for a user rather than completely delete the user.

1. Log into the CTS GUI, and select **Users** from the navigation bar ([Figure 2](#)).
2. Select the check box by a user, click **Delete**, and confirm Delete.

Importing users from an AD/LDAP server

Review the following sections to familiarize yourself with users in CTS:

- [“What is a CTS “user?”” on page 4](#)
- [“CTS authentication basics” on page 5](#)
- [“CTS authorization basics” on page 7](#)

AD/LDAP user import prerequisites

Before you import users, complete the following tasks:

- Have your AD/LDAP connection to CTS set up and configured using the CTS CLI, as described in [“Configuring an AD/LDAP Server \(optional\)” on page 54](#).
- Create the appropriate user groups in the CTS GUI, as described in [“Creating user groups” on page 71](#).
- Set up the AD/LDAP server to import and authenticate users and user groups.

To import CipherTrust Tokenization users from an AD/LDAP server:

1. Create three AD/LDAP server groups, one for REST API users (Tokenization/Crypto/Key Management) and the other two for CTS administrators who require GUI access. The group that has been designated as the “Superuser” group in the CLI will have read/write access in the GUI, whereas the group that has been designated as the “Staff” group in the CLI will have read-only access in the GUI.
2. Populate each group with the desired REST API (“active”) users and CTS administrators.



NOTE: The name with permission to access the REST APIs must match the name specified with the `active_user` command under the `auth` category of the CTS CLI. See [“Configuring an AD/LDAP Server \(optional\)” on page 54](#).

The name of the CTS users with read-only access to the GUI must match what was specified in the `staff_user` command of the `auth` category of the CLI.

The name of the CTS administrators’ group must match the name of the group that was specified in the `super_user` command of the `auth` category of the CTS CLI.

Example AD\LDAP server group names:

- `ctsUsers` - Users in this group can create and/or access data through the RESTful API.
- `ctsStaff` - Users in this group can access the CTS GUI (read only).
- `ctsAdmin` - Users in this group can access the CTS GUI (read/write)



NOTE: `staff_user` is valid only if the user also has `active_user` permission. `super_user` is valid only if the user has both `active_user` and `staff_user` permissions. Use corporate policy to define whether or not `staff_user` and `super_user` should actually be granted REST API permissions, and disallow by policy when necessary to maintain corporate standards or separation of duties.

3. Create three group permissions in the CTS GUI with the same names as the LDAP groups (example: `ctsUsers`, `ctsStaff`, `ctsAdmin`). Members of these AD/LDAP groups inherit these permissions.

An application program that uses Basic Auth (username/password) will try to authenticate to LDAP using these credentials. An application program that uses a client certificate for authentication will use the Bind user's username and password for looking up the client certificate's common name (CN) in LDAP. If the credentials are authorized, the LDAP user is imported into the user table of the CTS. Thus, there is no need to manually create the LDAP users in the GUI. Once the user has been added to the CTS, permissions can be modified with another group permission or individually.



NOTE: User names and passwords are case-sensitive. Application code that calls the REST APIs must pass user names and passwords in the same case that are defined in the AD/LDAP server.

Unlocking user accounts

As a security feature, users are locked out after six failed login attempts.

If a user fails to successfully login after six attempts, the account is locked for 15 minutes (to prevent brute force attacks). The maximum number of attempts and cool-off time duration can be modified in the CLI.

To unlock a user account:

1. Click **Logs** in the navigation menu.
2. Select the **Locked Accounts** tab.
The locked accounts are displayed
3. Click **Unlock** for the desired account(s).



NOTE: All successful and failed login attempts are displayed under the **Access Logs** tab in the **Logs** menu. Select **Success** or **Failed** from the pull-down menu to filter the list so that only the selected access type is shown

Specifying DSM Keys

Review “[Understanding CTS Keys](#)” to understand the definition and usage of symmetric keys, asymmetric keys, and opaque objects.

Creating a key name

Use the CTS GUI to enter a key name that corresponds to a key that has been created in the DSM and is to be used for CTS functions.

After the name has been entered into the CTS local database, the key is available to associate with user and user group permissions. Symmetric keys are also be available to associate with tokenization groups.



NOTE: The very first symmetric key entered into CTS needs to be entered from the GUI of the first cluster node. Please enter this key as soon as possible. It will be used to create large lookup tables for the “RANDOM” tokenization mode.

1. After “[Logging into the CTS GUI](#)”, select **Keys** from the navigation bar. The Keys List is displayed, with the **Symmetric Keys**, **Asymmetric Keys**, and **Opaque Objects** tabs in the top navigation.

Figure 8: Keys List in the Symmetric Keys Tab

2. Select the tab for the type of key you want (for example, **Symmetric Keys**) and click the **New Symmetric Key** button.



IMPORTANT: In CTS, the very first key generated cannot be renamed, edited, or destroyed. There is a warning when creating the key. Once this initial setup is complete, you create subsequent keys with the normal *New Symmetric Key* UI, for application FPE, and the like.

3. Enter the key name that was used in the DSM and click **Create**.
The key name is displayed on the Keys List and will show as available in the *Permissions* matrix and *Tokenization Group* pages.

Deleting key names

Deleting a key name from the CTS local database does not delete the key material (nor the key name) from the DSM. Deleting a key name deletes all tokenization groups, tokenization templates, and permissions associated to it.

1. Select **Keys** from the navigation bar.
The Keys List is displayed, with the **Symmetric Key**, **Asymmetric Keys**, and **Opaque Objects** tabs in the top navigation (Figure 8).
2. Select the tab for the type of key you want (for example, **Symmetric Keys**), click the check box by the desired key, and click **Delete**.
3. Confirm and click **Delete** again.



NOTE: The very first symmetric key entered into the CTS can serve as a regular key for all encryption, signing and tokenization operations, but it is also used to build internal cryptographic tables and hence must not (and cannot) be deleted.

Tokenization Setup

Review [“Tokenization Groups, Templates, and Masks” on page 13](#) for the concepts behind setting up tokenization in the CTS GUI.



IMPORTANT: Before setting up tokenization make sure you have specified the DSM keys

Although the GUI does not constrain you to one particular implementation path, one logical sequence is:

- [“Specifying DSM Keys”](#) for tokenization (must be symmetric keys).
- [“Creating tokenization groups”](#) and associate it with the appropriate key(s).
- [“Creating custom character sets”](#) as needed (optional).
- [“Creating a data mask”](#) for detokenization.
- [“Creating tokenization templates”](#) and associate the applicable tokenization group and character set.
- **Create users and user groups** as needed.
(See [“Managing CTS User Groups” on page 71](#) and [“Managing CTS Users” on page 73](#).)
- **Grant tokenization privileges to users and user groups** by going to their permissions page, selecting the symmetric key associated with the appropriate tokenization group, and checking the **Tokenize** box. (See [“Assigning key-based permissions” on page 88](#).)
- **Grant detokenization privileges to users and user groups** in the same way, assigning a **data mask** from the drop-down menu.



NOTE: See [“Other \(non-key dependent\) permissions” on page 9](#) for an explanation of the Tokenize/Detokenize check boxes listed under “Other Permissions” in the Permissions matrix. This feature is included primarily for legacy and upgrade purposes.

Managing tokenization groups

A *tokenization group* is a way to compartmentalize a CTS server into distinct data spaces, each of which is protected by an encryption key.

A tokenization group is used as a parameter in the Tokenization API calls, `tokenize` and `detokenize`. It allows application programmers to create islands of sensitive data that are available only to members of a particular tokenization group. To use tokenization, you must create at least one tokenization group.

Use case example

A credit card company accepts business from three retail outlets called *Store1*, *Store2*, and *Store3*. The three retail outlets require tokenization of the customer credit card numbers used at their store. You could enable this by creating three tokenization groups called *Store1*, *Store2*, and *Store3* (the tokengroup parameter is case-sensitive). The application code processing sensitive data must specify one of these tokengroup parameters in each tokenize and detokenize calls (see [“Tokenization REST API” on page 133](#)).

For example:

```
curl --tlsv1.2 -X POST -u EdYee:EdYee_password -d '{"tokengroup" :
"store1" , "data" : "9453677629008564", "tokentemplate" : "Credit
Card" }' https://192.168.12.88/vts/rest/v2.0/tokenize
```

To enable tokenization services for each group, an implementation team would:

1. Decide how many tokenization groups are required and what their precise names will be (Application programmer and CTS GUI Administrator)
2. Create tokenization groups that match the names to be used in the application code: *Store1*, *Store2*, and *Store3*. (CTS GUI Administrator)
3. Go to the DSM and:
4. Create tokenization group encryption keys.
 - Use key *Template*: **Default_SQL_Symmetric_Key_Template** for those keys
 - Set the *Key Type* to **Cached on Host**.
IMPORTANT: On a DSM (version 6.1 or later), make sure to choose the CBC Encryption Mode and not CBC_CS1.
 Be sure the administrator knows that the first key must be created on the first node installed, not on secondary or subsequent cluster nodes. **Do not delete this key.** (DSM Administrator)
5. Specify the appropriate tokengroup parameter in the tokenize and detokenize API calls. Using our example credit card company, the application code would specify *Store1*, *Store2*, or *Store3* for a REST call. (Application programmer)

Creating tokenization groups

Have at least one symmetric key created in the DSM and entered in the CTS GUI, as described in [“Specifying DSM Keys” on page 77](#).

1. Select **Tokenization > Tokenization Groups**.
The All Tokenization Groups list is displayed.



2. Click the **New Tokenization Group** button.
The New Tokenization Group creation page is displayed.

The screenshot shows a 'New Tokenization Group' dialog box. It has two input fields: 'Name*' with the value 'name' and 'Key*' with a dropdown menu showing 'Sym_key1'. At the bottom right, there are 'Create' and 'Cancel' buttons.

3. Enter a Tokenization Group **Name**.



NOTE: Tokenization group names are case-sensitive. Application code must pass tokenization group names in the same case that are defined in the CTS GUI.

4. Select a **Symmetric Key** from the drop-down list.



Warning! Do not delete keys used for tokenization. If you delete a key, you will have to recreate the corresponding tokenization group to regain access to your tokens. Do not use auto-key-rotation features with tokenization.

5. Click **Create**.

Deleting tokenization groups

Before deleting a tokenization group, make sure that tokens created with the symmetric key associated with this tokenization group are not stored anywhere within (or outside) of the organization.



Warning! Once the tokenization group used to create these tokens has been deleted, they can no longer be detokenized.

To delete a tokenization group:

1. Log into the CTS administration interface.
2. Select **Tokenization > Tokenization Groups**.
3. Select the group and click **Delete**.

Managing character sets

CTS now includes enhanced support for tokenizing international characters. In earlier versions, CTS supported three default character sets:

- All digits
- Alphanumeric
- All printable ASCII (not available for RANDOM Mode)

For FPE (FF3) and FF1 modes, you can also define a custom character set, which allows you to tokenize and detokenize UTF strings.

Creating custom character sets

To define a custom character set, you need to know the Unicode character range for the language(s) you want to support. For example, the range for Arabic is 0600-06FF, while the range for Thai is 0E00 - 0E7F.

To create a custom character set:

1. Select **Tokenization > Character sets**.



2. Click **New Character set**.



3. Enter a **Name** and **Range** or set of ranges for your set.
The maximum size of a custom UTF character set definition string is 1536.
This implies either 100 ranges, or 200 individual characters, or some combination thereof—for instance, 50 ranges plus 100 individual characters.
4. Click **Create**.
The new set appears in the All Character Sets list, and is available to apply to tokenization templates.

Editing character sets

To edit a character set, click on its name in the list. Predefined character sets cannot be edited and should not be deleted.

Applying character sets

The three predefined character sets always appear in the drop-down list of character sets when you create a tokenization template. Once a custom set has been defined, it appears in the tokenization template creation menu as well.



NOTE: See [“Creating tokenization templates” on page 84](#).

Deleting character sets

Before deleting a character set, make sure that tokens created with this character set are not stored anywhere within (or outside) of the organization.



Caution: After the character set used to create these tokens has been deleted, they can no longer be detokenized, unless the character set is recreated very precisely. Even the order of the characters in the character set matters.

To delete a character set:

1. Navigate to **Tokenization > Character Sets**.
2. Select the character set from the list, and click **Delete**.

Prototyping tokenization

CTS supports UTF-8 inside the `valuenam` field for "data" and "token".

Generate a file that has the JSON request in it containing this UTF-8 code, and then submit it via cURL.

A file can be submitted in cURL as shown here:

```
--data-binary @filename option.curl --tlsv1.2 -k -X POST -u  
superuser:ssl12345 https://127.0.0.1/vts/rest/v2.0/tokenize --  
data-binary @tokenizerequest.txt
```

Managing tokenization templates

Tokenization templates allow you to define the specifics of how you want your data tokenized and detokenized.

For example, you can specify that your data be Luhn checked before tokenizing, or that the last four digits of a data string be left unencrypted, or that an identifying prefix be attached to all tokens.

To use tokenization, at least one tokenization template must be created.

Creating tokenization templates

1. In the CTS GUI, select **Tokenization > Tokenization Templates**.
2. Click **New Tokenization Template**.
See [Figure 4](#) on page -15.
3. Enter the following values:

- **Name** - Give the new template a name. This name will be used in the API POST commands tokenize and detokenize.
- **Token group** - The tokenization group for which this template can be used. You must create at least one template for each tokenization group.
- **Format** - Specify which of the following formats to use when tokenizing data:
 - Format-preserving encryption (FPE) with or without a Luhn check. A Luhn check ensures that a submitted number is a valid credit card number.
Note: FPE is an alternate name for FF3.
 - Format-preserving encryption (FF1) with or without a Luhn check. A Luhn check ensures that a submitted number is a valid credit card number.
 - Random or Random with Luhn check. This format is optimized for use with numbers from 9 digits (without a Luhn digit, for instance SSNs) to 19 digits (including a Luhn digit), like credit card numbers or tax ID numbers. If the data includes more free-form text, such as names and addresses, either the FPE or FF1 format is preferable.



In a random or random-Luhn format, the template uses a pseudo-random number generator (PRNG) instead of an encryption algorithm. In addition, the very first symmetric key configured in the CTS GUI is not a true encryption key, but instead acts as a seed value for the PRNG.

- One of the supported date formats. Use this to anonymize dates. Between the month, date, and year portions of the date format, the input data can include one of the following separators: slash (/), comma (,), hyphen(-), or space (). Date formats are used along with the Start Year and End Year fields.



In a date format, if a two-digit year format is selected (for example, DDMMYY), the CTS assumes that the date is in the twenty-first century; that is, it assumes the first two digits of the year are 20.

- **Character Set** - The character set to use with tokenization.
- **Prefix** - (Optional) An optional prefix to be added to all tokens. Example: **CC**-1234-1234-1234-1234 or **SSN**:123-23-1234. Note that the format is no longer preserved with this feature.
- **Keep Left** - The number left digits to leave unencrypted. You need at least a total of two digits to tokenize. Cannot be left blank.
- **Keep Right** - The number right digits to leave unencrypted. You need at least a total of two digits to tokenize. Cannot be left blank.

- **Irreversible** - (Optional) Check this box if you never want the token to be detokenized. For example, if you have production data that you want to tokenize for test or development, and you know you'll never want to decrypt it.
- **Allow empty or 1 character inputs** - Enables null, empty, and one-character inputs, which are passed through. The token value returned is the original input value. The single-digit input supports both characters and numerals, and is available for all formats except RANDOM.

4. Click **Create**.

Minimum input characters with keepleft or keepright

If you configure using keepleft or keepright, note the following minimum numbers of input characters, according to tokenization format:

- For FPE: 2 characters plus the number of keepleft or keepright characters
- For RANDOM Numeric: 9 characters plus the number of keepleft or keepright characters
- For RANDOM Alphaumeric: 5 characters plus the number of keepleft or keepright characters



IMPORTANT: To edit a tokenization template, click on a template name.

Sample: Use a tokenization template in the REST API

Once you specify a tokenization template, it can be used in the RESTful API application code. This is shown in the following example:

```
curl --tlsv1.2 -X POST -u EdYee:EdYee_password -d '{"tokengroup" :
"store1", "data" : "9453677629008564", "tokenTemplate" :
"ctsUsersTemplate"}' https://192.168.12.88/vts/rest/v2.0/tokenize
```

Deleting tokenization templates

Before deleting a tokenization template, make sure that tokens created with this tokenization template are not stored anywhere within (or outside) of the organization. Once the tokenization template used to create these tokens has been deleted, they can no longer be detokenized, unless the tokenization template is recreated very precisely.



Caution: After the character set used to create these tokens has been deleted, they can no longer be detokenized, unless the character set is recreated very precisely. Even the order of the characters in the character set matters.

To delete a tokenization template:

1. Navigate to **Tokenization > Tokenization Templates**.
2. Select the tokenization template from the list, and click **Delete**.

Managing data masks

Tokenization data masks hide specified parts of detokenized data. A mask is required for detokenization, even if the user is authorized to view the entire unencrypted string of data.

Creating a data mask

Review [“Using data masks” on page 15](#).

To create a mask:

1. , Select **Tokenization > Masks**.
The All Masks list is displayed.
2. Click **New Mask** and enter the information on the *New Mask* dialog.
Some examples of data masks you could create:
 - A mask called SHOW_LAST_4 that shows the last four digits of a tokenized credit card: XXXX-XXXX-XXXX-7897
 - A mask called SHOW_FIRST_3 that shows only the first three digits of a social security number: 565-XX-XXXX.
 - A mask that shows everything (ALL_CLEAR): set **Show Left** and **Show Right** values to 999999.
3. Click **Create**.

Deleting a data mask

When a Mask is deleted, all user and user group permissions associated with that mask will be deleted.



Warning! This could result in production down time if a user is using the system and their permissions are deleted. They will be denied tokenization and detokenization rights. Be VERY sure these permissions are not in use before deleting the mask.

Applying Permissions to Users and User Groups

Review the following sections:

- [“Understanding CTS Authentication and Authorization” on page 4](#)
- [“Understanding CTS Keys” on page 11](#)

With CTS, users are authenticated, granted permissions (to tokenize, detokenize, encrypt, hash, etc.), and associated with particular keys, using a three-way permissions matrix.



NOTE: See additional help on the *Permissions* page by clicking the “i” icon in the upper right corner.

Assigning key-based permissions

The GUI Administrator assigns permissions based on user (or user group), action, and key, similar to the English sentence:

“User ctsuser1 may encrypt, decrypt, tokenize and detokenize with symmetric key key1.”

Before permissions can be assigned, you must:

- Create user groups and users
- [“Specifying DSM Keys”](#)

Use the following steps to assign key-based permissions:

1. Log in to the CTS GUI and select **Permissions**. The User Permissions list is displayed. If necessary, click **User Group** to see the User Group Permissions list. You can click the tabs to see the Symmetric Keys, Asymmetric Keys, and Opaque Objects assigned to a user (or group).

Figure 9: User Permissions List Page



2. Click a user or group name to see a user permissions summary page. You can review and change permissions there or from the list page. To continue from the list page:
3. Click the permission symbol for a particular user's key. The user permission matrix is displayed.



4. Assign permissions as needed and click **Apply**.

Assigning other permissions

The permissions in the “Other Permissions” category are those that are not tied to specific keys. They are assigned only once per user group, and apply to all key groups. Actions that are not supported in the API are not displayed.

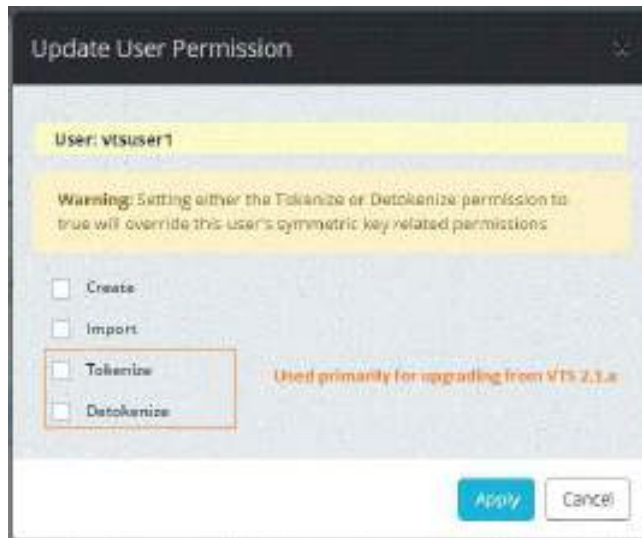
Creating and Importing Keys



IMPORTANT: Review [“Creating static and dynamic keys” on page 12.](#)

To grant a user or user group permission to create keys on the DSM or to import keys to the DSM:

1. Log in to the CTS GUI and click **Permissions**. The User Permissions list is displayed.
(If necessary, click **User Groups** to access the User Group Permissions list.)
2. Click the permission symbol for a user in the **Other Permissions** column.
The Other Permissions assignment window is displayed.



3. Check **Create** and **Import**, as desired, and click **Apply**.



NOTE: The **Create** permission is assigned only once per user and applies to all type of keys.

The **Import** permission is assigned only once per user and applies to symmetric keys and opaque objects.

Other permissions: tokenize/detokenize

Select **Tokenize/Detokenize** to apply privileges at a global level, on all symmetric keys in the system.

If your system is upgraded from an earlier version, any users who had been granted tokenize/detokenize privileges have these boxes checked automatically.

Remember to de-select the global check boxes and reassign privileges on a per-key basis to take advantage of the more granular structure of CTS configuration options.

Dashboard API Monitoring

The CTS Dashboard provides graphical and tabular monitoring for:

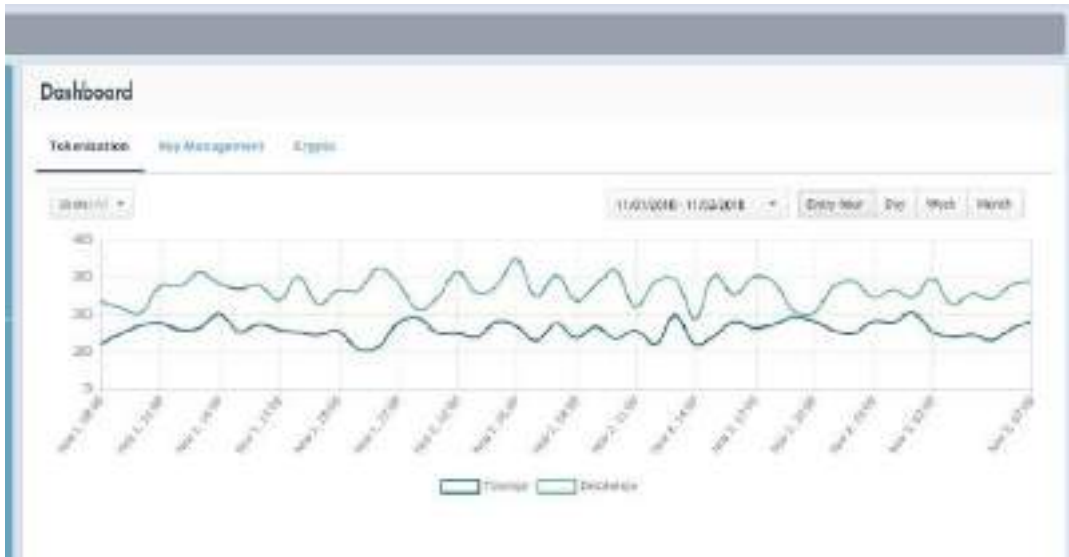
- Tokenization activity
- Key Management activity
- Cryptography activity

Activity data is updated hourly, and can be sorted and filtered by:

- User (or all users)
- Date range
- Hour
- Day
- Week
- Month

The tabular view of the API data (shown below the graphical view) can be filtered by user or by user IP address (which can be used to check for access by unusual IP addresses). Clicking a user or IP address row in the table filters the graph view above it by that selection

Figure 10: API Activity on the Dashboard



Exporting Data

Click the **Export** button to export into a CSV file the data as currently filtered and shown in the dashboard.

Administrator History Logs

Most tokenization and detokenization log messages are in the file `tokenization.log`. You can get this via web download with credentials from the file `tokenization.log`. Example:

`https://ctshost.vormetric.com/log/tokenization.log`

These log messages also go to the syslog with the facility code `daemon`.

Other log files:

- `django.request.log` - Logs any user request coming from the Tokenization Server GUI.
- `uwsgi_cts.log` - Debug log.
- `clish.log` - CLI activities and events.



IMPORTANT: Log data added by the server or server framework are not included in the log files

View the Tokenization Server logs by clicking the **Logs** icon in the Tokenization Server GUI. (Audit log messages also go to the syslog with the facility code user.) From here you can:

- View audit logs, access logs (login and logout attempts), active sessions, locked accounts, and server CPU usage
- Export the logs

Viewing audit logs

To view the audit logs:

1. Log into the CTS GUI.
2. Click **Logs** in the navigation pane.
3. Click the **Audit Logs** tab.

Details are shown for each audited event, including:

- **Date** specifies when the event occurred.
- **User** is the Tokenization user.
- **Content Type** is the database table that was modified. *Tenant* involves Group Tokens, *Key* is encryption keys, *user* is for users, *groups* for groups, and so on.
- **Object Name** is the specific database object that was created or changed.
- **Modification** is the action that was performed.



NOTE: To set the CTS log level, see [“Set Log Levels” on page 141](#).

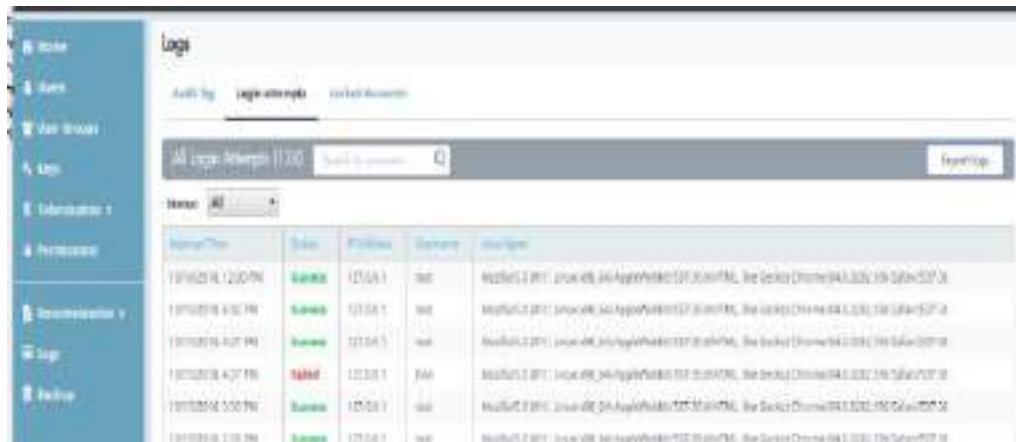
Viewing access logs

The access log shows information about all attempts to log into the server, and all logouts from it. The related information includes the access time, the IP address, username, and user agent (typically, a web browser). By default, all login attempts are listed, but the list can be filtered (by Status) to show only successful or only failed attempts.

To view the access logs:

1. Log into the CTS GUI.
2. Navigate to **Logs**.
3. Click the **Access Logs** tab.
4. Filter the listed results with the **Type** menu. When viewing the **Logins** access type, use the additional **Status** pull-down menu to filter by Successful or Failed login attempts.

Figure 11: Logs Page Showing Login Attempts



Name/Time	Status	IP/Device	Location	User Agent
1/10/2018 1:20:00 PM	Success	172.16.1.1	net	Mozilla/5.0 (Windows; MSIE 10.0; Trident/6.0; Chrome/44.0.2404.125; Safari/537.36)
1/10/2018 1:40:00 PM	Success	172.16.1.1	net	Mozilla/5.0 (Windows; MSIE 10.0; Trident/6.0; Chrome/44.0.2404.125; Safari/537.36)
1/10/2018 1:40:00 PM	Success	172.16.1.1	net	Mozilla/5.0 (Windows; MSIE 10.0; Trident/6.0; Chrome/44.0.2404.125; Safari/537.36)
1/10/2018 1:40:00 PM	Failed	172.16.1.1	net	Mozilla/5.0 (Windows; MSIE 10.0; Trident/6.0; Chrome/44.0.2404.125; Safari/537.36)
1/10/2018 1:40:00 PM	Success	172.16.1.1	net	Mozilla/5.0 (Windows; MSIE 10.0; Trident/6.0; Chrome/44.0.2404.125; Safari/537.36)
1/10/2018 1:40:00 PM	Success	172.16.1.1	net	Mozilla/5.0 (Windows; MSIE 10.0; Trident/6.0; Chrome/44.0.2404.125; Safari/537.36)

Downloading (export) access or audit logs

To download the history logs for an administrator:

1. Log into the CTS GUI.
2. Navigate to **Logs**.
3. Click the **Audit Logs** or **Access Logs** tab.
4. Click the **Export** button for the Audit or Access Logs.

Showing active sessions

To see a list of active sessions of users currently logged into the server:

1. Log into the CTS GUI
2. Click **Logs** in the navigation pane.
3. Click the **Active Session** tab.

4. To end a user's session on the GUI, click the **End Session** button in the row for that user.

Figure 12: List of Active Sessions



Showing locked accounts

If a user fails to enter valid credentials more than five times, the user is locked from accessing the CTS for 15 minutes. View the details of locked accounts, including when the lock was activated and when it was released. Superusers can unlock an account before the 15 minute time limit is completed.

To view and manage locked accounts:

1. Log into the CTS GUI.
2. Click **Logs** in the navigation pane.
3. Click the **Locked Accounts** tab.
4. Unlock an account by clicking **Unlock** in the row of the desired user.

System logging

Audit log messages go to the syslog with the facility code user.

tokenization.log messages go to the syslog with the facility code daemon.

CLI log messages go to the syslog with the facility code local3.

Configuring syslog servers

Configure remote syslog servers with the CLI, with the command under the vts category. You can also set up multiple syslog servers with this command, with the --add and --remove commands.

The remoteLog command has utilities to check the connection status to the servers, and to send a sample log message to them.

Backing Up and Restoring the CTS

Create regular backups of the server to restore the CTS in the event of failure.

Even in a clustered scenario, system failure can occur. To minimize this risk back up your CTS implementation regularly using the administration interface or the REST API.

Use the CTS GUI to create a backup before upgrading your CTS version, and before any major server configuration changes. Maintain a regular backup schedule using the CTS REST API.

Restore the state of your CTS implementation using the administration interface. Backups restore the CTS to the saved version in the following scenarios:

- Restoring the application after system failure
- Restoring the application after upgrade failure
- Restoring the application due to unrecoverable system configuration errors.



NOTE: A backup cannot restore a system using a CTS version older than 2.2.2. A backup cannot be applied to any version of the CTS software that is older than the version in use when the backup was created.

Backing up the CTS in the GUI

Use the administration interface to create and save a backup.

Use the following steps to create a system archive:

1. Select **Backup** from the administration interface left hand navigation.
2. Click the **Create Backup** tab.
3. Accept the default values, and click **Backup now**.

The archive generates and is saved to your default download folder. The file name is created with the following syntax:

`backup_year-month-day.`

For example: `backup_2018-10-20`

Move the backup to your storage directory.

Managing backup tasks

Use the REST API to create a backup and automate system backup tasks.

You can use the Backup API to generate backups via any valid HTTP request. The syntax requires authentication using user credentials.

See the reference documentation for more information.

Restoring the CTS from a backup

If a complete system restore is required, make sure the CTS installation is complete before beginning this task.



NOTE: Make sure the CTS version leveraged for the backup is the same as the backup or newer than the backup, and is at least version 2.2.2

Use the following steps to restore the CTS to its last recorded working state using a backup.

1. Select **Backup** from the administration interface left hand navigation.
2. Click the **Restore** tab.
3. Click **Choose File** and navigate to the required backup archive.
4. Confirm the following back up details:
 - CTS version
 - Backup date
 - DSM hostname
 - Encryption key name

Make sure the same DSM that was registered to the CTS that was backed up is also registered to the new CTS. Also make sure that the key used to encrypt the backup file has been exported to the registered DSM, and that it has the exact same name.

5. Click **Restore**.



NOTE: Depending on the size of the backup, the restore process may take some time.

If this node is a member of an existing cluster, it is removed from the cluster. Any existing data on this node is destroyed.

A new cluster is created with this node as the first node but with data restored from the backup.

When the restore completes, this node has all data from the backup including login credentials.



NOTE: If user forgot the login credentials in the backup, run the `createsuperuser CLI` command to create a new user.

This node becomes a new node of a new cluster. It is no longer a member of any previous cluster.



IMPORTANT: If this node was a member of an existing cluster prior to the restore, it must be removed from that cluster. Failure to remove this node from its previous cluster prevents new nodes from joining that cluster.

The CTS is restored to the state of archive. When the task is complete, a banner appears, indicating that the restore is complete. The CTS reboots.

When the login screen appears, enter your credentials to access the restored application.

Key Rotation

Regular encryption key rotation can increase security and also satisfy some PCI DSS requirements. To rotate CTS keys, ask your DSM Administrator to run the DSM CLI command `security gencert` on the DSM that creates and stores the CTS keys.

Versioned keys are not permitted for tokenization.

Monitoring via SNMP

You can monitor disk partitions, files, load average, and processes by setting up SNMP polling. Configure SNMP at the CLI in the network category, and the `snmp` command.

When enabling SNMP, you must specify a whitelist of IP addresses for SNMP managers. CIDR control access is supported.

CTS supports both SNMP v3 and v2c. SNMP v3 supports password authentication, encryption, and a choice of encryption and password hashing algorithms.

CTS CLI Reference

The CipherTrust Tokenization Server Command Line Interface (CLI) is used to configure the CTS network and do other system-level tasks.

This chapter summarizes the following CLI categories and the commands within each:

- [“Authentication Setup Category” on page 99](#)
- [“Cluster Management Category” on page 100](#)
- [“ICAPI SafeNet Setup” on page 100](#)
- [“Network Management Category” on page 101](#)
- [“System Configuration Category” on page 103](#)
- [“VAE Configuration Category” on page 104](#)
- [“CTS Configuration Commands” on page 104](#)

This chapter provides a high-level overview of each command category. For complete usage details on each command and its options, run the command with the “- -help” option. For tips on navigating the CLI, see [“CLI navigation cheat sheet” on page 36](#).



NOTE: To access the CLI, see [“Access the CLI through a terminal emulator like PuTTY.” on page 35](#).

Authentication Setup Category

The `auth` category contains authentication setup commands -- for OAuth2 (`oauth2` command), for setting and reviewing optional OAuth2 settings, and the `ldap` command, whose options configure and review the LDAP Server settings.

As an alternative to specifying the various LDAP settings as options to the `ldap` command (for example, `auth ldap --server ldap://hostname:636`), you can run an interactive wizard that prompts you for each setting. Run this wizard with the `ldap --setup` command.

Cluster Management Category

The `cluster` category is a subset of CTS commands for use with cluster management.

Table 7: Cluster Management Commands

	Description
<code>add</code>	Add a node that can later create or be joined to an existing cluster. This command must be used on a node before that node can join a cluster. NOTE: Do not use with <code>remotejoin</code> .
<code>apiadmin</code>	API admin user setup, for automated cluster set up.
<code>create</code>	Create a CTS cluster.
<code>join</code>	Join a node to an existing CTS cluster. NOTE: Do not use with <code>remotejoin</code> .
<code>remotejoin</code>	Remotely join nodes to a cluster. Use in automated cluster setups.
<code>remove</code>	Remove a node from a cluster.
<code>show</code>	Show cluster settings.

ICAPI SafeNet Setup

Use the `icapi` category of CTS commands to set up, configure, and control SafeNet server support (via ICAPI).

Table 8: ICAPI Setup Commands

Command	Description
<code>create</code>	Create a private key, or certificate signing request.
<code>enable</code>	Enable or disable the ICAPI interface to SafeNet servers.
<code>register</code>	Register to a SafeNet KeySecure host.
<code>set</code>	Set ICAPI properties, such as IP addresses, ports, and user credentials to a KeySecure server.
<code>setup</code>	Perform an interactive setup of ICAPI properties (instead of using <code>set</code>).
<code>show</code>	Show ICAPI and SafeNet status and settings.
<code>test</code>	Test connection to a KeySecure server, or access to a specified key on the KeySecure server.

Table 8: ICAPI Setup Commands

Command	Description
upload	Upload a private key, signed certificate, or local CA certificate.

Network Management Category

Use the network category to set, modify, or delete system IP addresses, and to set up CipherTrust Tokenization Servers. The network category also contains commands for showing and testing network configurations



NOTE: Each server is assigned a unique IP address.

Table 9: Network Category Commands

Command	Description
arp	Manipulate the system ARP cache.
arping	Send ARP requests to a neighbor host
checkport	Check port connection status of a host
dns	Show or configure settings for the CTS DNS server(s)
ethtool	Query network driver and hardware settings
ifconfig	Show or configure a network interface
ip	Show or configure the CTS network interface settings
netstat	Print network connections
nslookup	Query internet name servers
ntpservice	NTP service
ping	Pings an IP address, host name, or FQDN
route	Set static route
service	Start/stop/restart network service
set	Configure network device interface

Table 9: Network Category Commands

Command	Description
setup	Network setup wizard
show	Show network device configuration
snmp	Show, set, or change SNMP (v2c or v3) server configuration, which can be used to monitor processes, load average, disk space, and file size
traceroute	Traces route to IP address or host name

Example Request:

```
network> ip address
```

Response:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 00:0c:29:47:22:72 brd ff:ff:ff:ff:ff:ff
    inet 10.3.201.140/16 brd 10.3.255.255 scope global dynamic eth0
        valid_lft 216sec preferred_lft 216sec
    inet6 fe80::20c:29ff:fe47:2272/64 scope link
        valid_lft forever preferred_lft forever
```

WARNING! Changes made using this command are not persistent. Use the setup command to change network settings permanently.

System Configuration Category

The system configuration category enables you to configure the CTS host settings (such as, hostname, console port, or timezone), and view and manage other common system administration tasks for the server (such as, reboot or shut down, or view disk space usage or currently logged-in users).

Table 10: System Category Commands

	Description
banner	Configure SSH banner
cpuid	Show detailed CPUid information
cpuinfo	Show CPU information
daemon	Show daemon services that are running
date	Print or Set the system date and time
df	Report file system disk space usage
free	Display amount of free and used memory in the system
host	DNS lookup utility
hostname	Set/Show Hostname
hosts	Update hosts file
lastlogin	Show last logged-in users
meminfo	Show memory information
reboot	Reboot system
shutdown	Shut down system
terminal	Show or terminate terminal (CLI) sessions
timezone	Set or show timezone
top	Display Linux processes
uname	Print system information
uptime	Show how long the system has been running
vmstat	Report virtual memory statistics
who	Show who is logged into the system

VAE Configuration Category

The `vae` category includes commands to register the Tokenization Server to the DSM..

Table 11: VAE Management Commands

	Description
<code>certificates</code>	VAE certificates utilities
<code>dsmkeepalive</code>	Enable or Disable DSM keepalive
<code>register</code>	Register the current CTS host to the DSM.
<code>show</code>	Show registration information.
<code>test</code>	Test to be sure the CTS has been successfully registered with the DSM.

CTS Configuration Commands

The `vt s` category is used to configure the CTS, set time parameters, configure GUI login behavior, and upgrade Tokenization Server software.



NOTE: The CTS configuration commands still retain the `vt s` category name to ensure backward-compatibility.

The `vt s` category consists of the following commands:

Table 12: CTS/VT S Category Commands

Command	Description
<code>batch_size_limit</code>	Tokenization Batch size limit.
<code>cliadmin</code>	Cliadmin user setup. NOTE: If CTS runs on an AWS EC2 and has been set to use an EC2 key pair, password access is disabled, and <code>cliadmin password</code> commands will fail with an error message. For example, a password cannot be disabled or unlocked. To view if the <code>cliadmin</code> is using an AWS SSH key pair, run <code>vt s cliadmin keys --list CLI</code> command. If the instance uses an EC2 key pair, an entry in the SSH keys will show <code>[ec2]</code>.

Table 12: CTS/VTs Category Commands

Command	Description
client_certificate	Enables client authentication by requiring clients to pass a client key and CA certificate to CTS. NOTE: For best security, enable client certificates <i>with</i> identities (the <code>--enable-with-id</code> option) instead of without identities.
cors	Class-Origin Resource Sharing (CORS)
create_superuser	Create a superuser local account for Web administrative GUI access
logfile	List, view, or tail log files
loglevel	Set log level
randommode	Random Mode utility
rate_limiting	Limit the amount of tokenization requests
remotelog	Set remote logging
restore	Restore system from backup
server_certificate	Creates the self-signed certificate or certificate signing request and import signed certificate.
service	CTS Service utility
show	Show the security settings.
smtp	Configure, view, or test email notifications and underlying SMTP setup
ssl_ciphers	Show or set SSL cipher suites. WARNING: Non-FIPS approved SSL ciphers will be deprecated in the future. It is highly recommended you configure CTS to use the default SSL cipher suite (with the <code>vtsssl_ciphers --setdefault</code> command), otherwise the server can be open to security vulnerabilities.
ssl_protocol	Specifies what version of the SSL (TLS) is supported by the CTS.
upgrade	Upgrade CTS.
weblogin	Web login setup.

Encryption and Key Management API Programming Notes

CTS includes several sets of REST API services: Encryption, Key Management, Tokenization, and Admin. The Tokenization APIs (including the reference documentation, examples, and error codes) are described in the chapter [“Tokenization REST API” on page 133](#). The Admin APIs are described in the chapter [“Admin REST API” on page 155](#).

This chapter provides concepts, guidelines, and examples for the Encryption and Key Management APIs. There are also online reference documents for these two sets of services.

This chapter contains:

- [“Reference Documentation” on page 107](#)
- [“Encryption and Key Management API Concepts” on page 108](#)
- [“Key Management Examples” on page 120](#)
- [“Cryptographic operation examples” on page 123](#)
- [“Versioned Key Management Examples” on page 130](#)

Reference Documentation

The CipherTrust Encryption and Key Management services have built-in online documentation. To access this documentation, you must be logged in.

CipherTrust Encryption RESTful API documentation

https://<CTS_hostname_or_IP_address>/vts/crypto/v1/doc

CipherTrust Key Management RESTful API documentation

https://<CTS_hostname_or_IP_address>/vts/km/v1/doc

Encryption and Key Management API Concepts

The concepts in this section provide background context for API programmers who will use the Encryption and Key Management APIs in their code.



NOTE: The data input format for the APIs is base64.

TES Resource Names (TESRN)

A TESRN is a reference to a resource with the Crypto and Key Management services, specifically a way to address keys. These are used throughout all the cryptographic and key management operations that require a key reference.

It has the following format:

```
tesrn:<managing-service>:[<domain>][:<resource>]:[resource-  
type]:<resource-id>[:<version>|<public>|<private>]
```

The intention is to be able to identify where the resource resides and whether the service can handle such or not. There exists some abbreviated ways to access keys:

Reference keys by label

Quickly get information about a key by using the key label in your request: *foo123* key would be "*foo123*". In TESRN format it would be "*tesrn:cts::label:foo123*". Note that the domain is empty, and the resource type, which is absent from the name, defaults to "key".

Other default key types

In addition, you may want to use an alias, UUID, or MUID instead of a key label. For these cases, refer to each respectively as shown in the following examples:

By Alias:

```
alias:myfoo
```

TESRN format:

```
tesrn:cts::alias:myfoo
```

By UUID:

```
uuid:7fa96ada-18c6-11e8-8dab-0800277415ff
```

TESRN format:

```
"tesrn:cts::uuid:7fa96ada-18c6-11e8-8dab-0800277415ff"
```

By MUID:

```
muid:7fa96ada-18c6-11e8-8dab-0800277415ff
```

```
urn:7fa96ada-18c6-11e8-8dab-0800277415ff
```

TESRN format:

```
"tesrn:cts::muid:91eb92c2-18c6-11e8-accb-0800277415ff91ec0054-18c6-11e8-b01e-0800277415ff"
```

By Label:

```
label:foo123
```

```
"tesrn:cts::label:foo123"
```

Legacy resource names

While not specific to the CTS services, it is worth mentioning WSOP (nShield Web Services Option Pack) as a future interoperability consideration. WSOP keys are accessed by UUID only. These key references are expressed as:

```
urn:c1d6f102-18c6-11e8-b2f9-0800277415ff
```

Algorithms and key types

The CipherTrust Encryption and Key Management APIs make references throughout the specification to *algorithm types*. The algorithm types have names that conform to the JOSE RFC-7518 Specification, plus a decision to extend the algorithm namespace (maintaining the JOSE syntax) to meet customer needs.



NOTE: The algorithm name will be **case insensitive** for use with the services. The complete list of algorithm names is listed in Table 13 on page 110. Key types also conform to the JOSE RFC-7518 section 6.1 specification. Although this section of the specification describes the key types as **case sensitive** for JOSE, the current implementation is **case insensitive** for key types.

Table 13: Key Types

	Description	Algorithms	State
Symmetric	Oct or octet sequence. Used for: - tokenize - detokenize - encrypt - decrypt Symmetric keys are also used for Tokenization, and while the symmetric keys are typical AES keys with 128 or 256 bits of key material, the algorithms are FPE (FF3) and FF1. Note: Versioned keys may not be used for tokenization, only non-versioned keys.	A128ECB - AES 128 ECB cipher mode A256ECB - AES 256 ECB cipher mode A128CBC - AES 128 CBC cipher mode A256CBC - AES 256 CBC cipher mode A128GCM - AES 128 GCM cipher mode A256GCM - AES 256 GCM cipher mode A128CBCPAD - AES 128 CBC-PAD cipher mode A256CBCPAD - AES 256 CBC-PAD cipher mode A128CTR - AES 128 CTR (counter) cipher mode A256CTR - AES 256 CTR (counter) cipher mode	At this time, only symmetric keys support states. On create and import, only a preactive or active state may be given. On a key update all states can be given except for preactive. The following states are available: - Preactive - Active - Default when a key is created - Suspended - Compromised - Deactivated - Destroyed
Asymmetric	RSA key type. Used for: - encrypt - decrypt - sign - verify	RSA	
Opaque object	Octet array. This is a Thales extension (not part of JOSE RFC-7518 spec). Used for - sign - verify Does not have the option of store on server. Size limit: 0-4096 bytes.	HS1 - HMAC-SHA-1 HS224 - HMAC-SHA-224 HS256 - HMAC-SHA-256 HS384 - HMAC-SHA-384 HS512 - HMAC-SHA-512	N/A



NOTE: EC Key type is an elliptic curve cryptographic key, which is currently not supported.

Opaque objects

An opaque object is a generic form of object that is securely stored on the DSM and is currently used by the underlying VAE PKCS11 library to implement some extensions that are not available natively on the DSM.

CTS and VAE primarily use opaque objects to represent HMAC keys. They are created or imported using the same endpoint as for symmetric and asymmetric keys, except that the key type (kty) must be opaque. If a wrapped key is provided, this becomes the key material used to create the opaque object key. If only a size is given, the CTS cryptographic service generates a key using the DSM true random number generator.

The size limit for an object is 4096 bytes. When size is omitted, or set to zero, and key bytes are present, the size is computed from the size of the byte sequence. When an opaque object has empty key bytes with no specific size defined, it creates an object.



NOTE: In this release opaque objects can only be used for signing and verifying. Opaque objects support cryptographic header versions 2.1 and 2.7 only.

Table 14: Opaque Object Creation Rules

	Key Bytes	Result
size = 0 (or omitted)	"" (or omitted)	returns error
size = 0 (or omitted)	present	sets size to the length of key bytes
size > 0	length does not match size	returns error
size > 0	"" (or omitted)	generates Opaque Object with size bit key
size > 0	length does not match size	returns error

Cryptographic headers

Cryptographic headers is a convenient feature that allows a ciphertext to be decrypted without knowledge of the specific key name and/or version that was used to encrypt it. When encrypting with headers enabled, the ciphertext includes bytes containing key information inserted before the encrypted data. To enable cryptographic headers, the **header** attribute must appear in the POST body for sign, verify, encrypt, and decrypt calls.



NOTE: The two versions of headers supported in this release are 2.1 and 2.7 (HDR_V21, HDR_V27).

Verify and decrypt also take an **all** header attribute value to let the service figure out the header version in a more transparent way for the caller.

Versioned keys

Key versioning is available for symmetric keys in the current release of CTS with the VAE library. The rules for versioned keys are different from conventional symmetric keys. In general:

- The label represents the key class. A reference to the key by label will always refer to the most recent activated key version.
- Every key version has its own UUID and MUID
- A key can be deleted by UUID or MUID
- Deleting version zero (0) of a key class by UUID or MUID, deletes the entire key class (all versions)
- Versioned keys can be used for signing, verifying, encryption and decryption
- To create a new versioned key, the create endpoint must have an **action** attribute set to **create**. A version number must also be supplied.

Some limitations for this release are:

- Inability to find a particular version of a key by label and version number
- Internal key IDs are not supported

Forcing refreshed key information from DSM

If a key is rotated on the DSM, then retrieving that key information via CTS API calls will reflect the key information prior to that DSM rotation. You can ensure refreshed key information is fetched by CTS API calls by appending “:refresh” to the end of the **kid** value of the API call. For example:

```
curl -X GET \
https://192.168.159.141/vts/km/v1/keys/vae-key-rotation-via-cryptoserver:refresh\
-H 'authorization: Basic dnRzcm9vdDpWb3JtZXRYaWMxMjMh' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-H 'postman-token: 2a69ef4b-c849-8a97-1db5-2a9e93c594a1'

curl -X POST \
https://192.168.159.141/vts/crypto/v1/encrypt \
-H 'authorization: Basic dnRzcm9vdDpWb3JtZXRYaWMxMjMh' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-H 'postman-token: 7fb55525-eea2-b12f-1add-66bdee3e34e4' \
-d '{ "plaintext": "NDU0NTQ1", "alg": "A128CTR", "params" :
{"iv": "Ff0lom2eDm0phLNHLvpUvA==" }
, "kid": "vae-key-rotation-via-cryptoserver:refresh" }'
{ "ciphertext": "6VhMegkk", "tag": "" }
```

This does not apply to keys that are rotated by CTS, which will always be current, only to keys rotated on the DSM.

Key states

Symmetric and asymmetric keys support states. The following states are available:

- Preactive - When a key was created to have a preactive state
- Active - Default when a key is created
- Suspended - A key has been suspended
- Compromised - Key is compromised
- Deactivated - Key is deactivated
- Destroyed - Key is destroyed

To set a key to the suspended, compromised, or deactivated state, call the modify endpoint (PATCH). A key can be set to the active state again only if it is in the suspended state.

Valid key operations

This table lists the valid key operations from the API. It does not include permissions related to authorization. See the Authorization section below for more details.

Table 15: Valid Key Operations

Operation	Symmetric Key	Versioned Symmetric Key	Asymmetric Key	Opaque Object
Create	yes	yes	yes	yes
Import	yes	yes	no	yes
Destroy	Key delete custom attribute delete	Delete key by UUID or MUID only Delete key of version 0 deletes entire set Delete custom attribute	Key delete custom attribute delete	Key delete custom attribute delete
Modify	custom attributes add alias key states key migration to versioned key	custom attributes add alias key states key rotation lifespan interval	custom attributes	custom attributes add alias
Find	By label/name By UUID By MUID By alias	By label/name - last version By UUID By MUID By alias	public key only By label/name	By label/name By UUID By MUID By alias
Export	clear_key kid for plain bytes wrapping key with A256CBC or A256CBCPAD	clear_key kid for plain bytes wrapping key with A256CBC or A256CBCPAD	no	no
Encrypt	Standard encryption cryptographic headers	Standard encryption cryptographic headers	Standard encryption only	no
Decrypt	Standard decryption cryptographic headers	Standard decryption cryptographic headers	Standard decryption only	no
Sign	HMAC signing cryptographic headers	HMAC signing cryptographic headers	RSA signing only	HMAC signing cryptographic headers

Table 15: Valid Key Operations (Continued)

Operation	Symmetric Key	Versioned Symmetric Key	Asymmetric Key	Opaque Object
Verify	HMAC verifying cryptographic headers	HMAC verifying cryptographic headers	RSA verifying only	HMAC verifying cryptographic headers

Wrapping keys

When importing or exporting a key, it is recommended to use a wrapping key in order not to compromise the key itself.

In the current version of the CTS Key Management module a wrapping key must be an AES 256-bit key, and only the CBC or CBCPAD ciphers may be used. The ideal method for wrapping a key on both import and export is to use asymmetric keys, however such is currently unsupported by the underlying libraries and services which implement the CTS Key Management module.

Importing an exporting key requires that the user has a wrapping key, and that such key is available on both the client and the DSM.

Importing with wrapping key

When importing with a wrapping key, the following steps need to be taken:

1. Create a 256-bit AES key that is known to the caller and the DSM. This is the wrapping key.
2. Encrypt the key bytes with the wrapping key using an AES-CBC or AES-CBCPAD cipher. This is the wrapped key.
3. Call the create endpoint and supply the base64-encoded wrapped key and the wrapping key ID.

Exporting with wrapping key

When exporting with a wrapping key, the following steps need to be taken:

1. Create a 256-bit AES key that is known to the caller and the DSM. This is the wrapping key.
2. Call the export endpoint, supplying a wrapping key ID. This will return a wrapped key.
3. Decrypt the base64 encode wrapped key using the wrapping key on the client side.

Importing and exporting using clear bytes

The API offers the ability to perform import and export function using raw key bytes, however such method is not recommended.

To import a key using clear bytes, omit the wrapping key ID or give the **clear_key** label. For exporting, give the **clear_key** label as the wrapping key ID. In both cases, the **alg** and the **iv** attributes must be omitted.

Authorization

Using any key on CTS requires user or group level authorization for a particular key. The following sections describe tasks associated with authorization:

- [“Creating and importing keys” on page 116](#)
- [“Managing keys” on page 116](#)
- [“Cryptographic operations” on page 116](#)
- [“Authorization operations” on page 117](#)

Creating and importing keys

This category covers both key import and key creation. A user or group that has either import, create, or both is allowed to add keys to the CTS. The difference between import from create is in the ability to add a wrapped key representing the key itself to the key creation request. This ability to create and import keys is set for each user through the CTS Administration UI.

If a user or group has create and/or import permissions, when invoking the corresponding API endpoint will create a key with all the permissions for key management and cryptographic operations. Any other user needing to perform operations on such keys, can only be added through the CTS Administration UI. If the user does not have permission to create or import, the corresponding creation endpoint will return an error.

Managing keys

A user or group may perform each of the following key management operations if he or she has permissions to perform such. This can be configured for each individual key from inside the CTS Administration UI. The allowed key management operations are:

- Destroy - Destroy a key
- Modify - Modify a key’s attributes
- Find - Look up or retrieve a specific key
- Export - Export a specific key. This only applies to symmetric keys.

Cryptographic operations

A user or group may perform each of the following cryptographic operations using a specific key if he or she has permissions to perform such as configured for each individual key from inside the CTS Administration UI. The allowed cryptographic operations available are:

- Encrypt - Encryption using a key
- Decrypt - Decryption using a key
- Sign - Sign with a given key
- Verify - Verify a signature with a given key

Authorization operations

Table 16, “Authorization Operations,” on page 117, shows the possible authorization operations when a key is or was created or imported via the API. The owner is the user who creates the key. This table only applies to authorization. Table 15, “Valid Key Operations,” on page 114 should be used to determine what are valid operations, without taking authorization into account.

Table 16: Authorization Operations

Operation	Owner	Group	Other user	Other Group
Create	Yes, if owner has create permission Owner can specify on create the permissions allowed though the usage attribute If usage attribute is not supplied, key will be created with all permissions	Yes, if owner has create permission granted through a group Owner can specify on create the permissions allowed though the usage attribute If usage attribute is not supplied, key will be created with all permissions	N/A	N/A
Import	Yes, if owner has import permission Owner can specify on import the permissions allowed though the usage attribute If usage attribute is not supplied, key will be imported with all permissions	Yes, if owner has import permission granted through a group Owner can specify on import the permissions allowed though the usage attribute If usage attribute is not supplied, key will be imported with all permissions	N/A	N/A
Destroy	Yes, if owner gave it permission on creation Key if removed from the CTS Authorization table unless it is a versioned key	No	No	No
Export	Yes, if owner gave it permission on import or create	No	No	No

Table 16: Authorization Operations (Continued)

Operation	Owner	Group	Other user	Other Group
Find	Yes, if owner gave it permission on import or create	No	No	No
Modify	Yes, if owner gave it permission on import or create	No	No	No
Encrypt	Yes, if owner gave it permission on import or create	No	No	No
Decrypt	Yes, if owner gave it permission on import or create	No	No	No
Sign	Yes, if owner gave it permission on import or create	No	No	No
Verify	Yes, if owner gave it permission on import or create	No	No	No

Size

All sizes throughout the API refer to size in bytes, with the exception of the key creation/import where the size is in bits. The maximum size for the random generator endpoint is 4096 bytes.

Payload limits

The total payload for any request is 5 MB. However, you can work around this size limit by using the chaining script shown in [“Encrypting and Decrypting Large Files” on page 126](#).

Date handling

Throughout the API specification, dates are all represented in RFC-3339 format. Any date that appears within the request or result body in JSON format will be in this format. As a general pointer, when using different programming languages in writing client code to access the APIs, we prepared the following:

Go

The time struct/interface in golang is already marshaled/unmarshaled for JSON in RFC339.

Python

Python requires the installation of a package that implements rfc3339. A "*pip install rfc3339*" will address most users needs.

```
Python rfc3339
import rfc3339, time
rfc3339.rfc3339(time.time())
```

PHP

RFC3339 is a predefined constant in date for PHP: `date(DATE_RFC3339)`

Bash/cURL

RFC-3339 dates can be obtained by typing: `date --rfc-3339=seconds`

C++/boost

```
#include <boost/date_time.hpp>
std::cout.imbue(std::locale(std::cout.getloc(), new
boost::local_time::local_time_facet("%Y-%m-%dT%H:%M:%S%F%Q")));
std::cout <<
boost::local_time::local_microsec_clock::local_time(boost::local_time::ti
me_zone_ptr()) << std::endl;
```

Successful responses

All successful requests return a 200 HTTP status.

All calls return a response body, except for destroy (DELETE) of a specific key which has an empty body.

In a batch or vectorized call, where a series of requests are done simultaneously, the result is always 200, and each element of the vector contains the corresponding result for each request, whether successful or not.

Error responses and error codes

Error responses from the cryptographic and key management layer only return a JSON body with the following format. Anything different from this format is from the NGINX reverse proxy or from other components.

For example:

```
{"error": "TES_INVALID_OBJECT_HANDLE", "message": "find object returned
an invalid handle", "status": 400, "timestamp": "2018-03-
08T11:56:38.793756418-08:00"}
```

The following table describes the response parameters by listing the attribute names and their corresponding descriptions:

Table 17: Response Parameters

Attribute	Description
error	This response parameter indicates an error is returned. The error codes start with a TES_ for service related error checking and a CKR_ for PKCS11 related error codes encountered.
message	
status	This response parameter provides the HTTP status code associated with the error. For a list of possible error codes see: Table 18, "Error HTTP Status Codes," on page 120.
timestamp	

Table 18: Error HTTP Status Codes

	Description
400	An HTTP 400 Bad Request status indicating that the user supplied a malformed or invalid request.
401	An HTTP 401 Unauthorized status indicating that the user need to login or authenticate before the request can be handled.
403	An HTTP 403 Forbidden status indicating that a logged in user does not have authorization to access or perform an operation on the given resource.
404	An HTTP 404 Not Found status indicating that the URL or resource (TESRN) references does not exist.
500	An HTTP 500 Server Internal Error indicating that an error occurred on the service side and that it may be available later.

Key Management Examples

Key creation

Key creation allows to create a key whose contents (key material or bytes) is generated by the DSM



IMPORTANT: You cannot create a key via the REST API and make it a seed key in one step. You must create the key and add it via the CTS GUI in separate steps. You cannot just pass in "seedkey" as a keyword to the create key REST API. If you do, the key is created, but it is not a seed key.

This example shows how to create an AES 128-bit key named foo123key:

Request:

```
POST /vts/km/v1/keys {  
  "size": 128, "name": "foo123key", "kty": "oct"  
}
```

Response:

```
200 OK  
{  
  "attributes": {"usage": ["encrypt", "decrypt", "sign", "verify", "destroy", "modify", "find", "export"], "size": 16}, "metadata": {"key_management": {"uuid": "9f3c2775-405b-33e8-91af-904faa0e551f", "caching_duration": 44640, "state": "active", "lifespan_unit": "days", "muid": "9f3c2775-405b-33e8-91af-904faa0e551f39b6aee0-6821-39f9-a186-9a9904abff32", "caching_enabled": true, "name": "foo123key"}}, "kid": "tesrn:vts::label:foo123key"}  
}
```

Key import

Key import is basically the same as a key create, however the request contains information on the wrappedkey and the wrapping key.

This request imports an AES 128-bit key named foo1234key with key bytes equal to: "HelloWorldAlways".

Request:

```
POST /vts/km/v1/keys{  
  "size": 128,  
  "name": "foo1234key",  
  "kty": "oct",  
  "wrappedkey": "SGVsbG9Xb3JsZEFsd2F5cw=="  
}
```

Response:

```
200 OK
{"attributes":{"usage":["encrypt","decrypt","sign","verify"],"size":16},
"metadata":{"key_management":{"uuid":"9f3c2775-405b-33e8-91af-904faa0e551f",
" caching_duration":44640,"state":"active","lifespan_unit":"days",
"muid":"9f3c2775-405b-33e8-91af-904faa0e551f39b6aee0-6821-39f9-a186-9a9904abff32",
" caching_enabled":true,"name":"foo123key"}},"kid":"tesrn:cts::label:foo1234key"}
```

Key delete

This example deletes a key named foo123key.



NOTE: Upon success, the HTTP status code must be 200.

Request:

```
DELETE /vts/km/v1/keys/foo123key
```

Response:

```
200 OK
```

Key update

Request:

```
PATCH /vts/km/v1/keys/fookey {"state": "suspended"}
```

Response:

```
200 OK
{"attributes":{"usage":["encrypt","decrypt","sign","verify","destroy","find"],
"size":16},
"metadata":{"key_management":{"uuid":"bfd2a7b7-20db-31d7-b22f-01e625a54961",
" caching_duration":44640,"state":"suspended",
" versioning_enabled":1,"lifespan_unit":"days",
"muid":"bfd2a7b7-20db-31d7-b22f-01e625a549618b0ce745-fcb2-3718-97b3-d4a0b415ac5d",
" caching_enabled":1,"name":"fookey"}},"kid":"tesrn:cts::label:fookey"}
```


Response:

```
200 OK
{"digest": "RfdwuYKwBn0l03ZMGB7r59S+P+T9/s8ufPJfcSQXNfE="}
```

Encryption examples

AES-256 CBC encryption example

Request:

```
POST /vts/crypto/v1/encrypt{
  "plaintext": "JLns9BI4Pa7TTrCaeJIjQQ==",
  "alg": "A256CBC",
  "params": {
    "iv": "tler+lrCnchFRJDCFs1F/A=="},
  "kid": "mykey123"
}
```

Response:

```
200 OK
{"u"ciphertext": "SZASu6Ts+01IMtb14+djhg==", "params": {}}
```

Simple AES-256 CTR encryption with headers example

Request:

```
POST /vts/crypto/v1/encrypt
{"plaintext": "JLns9BI4Pa7TTrCaeJIjQQ==", "alg": "A256CTR", "params":
{"iv": "tler+lrCnchFRJDCFs1F/A=="}, "kid": "myversionedkey",
"headers"="hdr_v2.7"}
```

Response:

```
200 OK
{"ciphertext":
"UktNQzIxMAD/////AAAABXV1awQAAAAAEDkKfAiGhjeLkp50NDgjPQT/////AAAAA2l
2AAAAABbhYmMxMjNkZWZnaDY3ODkw/////wAAAVjc3VtAAAAACAbtDVcSesXu7SYqoZ
xNR+hC6Cq8H3gE5xv5vvSzRhUrKcwnBu8", "params": {}}
```

Decryption examples

Simple AES-256 CBC encryption example

Request:

```
POST /vts/crypto/v1/decrypt
{"ciphertext":
"wInPTVRugkxP0KBm46yCo8lpQMaDDRo15bEwt9Mz629MKh7IqB9roh36ZkyIumLi",
"alg": "A256CBC", "params": {"iv": "ek5Lmcw0JATvux5PoSaZ6w=="},
"kid": "fookey"}
```

Response:

```
200 OK
{"plaintext":
"40idw01cBKKcF1tK4ZBdZg7yDe9KbvBP1npfs382HT+yz0Fb3CeCh/eBIhH7a1rb",
"params": {}}
```

Simple AES-256 CTR decryption with headers example

Request:

```
POST /vts/crypto/v1/decrypt
{"ciphertext":
"UktNQzIxMAD/////AAAABXV1awQAAAAAEDkKfAiGhjeLkp50NDgjPQT/////AAAAA2L
2AAAAABbhYmMxMjNkZWZnaDY3ODkw////wAAAAVjc3VtAAAAACAbtDVcSesXu7SYqoZ
xNR+hC6Cq8H3gE5xv5vvSzRhUrKcwnBu8", "alg": "A256CTR", "params":
{"iv": "tler+lrCnchFRJDCFs1F/A=="}, "headers"="all"}
```

Response:

```
200 OK
{"plaintext": "JLns9BI4Pa7TTrCaeJIjQQ==", "params": {}}
```


Encrypting and Decrypting Large Files

To handle files that are larger than the payload limit of 5 MB, use the following script to automatically break down a large file into a chain of smaller files for encryption, and to join these files and decrypt them:

```
#!/bin/bash

URL=http://localhost:8000/crypto/v1
KEY=cc01
IV="MTAwMDIwMDAzMDAwNDAwMA=="
CHUNKSIZE=256
FILE="/etc/hosts"
ALG="A256CTR"

BASE=`basename $FILE`
OIV=$IV
if [ ! -e ./$BASE ]
then
    cp $FILE .
fi
rm -f ${BASE}.* *-${BASE}.*
split -b $CHUNKSIZE ./$BASE ${BASE}-
NOMORE=0

echo "===== Encryption ====="
for i in ${BASE}.*
do
    if [ $NOMORE -eq 1 ]
    then
        break
    fi

    PTEXT=`cat $i | base64 -w 0`
    PAYLOAD=$(jq -n -r \
        --arg kid $KEY \
        --arg iv $IV \
        --arg pt "$PTEXT" \
        --arg alg $ALG \
        '{ plaintext: $pt, alg: $alg, kid: $kid, params: {iv: $iv} }'
    )
```

```

RESULT=`echo $PAYLOAD | curl -Ss -X POST -k -d @- -H "Accept: application/json"
$URL/encrypt`
    NEWIV=`echo $RESULT | jq -r '.params.iv'`
    CTEXT=`echo $RESULT | jq -r '.ciphertext'`
    if [ -z "$NEWIV" ] || [ "$NEWIV" == "null" ]
    then
        NOMORE=1
    fi
    IV=$NEWIV
    echo -n $CTEXT | base64 -d -w0 > ${i}.enc
done

cat ${BASE}/*.enc > ./${BASE}.enc
split -b $CHUNKSIZE ./${BASE}.enc enc-${BASE}-
IV=$OIV
NOMORE=0

echo "===== Decryption ====="
for i in enc-${BASE}-*
do
    if [ $NOMORE -eq 1 ]
    then
        break
    fi
    CTEXT=`cat $i | base64 -w0`

    PAYLOAD=$(jq -n -r \
        --arg kid $KEY \
        --arg iv $IV \
        --arg ct "$CTEXT" \
        --arg alg $ALG \
        '{ ciphertext: $ct, alg: $alg, kid: $kid, params: {iv: $iv} }'
    )

```

```
RESULT=`echo $PAYLOAD | curl -Ss -X POST -k -d @- -H "Accept: application/json"
$URL/decrypt`
NEWIV=`echo $RESULT | jq -r '.params.iv'`
PTEXT=`echo $RESULT | jq -r '.plaintext'`
if [ -z "$NEWIV" ] || [ "$NEWIV" == "null" ]
then
    NOMORE=1
fi
IV=$NEWIV
echo -n $PTEXT | base64 -d -w0 > ${i}.dec
done
cat enc-${BASE}/*.dec > ${BASE}.dec
diff ${BASE}.dec ${BASE}
echo diff status = $?
```

The large file is broken into chunks of size whose modulo 16 equals 0 (that is, into chunk sizes that are multiples of 16 bytes).

When submitting the chunks to the encryption endpoint, the response includes an IV that is used to encrypt the next chunk. If the payload length is not 16-byte aligned, the response does not return the next IV, assuming that chaining is complete.

For decryption, the same rules apply. The first call must use the same IV as the call used for encrypting. The ciphertext chunk size can be a different size from the original plaintext encryption request, as long as the chunks are 16-byte aligned.

For example a 1030 byte file can be encrypted in 8 chunks of 128 bytes each and a last one of 6 bytes. The same file once encrypted can be split in 3 chunks of 320 bytes and a last one of 70 bytes.

Sign

Simple HMAC-SHA256 sign example

Request:

```
POST /vts/crypto/v1/sign
{"alg": "HS256", "payload": "aGVsbG8gd29ybGQh", "kid": "mykey"}
```

Response:

```
200 OK
{"signature": "13pCxjIXjY8MNhsx69szyU0VraQc6Tn0C0irqCut0Ls="}
```

Verify

Simple HMAC-SHA256 verify example

Request:

```
POST /vts/crypto/v1/verify
{"alg": "HS256", "signature":
"13pCxjIXjY8MNhsx69szyU0VraQc6Tn0C0irqCut0Ls=", "payload":
"aGVsbG8gd29ybGQh", "kid": "mykey"}
```

Response:

```
200 OK
{"valid": 1}
```

Random

Simple random example

The following example shows a simple request to generate an 8-byte random number:

Request:

```
POST /vts/crypto/v1/random
{"seed": "q8qqyw==", "size": 8}
```

Response:

```
200 OK
{"random": "152LM5GzL8U="}
```

.....

This section shows how to use versioned keys within the Cryptographic and Key Management set of APIs.

Versioned key creation

Versioned key creation example

Creation of an AES 256-bit versioned key named *vkey256* with each version of the key with a lifespan of 7 days.

Request:

```
POST /vts/km/v1/keys
{"size": 256, "name": "vkey256", "kty": "oct", "versioning_enabled":
true, "version": 0, "version_action": "create", "lifespan_interval":
7}
```

Response:

```
200 OK
{"attributes": {"usage": ["encrypt", "decrypt", "sign", "verify",
"destroy", "modify", "find", "export"],
"size": 16},
"metadata": {"key_management": {"uuid": "9f3c2775-405b-33e8-91af-
904faa0e551f",
"caching_duration": 44640,
"state": "active",
"lifespan_unit": "days",
"muuid": "9f3c2775-405b-33e8-91af-904faa0e551f39b6aee0-6821-39f9-a186-
9a9904abff32",
"caching_enabled": True,
"name": "vkey256"}},
"kid": "tesrn:vts::label:vkey256"}
```

Versioned key import example

Creation of an AES 256-bit versioned key named *vkey256* with each version of the key with a lifespan of 7 days.

Request:

```
POST /vts/km/v1/keys
{"size": 256, "name": "vkey256", "kty": "oct", "versioning_enabled":
true, "version": 0, "lifespan_interval": 7, "wrappedkey":
"SGVsbG9Xb3JsZEFsd2F5cw=="}
```

Response:

```
200 OK
{"attributes": {"usage": ["encrypt", "decrypt", "sign", "verify",
"destroy", "modify", "find", "export"],
"size": 16},
"metadata": {"key_management": {"uuid": "9f3c2775-405b-33e8-91af-
904faa0e551f",
"caching_duration": 44640,
"state": "active",
"lifespan_unit": "days",
"moid": "9f3c2775-405b-33e8-91af-904faa0e551f39b6aee0-6821-39f9-
a186-9a9904abff32",
"caching_enabled": True,
"name": "vkey256"}},
"kid": "tesrn:cts::label:vkey256"}
```

Standard key to version key migration example

Creation of an AES 256-bit versioned key named *vkey256* with each version of the key with a lifespan of 7 days.

Request:

```
PATCH /vts/km/v1/keys/existingkey {"version_action": "migrate"}
```

Response:

```
200 OK
{
  "attributes": {
    "size": 32
  },
  "metadata": {
    "key_management": {
      "uuid": "7f460ca4-04e2-37d9-8d57-b2c36fbed55e",
      "caching_duration": 44640,
      "state": "active",
      "versioning_enabled": true,
      "moid": "7f460ca4-04e2-37d9-8d57-b2c36fbed55ecd222707-123a-
3df5-baef-c9b1f77ac3fb",
      "caching_enabled": 1,
      "name": "existingkey"
    }
  },
  "kid": "tesrn:cts::uuid:7f460ca4-04e2-37d9-8d57-b2c36fbed55e"
}
```

Version key rotation example

Rotation of an existing versioned key named `vkey256` with each version of the key with a lifespan of 7 days.

Request:

```
PATCH /vts/km/v1/keys {"version_action": "rotate"}
```

Response:

```
200 OK
{"attributes": {"usage": ["encrypt", "decrypt", "sign", "verify",
"destroy", "modify", "find", "export"],
"size": 16},
"metadata": {"key_management": {"uuid": "d0a031de-5102-3bba-80f4-
7019306da03e",
"cached_duration": 44640,
"state": "active",
"lifespan_unit": "days",
"muid": "d0a031de-5102-3bba-80f4-7019306da03ed0a031de-6821-39f9-
a186-9a9904abff32",
"cached_enabled": True,
"name": "vkey256"}},
"kid": "tesrn:cts::label:vkey256"}
```

Tokenization REST API

Learn about the CTS Tokenization API in this chapter. Also note that the Tokenization APIs are also described in an online reference guide, under **Documentation** in the CTS Administration GUI navigation bar.

The CipherTrust Tokenization API consists of:

- **tokenize** - A POST call that tokenizes data. A user submits sensitive data, the CipherTrust Tokenization Server returns a token. The token is then stored in the production database.
- **detokenize** - A POST call that detokenizes data. A user submits a token to the CipherTrust Tokenization Server and the actual data is returned detokenized. Permission settings for that user dynamically specify the appropriate data mask.
- **log** - Changes the log level. Four different log levels (error, info, debug and trace) can be selected. Chose 'error' or 'info' for normal operation, or 'debug' when instructed to do so by Thales eSecurity support.

This chapter includes the following sections:

- [“Using the API” on page 134](#)
- [“Create a Token” on page 134](#)
- [“Get Detokenize Resource Data” on page 138](#)
- [“Set Log Levels” on page 141](#)
- [“Tokenization Examples” on page 143](#)
- [“Tokenization API Call Failure Error Messages and Workarounds” on page 148](#)



NOTE: See [“Encryption and Key Management API Programming Notes” on page 107](#) for information about the Encryption and Key Management REST APIs. The Admin REST APIs are described in [“Tokenization REST API” on page 133](#).

Using the API

To enable tokenization, you do not have to change your database schema, because CipherTrust Tokenization preserves the format of your data. However, you must change your application code.

When a user makes a request to tokenize or detokenize, the username and password must be provided. This is a part of a REST request. The username/password is validated by either the AD/LDAP server or CTS, depending on where the user is defined.



NOTE: The data passed to and from the Tokenization Server—user name, tokenization group name, tokens, and sensitive data—is case-sensitive, and the application code that passes data to CTS must be aware of this.



NOTE: Tokenization input can be a normal text format.

Create a Token

Description

Use the tokenize endpoint to encrypt a data string to a token using format-preserving encryption (FPE or FF1).

Table 19: tokenize Endpoint Details

Data	Description
HTTP Request	<code>https://CTS_IP_Address/vts/rest/v2.0/tokenize</code>
Method	POST
CTS IP Address	The IP address of the CTS to access with the API call.
Response Format	JSON
Required Authentication	A valid user name and password must be passed as an argument. The values are case sensitive.

The parameters for tokenize endpoint are described in the table below:

Table 20: tokenize Parameters

Parameter	Required	Type	Description
tokengroup	Y	string	Defines a group name space in the configuration database. For example, let's say you are a credit card company and you want the CTS to support three departments supporting Store1, Store2, and Store3. You want employees in these departments to have access to data for those respective stores. Two actions are required: - The application code must specify a different tokengroup parameter in the POST calls for each group supported (example: Store1, Store2, and Store3). The tokengroup parameter is case-sensitive. - The CTS GUI administrator must create token groups that match the tokengroup parameter names used in the application code. If you do not support multiple token groups, then tokengroup will always be the same value. Example value: store1
data	Y	string	The data string to tokenize. The argument considers case and is limited to 128KiBs Example value: 1432445695309134
tokentemplate	Y	string	CTS GUI Administrator-defined name for a group of properties that define tokenization operation. Properties include the token group to which the template applies, the tokenization format (date, FPE or FPE with Luhn check, or FF1 or FF1 with Luhn check, or Random (lookup) or Random with Luhn check), number of leftmost or rightmost characters to <i>not</i> tokenize, whether you wish to never detokenize a tokenized entry, the character set used for tokenization, and an optional prefix for tokens. Example value: Credit Card
returnData	N	string	Tokenization requests can return the plaintext string together with the encrypted data in the response. To enable, add the "returnData" parameter and set it to true.

The tokenize endpoint response parameters, their types, and corresponding descriptions are listed in the table shown below:

Table 21: tokenize Response Parameters

	Type	Description
token	string	Token for the sensitive data. Example value: 2134234509874567
status	string	• Succeeded (if request succeeded) • Error (if request failed)
reason	string	If status is Error a reason is displayed. Otherwise the reason is omitted.
data	string	If returnData in the request is set to <code>true</code> , contains the plaintext string for the encrypted data.

Create token CURL example



NOTE: If you use the `--tlsv1.2` CURL Command syntax option, you must use CURL version 7.34 or higher.

Example Request:

```
curl -k -X POST -u Donald_Duckmiller:Panda45# -d '{"tokengroup" :  
"ctsUsers-t", "data" : "9453677629008564", "tokenTemplate" :  
"ctsUsersTemplate" }' https://192.168.118.140/vts/rest/v2.0/tokenize
```



NOTE: The user name (Donald_Duckmiller) and tokengroup (ctsUsers-t) are case sensitive.

Example Response

```
{"token": "CC-0332335756020172", "status": "Succeed"}
```



IMPORTANT: If FPE-luhn, FF1-luhn, or Random-luhn is selected in the Change Token Template page of the CTS GUI, the data must pass a Luhn check or tokenization will fail with the error message:

```
{"status":"error","reason":"Luhn check failed for input data."}
```

Tokenizing multiple data items

Up to 1,000 data items can be tokenized in a single curl command by submitting a JSON array in the REST request. For example, to tokenize four data items you might enter the following in a postccn.txt file:

Example Request:

```
cat postccn.txt

[{"tokengroup" : "ctsUsers-t" , "data" : "9453677629008564",
"tokentemplate" : "ctsUsersTemplate" }, {"tokengroup" : "ctsUsers-
t" , "data" : "9453677629008564", "tokentemplate" :
"ctsUsersTemplate" }, {"tokengroup" : "ctsUsers-t" , "data" :
"9453677629008564", "tokentemplate" : "ctsUsersTemplate"
}, {"tokengroup" : "ctsUsers-t" , "data" : "9453677629008564",
"tokentemplate" : "ctsUsersTemplate" }]

curl -k -X POST -u Donald_Duckmiller:Panda45# --data-binary
@postccn.txt https://192.168.118.140/vts/rest/v2.0/tokenize
```



Warning! Newlines are NOT allowed in the input.

Example Response:

```
[{"token":" 9465976792116170", "status":"Succeed"},
{"token":" 9465976792116170", "status":"Succeed"},
{"token":" 9465976792116170", "status":"Succeed"},
{"token":" 9465976792116170", "status":"Succeed"}]
```

The tokenized data is returned in the CURL Command syntax response, in order of submission.

Get Detokenize Resource Data

Description

Get detokenized data from a resource..

Table 22: detokenize Endpoint Details

	Description
HTTP Request	<code>https://CTS_IP_Address/vts/rest/v2.0/detokenize</code>
Method	POST
CTS IP Address	The IP address of the CTS to access with the API call.
Response Format	JSON
Required Authentication	A valid user name and password must be passed as an argument. The values are case sensitive.

Table 23: detokenize Request Parameters

Parameter	Required	Type	Description
token	Y	string	The token to detokenize. token is case-sensitive. Example value: 6029541314537206
tokengroup	Y	string	Defines a group name space in the configuration database. For example, let's say you are a credit card company and you want the CTS to support three departments supporting Store1, Store2, and Store3. You want employees in these departments to have access to data for those respective stores. Two actions are required: - The application code must specify a different tokengroup parameter in the POST calls for each group supported (example: Store1, Store2, and Store3). The tokengroup parameter is case-sensitive. - The CTS GUI administrator must create token groups that match the tokengroup parameter names used in the application code. If you do not support multiple token groups, then tokengroup will always be the same value. Example value: store1

Table 23: detokenize Request Parameters

Parameter	Required	Type	Description
tokentemplate	Y	string	CTS GUI Administrator-defined name for a group of properties that define tokenization operation. Properties include the token group to which the template applies, the tokenization format (either FPE or FPE with Luhn check, or FF1 or FF1 with Luhn check), number of leftmost or rightmost characters to <i>not</i> tokenize, whether you wish to never detokenize a tokenized entry, the character set used for tokenization, and an optional prefix for tokens. Example value: Credit Card

Table 24: detokenize Response Parameters

	Type	Description
data	string	The detokenized sensitive data. May be masked depending on the configuration. data is case sensitive and limited to 128KiB characters.
status		Succeeded (if request successful) or Error (if request failed).
reason	string	If status is Error , then a reason will be displayed. Otherwise the reason will be omitted.

Example:



NOTE: If you use the --tlsv1.2 CURL option, you must use CURL version 7.34 or higher.

Request:

```
curl -k -X POST -u tok-user1:Panda45# -d '{"tokengroup" : "ctsUsers-t" ,
"token" : "CC-0332335756020172", "tokentemplate" : "ctsUsersTemplate" }'
https://192.168.118.140/vts/rest/v2.0/detokenize
```

If the client authentication feature is enabled (See also [“Upgrading CTS” on page 30.](#)) the example syntax is as follows:

```
curl -k --key client.key --cert client.crt -X POST -u tok-
user1:Panda45# -d '{"tokengroup" : "ctsUsers-t" , "token" :
"CC-0332335756020172", "tokentemplate" : "ctsUsersTemplate"
}' https://192.168.118.140/vts/rest/v2.0/detokenize
```

Response:

```
{"data": "9453677629008564", "status": "Succeed"}
```

Detokenizing multiple data items

Up to 1,000 data items can be detokenized in a single curl command by adding multiple data items. For example, to detokenize four data items, you might enter the following in the *posttoken.txt* file:

Request:

```
cat posttoken.txt

[{"tokengroup": "ctsUsers-t", "token": "CC-0332335756020172",
"tokentemplate": "ctsUsersTemplate" }, {"tokengroup": "
ctsUsers-t", "token": "CC-0332335756020172", "tokentemplate"
: "ctsUsersTemplate" }, {"tokengroup": "ctsUsers-t", "token": "
CC-0332335756020172", "tokentemplate": "ctsUsersTemplate"
}, {"tokengroup": "ctsUsers-t", "token": "CC-
0332335756020172", "tokentemplate": "ctsUsersTemplate" }]

curl -k -X POST -u tok-user1:Panda45# --data-binary
@posttoken.txt https://192.168.118.140/vts/rest/v2.0/detokenize
```

Response:

```
[{"data": "9453677629008564", "status": "Succeed"}, {"data": "9453677
629008564", "status": "Succeed"}, {"data": "9453677629008564", "statu
s": "Succeed"}, {"data": "9453677629008564", "status": "Succeed"}]
```

The detokenized data is returned in the CURL response in corresponding order of submission.



NOTE: Newlines are NOT allowed in the input.



IMPORTANT: If FPE-luhn, FF1-luhn, or Random-luhn is selected in the Change Token Template page of the CTS GUI, the data must pass a Luhn check or detokenization will fail with the error message:

```
{"status": "error", "reason": "Luhn check failed for input data."}
```

Set Log Levels

Description

Set the log level for the REST API.

Table 25: log Endpoint Details

Data	Description
HTTP Request	https://CTS_IP_Address/vts/rest/v2.0/log
Method	POST
CTS IP Address	The IP address of the CTS to access with the API call.
Response Format	JSON
Required Authentication	None

Table 26: log Request Parameter

Parameter	Required	Type	Description
loglevel	Y	Enum	loglevel can take one of the following values: - error - Default value used in normal production environment. - debug - Use only when directed by support. - info - Do not use unless maximum logging is required. NOTE: The performance of tokenization and detokenization operations are severely affected by a log level of “debug” and substantially affected by a log level of “info.” For maximum throughput, use a loglevel of “error.”

Table 27: log Response Parameter

Parameter	Type	Description
status	string	Success (if request succeeded) or Error (if request failed).

Example:



NOTE: If you use the --tlsv1.2 cURL option, you must use CURL version 7.34 or higher.

Request:

```
curl -k --tlsv1.2 -X POST https://127.0.0.1/vts/rest/v2.0/log --data-binary '{"loglevel":"debug"}'
```

Response:

```
{"status":"success"}
```

Tokenization Examples

Java client example

```
package com.vormetric.app;

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;

import org.apache.commons.codec.binary.Base64;

import com.jayway.jsonpath.*;

import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLSession;
import javax.net.ssl.SSLSocket;
import javax.net.ssl.SSLSocketFactory;

public class CTSClientSample {

    public static void main(String[] args) {
        new CTSClientSample().DoIt();
    }
    @SuppressWarnings("null")
    private void DoIt(){
        String credential = Base64.encodeBase64String("ctsuser1:Panda45".getBytes());
        String ccn[] = {
            "5471949763376677", "5545127221796024", "5353800637453171", "5139918930790601",
            "5267003853942275"
        };
        try {
            // Tokenize request
            String https_url = "https://cts-qa/vts/rest/v2.0/tokenize/";
            URL myurl = new URL(https_url);
            HttpsURLConnection con = (HttpsURLConnection)myurl.openConnection();
            String ccNum = "9453677629008564";
            String jStr =
                "{\"data\":" + "\"" + ccNum + "\", \"tokengroup\":" + "\"t1\", \"tokentemplate\":" + "\"Credit Card\"}";
            con.setRequestProperty("Content-length", String.valueOf(jStr.length()));
            con.setRequestProperty("Content-Type", "application/json");
            con.setRequestProperty("Authorization", "Basic " + credential);
            con.setRequestMethod("POST");
            con.setDoOutput(true);
            con.setDoInput(true);
            DataOutputStream output = new DataOutputStream(con.getOutputStream());
            output.writeBytes(jStr);
            output.close();
            BufferedReader rd = new BufferedReader(new InputStreamReader(con.getInputStream()));
        }
    }
}
```

```

String line = "";
String strResponse = "";
while ((line = rd.readLine()) != null) {
    strResponse=strResponse+line;
}
rd.close();
String token = JsonPath.read(strResponse, "$.token").toString();
con.disconnect();
System.out.println("Tokenize server: "+https_url);
System.out.println("Tokenize request: "+jStr);
System.out.println("Tokenize response: "+strResponse);

// Bulk tokenize
String jStrArray="[";
for (int i=0;i<ccn.length;i++) {
    jStrArray = jStrArray +
    "{\"data\":\""+ccn[i]+"\", \"tokengroup\":\"t1\", \"tokentemplate\":\"Credit Card\"}";
    if (i<ccn.length-1) {
        jStrArray = jStrArray + ",";
    }
}
jStrArray=jStrArray+"]";

con = (HttpsURLConnection)myurl.openConnection();
con.setRequestProperty("Content-length", String.valueOf(jStrArray.length()));
con.setRequestProperty("Content-Type", "application/json");
con.setRequestProperty("Authorization", "Basic "+credential);
con.setRequestMethod("POST");
con.setDoOutput(true);
con.setDoInput(true);
output = new DataOutputStream(con.getOutputStream());
output.writeBytes(jStrArray);
output.close();
rd = new BufferedReader(new InputStreamReader(con.getInputStream() ));
line = "";
strResponse = "";
while ((line = rd.readLine()) != null) {
    strResponse=strResponse+line;
}
rd.close();
System.out.println("Bulk Tokenize request: "+jStrArray);
System.out.println("Bulk Tokenize response: "+strResponse);
String[] tokens = new String[5];
for (int i=0;i<5;i++) {
    tokens[i]=JsonPath.read(strResponse, "$["+i+"].token").toString();
}

// Detokenize request
https_url="https://cts-qa/vts/rest/v2.0/detokenize/";
myurl = new URL(https_url);
con = (HttpsURLConnection)myurl.openConnection();
jStr="{\"token\":\""+token+"\", \"tokengroup\" :
\"t1\", \"tokentemplate\":\"Credit Card\"}";
con.setRequestProperty("Content-length", String.valueOf(jStr.length()));
con.setRequestProperty("Content-Type", "application/json");
con.setRequestProperty("Authorization", "Basic "+credential);
con.setRequestMethod("POST");

```

```

con.setDoOutput(true);
con.setDoInput(true);
output = new DataOutputStream(con.getOutputStream());
output.writeBytes(jStr);
output.close();
rd = new BufferedReader(new InputStreamReader(con.getInputStream() ));
line = "";
strResponse = "";
while ((line = rd.readLine()) != null) {
strResponse=strResponse+line;
}
rd.close();
con.disconnect();
System.out.println("Detokenize server: "+https_url);
System.out.println("Detokenize request: "+jStr);
System.out.println("Detokenize response: "+strResponse);

// Bulk DeTokenize
jStrArray="[";
for (int i=0;i<ccn.length;i++) {
jStrArray = jStrArray + "{\"token\":\""+tokens[i]+"\", \"tokengroup\" : \"t1\", \"tokentemplate\":\"Credit Card\"}";
if (i<ccn.length-1) {
jStrArray = jStrArray + ",";
}
}
jStrArray=jStrArray+"]";

con = (HttpsURLConnection)myurl.openConnection();
con.setRequestProperty("Content-length", String.valueOf(jStrArray.length()));
con.setRequestProperty("Content-Type", "application/json");
con.setRequestProperty("Authorization", "Basic "+credential);
con.setRequestMethod("POST");
con.setDoOutput(true);
con.setDoInput(true);
output = new DataOutputStream(con.getOutputStream());
output.writeBytes(jStrArray);
output.close();
rd = new BufferedReader(new InputStreamReader(con.getInputStream() ));
line = "";
strResponse = "";
while ((line = rd.readLine()) != null) {
strResponse=strResponse+line;
}
rd.close();
con.disconnect();
System.out.println("Bulk Detokenize request: "+jStrArray);
System.out.println("Bulk Detokenize response: "+strResponse);

} catch (MalformedURLException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
}
}
}

```

C# client example

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Linq;
using System.Text;
using System.Net;
using System.IO;
using System.Threading.Tasks;
using RestSharp;
using RestSharp.Authenticators;
using System.Windows.Forms;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;

namespace CTS2ClientSample
{
    class Program
    {
        static public bool MyRemoteCertificateValidationCallback(System.Object sender,
            X509Certificate certificate, X509Chain chain, SslPolicyErrors sslPolicyErrors)
        {
            return true; // unsafe! Do not do this in production!
        }

        static void tokenize(RestClient client)
        {
            var request = new RestRequest("vts/rest/v2.0/tokenize", Method.POST);
            request.RequestFormat = DataFormat.Json;
            request.AddBody(new { tokengroup = "t1", tokentemplate = "Credit Card",
data = "1234567812345670" });
            // execute the request
            IRestResponse response = client.Execute(request);
            var responsestatus = response.ResponseStatus;
            var content = response.Content; // raw content as string

            MessageBox.Show("Tokenize API returns: " + content,
responsestatus.ToString(), MessageBoxButtons.OK);
        }

        static void detokenize(RestClient client)
        {
            var request = new RestRequest("vts/rest/v2.0/detokenize", Method.POST);
            request.RequestFormat = DataFormat.Json;
            request.AddBody(new { tokengroup = "t1", tokentemplate = "Credit Card",
token = "1234567812345670" });
            // execute the request
            IRestResponse response = client.Execute(request);
            var responsestatus = response.ResponseStatus;
            var content = response.Content; // raw content as string
        }
    }
}
```

```
    MessageBox.Show("Detokenize API returns: " + content, responsestatus.ToString(),
    MessageBoxButtons.OK);
}

    static void Main(string[] args)
    {
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
        ServicePointManager.ServerCertificateValidationCallback =
MyRemoteCertificateValidationCallback; // unsafe! Do not do this in production!

        var client = new RestClient("https://10.10.10.10");
        client.Authenticator = new HttpBasicAuthenticator("username",
"password");

        tokenize(client);

        detokenize(client);
    }
}
```

Tokenization API Call Failure Error Messages and Workarounds

These error messages may occur when an Tokenization API call is made.

Table 28: Error Message Reference

Message	Description
--tlsv1.2: is unknown	Only CURL version 7.34 or later supports the --tlsv1.2 option. If you get this error message, upgrade to CURL version 7.34 or higher.
API call aborted and no JSON vector elements are processed	These are error messages which pertain to the whole JSON vector. In other words, in these cases no vector elements are processed at all. The whole operation is aborted
User name and/or password mismatch.	Due to a wrong user name and/or password, access to the Tokenization Server is denied.
Unknown JSON keyword encountered. The valid keyword for the log request is loglevel.	For the REST API called "log", the only valid keyword on the left of the JSON value-name-value tuple is "loglevel".
Could not parse JSON input, perhaps it is too long.	An input size of 1 MByte cannot be exceeded. It could also mean that the JSON input is malformed.
User name and/or password missing	Either the user name, or the password, or both are missing from the request.
This user does not have permission to tokenize.	While the user exists and the password is correct, this particular user does not have the permission to tokenize. Consider assigning this user the permission to tokenize via the GUI.
User name found, but the user is not marked active.	While the user exists and the password is correct, this particular user is not marked as an active user. Consider marking this user an active user in the GUI.
Invalid operation. Use /vts/rest/v2.0/tokenize or /vts/rest/v2.0/detokenize	An invalid URL was specified. Allowed operations are /vts/rest/v2.0/tokenize, /vts/rest/v2.0/detokenize, or /vts/rest/v2.0/log.
No password specified.	While a user name was specified, the password is missing.
User name not found.	The specified user name exists neither in the local database which is configured via the GUI, nor in the optional LDAP database.
LDAP user found in local database, but LDAP is disabled.	Presumably due to a prior use of LDAP, an LDAP user was cached in the local database, but LDAP has since been disabled, disallowing authentication with this user. Prior to using LDAP users for authentication, at least one of the groups this user belongs to according to the LDAP database must be configured as a user group via the GUI.

Table 28: Error Message Reference (Continued)

Message	Description
API call aborted, one JSON vector element fails, others are processed	These messages pertain to only one element of the JSON vector. In these cases, one vector element may result in an error, but all other vector elements are still processed.
The length of the token exceeds 128 Kbytes.	The length of a token is limited to 128 KiB. However, the token supplied was longer than 128 KiB.
Unknown JSON keyword encountered. Valid keywords for the detokenize request are tokengroup, token and tokentemplate.	For the detokenize operation, only three specific keywords are allowed on the left side of a JSON tuple: "tokengroup", "token", and "tokentemplate". Please review and correct your JSON string.
The length of the data exceeds 128 Kbytes.	The length of the input data is limited to 128 KiB. However, the input data supplied was longer than 128 KiB.
Unknown JSON keyword encountered. Valid keywords for the tokenize request are tokengroup, data and tokentemplate.	For the tokenize operation, only three specific keywords are allowed on the left side of a JSON tuple: • tokengroup • data • tokentemplate
Please review and correct your JSON string.	A token group was specified in the JSON input, but it doesn't exist in the GUI-administered configuration database. Consider adding this token group via the GUI.
The requested key name was not found on the DSM	A key name was configured via the GUI and now used in an operation, however this key name does not exist on the DSM .
The token group specified was not found in the database	Consider creating this key name on the DSM .
tokengroup not specified in JSON request.	The JSON tuple named "tokengroup" is missing from a JSON vector element. A tokengroup must be specified.
The requested combination of tokentemplate and tokengroup was not found in the tokentemplate table.	Either the tokengroup specified has not been entered via the GUI, or the tokentemplate has not been entered via the GUI, or the token template of the same name has been entered, but for a different tokengroup. Please ensure that the tokentemplate you are trying to use has been entered for the tokengroup you are trying to use.
It is not allowed to specify the FPE-luhn token format with a non-zero keepright value.	After accounting for keepleft and keepright, not enough input characters are left to successfully tokenize the input. When you use the FPE token format, the number of input characters must exceed the sum of the Keep Left value and the Keep Right value by at least two. When you use the FPE-luhn token format, the number of input characters must exceed the sum of the Keep Left value and the Keep Right value by at least three.
There are not enough input characters for the detokenize operation.	At least two input characters are needed in the token in order to try to detokenize it.

Table 28: Error Message Reference (Continued)

Message	Description
Token format FPE-luhn is only compatible with an alphabet consisting of the 10 decimal digits (in ASCII encoding).	The FPE-luhn token format cannot be used with an alphabet containing letters and/or special characters. It must be used with an alphabet only containing the 10 digits.
After accounting for the keepleft and keepright requirements, not enough input characters left to tokenize.	When you use the FPE token format, the number of input characters must exceed the sum of the keepleft value and the keepright value by at least two.
Luhn check failed for input data.	The token format FPE-luhn was specified, but the input data sports an incorrect Luhn checksum. Please check your input data, or change the token format from FPE-luhn to FPE.
After accounting for keepleft and keepright, there are not enough input characters present to successfully tokenize.	When you use the FPE token format, the number of input characters must exceed the sum of the keepleft value and the keepright value by at least two. When you use the FPE-luhn token format, the number of input characters must exceed the sum of the keepleft value and the keepright value by at least three.
Token does not begin with the prefix specified in the database.	While a certain token prefix is specified in the token template which is being used in the JSON input, the token seen in the JSON input does not sport this particular token prefix.
The specified token only contains the prefix and no payload.	While the token seen in the JSON input sports the correct token prefix, there are no other characters present in the token.
Luhn check failed for input token.	The token format FPE-luhn was specified, but the token specified sports an incorrect Luhn checksum. Please check your input data.
There are not enough input characters present to successfully detokenize.	When you use the FPE token format, the number of input characters, not counting the token prefix, must exceed the sum of the keepleft value and the keepright value by at least two. When you use the FPE-luhn token format, the the number of input characters, not counting the token prefix, must exceed the sum of the keepleft value and the keepright value by at least three.

API Reference

For the complete Tokenization API reference, open the CTS GUI and navigate to:

Documentation > Tokenization API

Key Management Service REST API

Use the CTS Key Management Service (KMS) REST APIs to manage CTS keys.

Requirements

To use the KMS REST APIs, use:

- The CipherTrust Token Server 2.3.0 or newer
- Any keystore version compatible with CTS 2.3.0

Base URL

The following is the base URL for KMS REST API operations:

```
/vts/km/v1
```

Authentication

JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JavaScript Object Notation (JSON) object that is used as the payload of a JSON Web Signature (JWS) structure or as the plaintext of a JSON Web Encryption (JWE) structure, enabling the claims to be digitally signed or MACed and/or encrypted.

By default, JWT authentication expires in 10 minutes. This setting can be changed using the CLI, under the auth menu.

Retrieve a Token Request

Request:

```
curl -X POST \  
-H "Content-Type: application/json" \  
-d '{"username":"admin","password":"Password123!"}' \  
https://{server_address}/api/api-token-auth/
```

Response:

```
{ "token": "eyJ0eXA [...] KnA74" }
```

Refresh a Token Request

Request:

```
curl -X POST \  
-H "Content-Type: application/json" \  
-d '{"token":"<EXISTING_TOKEN>"}' \  
https://{server_address}/api/api-token-refresh/
```

Response:

```
{ "token": "eyJ0eXA [...] KnA74" }
```

Operations

Table 29: Paths

Path	Operations	Description
/keys	POST	Create or Import a key.
/keys/{id}	DELETE	Delete the path specified key.
	GET	Retrieve the path specified key.
	PATCH	Update the path specified key.
	POST	Export the path specified key.

Object Format

The response format for all requests is a JSON object.

API Reference

For the complete API reference, open the CTS GUI and navigate to:

Documentation > Key Management API

Response Codes

HTTP status codes are grouped into five categories:

- 1xx for informational
- 2xx for success
- 3xx for redirection
- 4xx for client errors
- 5xx for server errors

Admin REST API

The CTS Administration REST API allows you to manage the administration of the CTS. After completing the administrative tasks, you will be able to run the tokenize and detokenize data commands.

To use the Admin REST API you must be using:

- The Tokenization Server version 2.3.0 or newer
- Any keystore server version compatible with CTS 2.3.0

Base URL

The following is the base URL for Admin REST API operations:

```
https://<server_address>/api/
```

Authentication

JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JavaScript Object Notation (JSON) object that is used as the payload of a JSON Web Signature (JWS) structure or as the plaintext of a JSON Web Encryption (JWE) structure, enabling the claims to be digitally signed or MACed and/or encrypted.

By default, JWT authentication expires in 10 minutes. This setting can be changed using the CLI, under the auth menu.

Retrieve a token request

Request:

```
curl -X POST \  
-H "Content-Type: application/json" \  
-d '{"username":"admin","password":"Password123!"}' \  
https://{server_address}/api/api-token-auth/
```

Response:

```
{ "token": "eyJ0eXA [...] KnA74" }
```

Refresh a token request

Request:

```
curl -X POST \  
-H "Content-Type: application/json" \  
-d '{"token":"<EXISTING_TOKEN>"}' \  
https://{server_address}/api/api-token-refresh/
```

Response:

```
{ "token": "eyJ0eXA [...] KnA74" }
```

Responses

This section describes request response codes and response pagination details.

Response codes

HTTP status codes are grouped into five categories:

- 1xx for informational
- 2xx for success
- 3xx for redirection
- 4xx for client errors
- 5xx for server errors

Pagination

Requests that return multiple items will be paginated to 25 items.

Responses include a "count" field, which specifies the total number of objects available via pagination. Responses also include a "next" and "previous" field. If "count" is superior to 25, these fields contain a URL path to access the objects located respectively after or before.

Request:

```
GET https://{{server_address}}/api/v1/users/?limit=100&offset=400
```

Response:

```
HTTP 200 OK
{
  "count": 1023
  "next": "https://{{server_address}}/api/users/?limit=100&offset=500",
  "previous": "https://{{server_address}}/api/users/?limit=100&offset=300",
  "results": [ ... ]
}
```

Operations

The CTS Admin REST API supports the following operations:

Searching

To search objects, add the attribute `?search=` at the end of the URL followed by the pattern you want to look for. All the objects containing that pattern in one of their attributes will be included in the Response.

Request:

```
GET https://{{server_address}}/api/users/?search=robert
GET https://{{server_address}}/api/groups/?search=finance
```

Ordering

To query object in a specific order, add the attribute `?ordering=` at the end of the URL followed by the attribute you want to order by.

Ordering examples:

Order users by username

```
GET https://{server_address}/api/v1/users/?ordering=username
```

You may also specify reverse orderings by prefixing the field name with '-', like so:

```
GET https://{server_address}/api/v1/users/?ordering=-username
```

Multiple orderings may also be specified:

```
GET https://{server_address}/api/v1/users/?ordering=username,email
```

Object format

The response format for all requests is a JSON object.

Creating objects

To create a new object, send a POST request to the class URL containing the contents of the object. For example:

```
curl -X POST \
  -H '{"Authorization": "JWT '+<token>'}" \
  -H "Content-Type: application/json" \
  -d '{"name": "User1", ...}' \
  https://{server_address}/api/{some_resource}/
```

Retrieving objects

To retrieve a list of objects, send a GET request to the class URL. For example:

```
curl -X GET \
  -H '{"Authorization": "JWT '+<token>'}" \
  -H "Content-Type: application/json" \
  https://{server_address}/api/{some_resource}/
```

Updating objects

To edit an object, send a PUT request to the URL containing the new content with the object's ID. For example:

```
curl -X PUT \
  -H '{"Authorization':'JWT '+<token>}' \
  -H "Content-Type: application/json" \
  -d '{"name":"User1", ...}' \
  https://{server_address}/api/{some_resource}/:id
```

Deleting objects

To delete an object, send a DELETE request to the class URL with the object's ID. For example:

```
curl -X DELETE \
  -H '{"Authorization':'JWT '+<token>}' \
  -H "Content-Type: application/json" \
  https://{server_address}/api/{some_resource}/:id
```

Backing up CTS

Schedule automatic backups of the CTS.

```
GET https://<CTS_IP_Address>/api/backup/create
```

This request must be preceded by a successful authentication request. Use the secure token provided with the response in the *<auth token>* portion of this example..

Table 30: Set backup Endpoint Details

	Description
HTTP Request	https://{CTS_IP_Address}/api/backup/create
Method	GET
CTS IP Address	The IP address of the CTS to access with the API call.
Response Format	JSON
Required Authentication	Yes

Example cURL Requests

Authentication:

```
curl -X POST\
-H "Content-Type: application/json" \
-d '{username:"ctsuser","password:"Password123!"}' \
https://cts_server_address/api/api-token-auth
```

Backup request:

```
curl -X GET\
-H '{Authorization::"JWT <token>"}'\
-d '{username:"ctsuser","password:"Password123!"}' \
https://cts_server_address/api/backup/create -o backup.zip
```

Unauthenticated Healthcheck

Description

Use the healthcheck endpoint to get the status of the CTS. This endpoint integrates with load balancers to ensure all the daemons are running, and the tokenization server is healthy. An HTTP response code not equal to 200 indicates failure in CTS. The HTTP response also provides a display of status of all services, in JSON format.

Table 31: Healthcheck Endpoint Details

Data	Description
HTTP Request	https://CTS_IP_Address/healthcheck
Method	GET
VTS IP Address	The IP address of the VTS to access with the API call.
Response Format	JSON

The unauthenticated healthcheck response parameters and their respective types are described in the table shown below:

Table 32: Unauthenticated Healthcheck Response Parameters

	Type	Description
status	string	•200 (if request succeeded) •503 (if one of the daemons is not running)
	string	Configured services are listed along with each one's status (for example: {"name": "vaed", "status": "started"})

Healthcheck example

Example Request:

```
curl -k -X GET https://CTS_IP_Address/healthcheck
```

Example Response:

```
HTTP/1.1 200 OK
Date: Thu, 29 Oct 2020 22:51:16 GMT
Content-Type: application/json
Content-Length: 310
Connection: keep-alive
Server: VTS server
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Pragma: no-cache
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Strict-Transport-Security: max-age=31536000; includeSubdomains; preload
Content-Security-Policy: default-src 'self'

{
  "services": [
    {
      "name": "vtsauthz",
      "status": "Started"
    },
    {
      "name": "vaed",
      "status": "Started"
    },
    {
      "name": "uwsgi",
      "status": "Started"
    }
  ]
}
```

```
{
  {
    "name": "nginx",
    "status": "Started"
  },
  {
    "name": "kmd",
    "status": "Started"
  },
  {
    "name": "iam",
    "status": "Started"
  },
  {
    "name": "cryptod",
    "status": "Started"
  }
],
"status": "Started"
}
```

API Reference

For the complete API reference, open the CTS GUI and navigate to:

Documentation > Admin API

Importância da aplicação de medidas de segurança da informação para a garantia da proteção de dados pessoais



Documento elaborado por Peck+ Advogados para Thales

Contents

3 Segurança Da Informação Para Proteção Dos Dados Pessoais

3 Mas como adotar medidas técnicas e administrativas? Quais são essas medidas?

3 O que é ANONIMIZAÇÃO?

4 E por que ANONIMIZAR um dado pessoal?

4 Violação De Dado Pessoal

4 A Anonimização Como Ação Mitigadora De Riscos Aos Titulares

5 Tokenização Ciphertext

5 About Thales

Segurança da informação para proteção dos dados pessoais



Afinal, como aplicar boas práticas de Segurança da Informação quando do tratamento de dados pessoais?

A Lei Geral de Proteção de Dados (LGPD) veio estabelecer responsabilidades e direitos sobre os tratamentos de dados pessoais e a Segurança é um dos princípios que permeia tal atividades.

Nesse contexto, a citada legislação apresenta uma seção específica sobre Segurança e Sigilo dos dados e dispõe sobre a adoção de medidas técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito.

Mas como adotar medidas técnicas e administrativas? Quais são essas medidas?

Dentre tantas respostas para tais questões, a ANONIMIZAÇÃO é uma solução viável para aplicação da segurança dos dados de sua companhia.

O que é ANONIMIZAÇÃO?

O processo de anonimização, conforme apresentado no art. 5º, XI da Lei Geral de Proteção de Dados (LGPD), refere-se à “utilização de meios técnicos razoáveis e disponíveis no momento do tratamento, **por meio dos quais um dado perde a possibilidade de associação, direta ou indireta, a um indivíduo.**”.

A anonimização está relacionada ao uso de meios técnicos **razoáveis** para a transformação de um dado pessoal em dado anônimo, tornando impossível ou muito difícil a identificação de um indivíduo determinado por meio dos dados apresentados. O aspecto da razoabilidade da anonimização refere-se ao fato de que, para que certo dado seja considerado anonimizado, é necessário que, sem a realização de um esforço excessivo, não seja possível o reconhecimento de um indivíduo a partir da análise dos dados.

“ ANONIMIZAÇÃO, conforme a LGPD, é utilização de meios técnicos razoáveis e disponíveis no momento do tratamento, por meio dos quais um dado perde a possibilidade de associação, direta ou indireta, a um indivíduo.”



A Lei Brasileira inclusive traz também a definição do que é DADO ANONIMIZADO:

“...dado relativo a titular que não possa ser identificado, considerando a utilização de meios técnicos razoáveis e disponíveis na ocasião de seu tratamento.”

E por que ANONIMIZAR um dado pessoal?

Simple: a ANONIMIZAÇÃO AFASTA a aplicação da LGPD.

“Art. 1º Esta Lei dispõe sobre o tratamento de DADOS PESSOAIS, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural.”

Na maioria das leis que tratam sobre proteção de dados pessoais há sempre a menção acerca do processo de anonimização e de pseudonimização, sendo este último uma espécie de anonimização reversível dos dados, isto é, a partir deste processo ainda é possível a identificação posterior de indivíduos através da reassociação dos dados.

Na lei brasileira tais conceitos encontram-se no art. 5, XI (anonimização) e art. 13, § 4º (pseudonimização). À título de exemplo, ainda pode-se observar as descrições destes processos na lei de proteção de dados peruana, quando trata do procedimento de anonimización (processo irreversível de anonimização) e procedimento de disociación (processo reversível de anonimização); na lei argentina, que trata os procedimentos em apenas uma categoria chamada de disociación de datos; na lei mexicana, também sob uma única definição de disociación; entre outros.

É importante pontuar que para que os dados sejam considerados verdadeiramente anonimizados, é necessário que o processo impossibilite a reassociação dos dados posteriormente, de modo a tornar os titulares identificáveis. A fragilidade de grande parte das soluções de anonimização já é de conhecimento científico e prático, como observado abaixo:

Assim, o que se propõe a partir da utilização da solução CipherTrust Platform, é a garantia de um processo de anonimização eficaz e seguro, que garante a integridade dos dados tratados e o total anonimato dos titulares envolvidos na atividade.

“ Detentores de dados, incluindo agências governamentais, frequentemente removem todos os identificadores explícitos, tais como nome, endereço e número de telefone, de dados para que outras informações nos dados possam ser compartilhadas, de forma incorreta acreditando que as identidades dos indivíduos não podem ser determinadas. Ao contrário, a desidentificação dos dados não oferece garantia de anonimato.”²

Violação de dado pessoal

Considera-se como violação de dado pessoal todo e qualquer incidente de segurança que consiga atingir a integridade dos dados pessoais armazenados por determinado agente de tratamento, afetando a qualidade da informação e invadindo a privacidade dos titulares cujos dados foram atingidos.

Reforça-se que para que seja configurada uma violação de dado pessoal é necessário que os dados envolvidos no incidente sejam capazes de identificar ou de tornar identificável uma pessoa por meio de seu tratamento, portanto, sem a possibilidade de reconhecimento de indivíduos através dos dados, como no caso de dados anonimizados, não há a ocorrência de violação de dados pessoais, apenas configura-se um incidente de segurança na estrutura interna informacional da empresa.

Todas as empresas estão em maior ou menor medida expostas ao risco de sofrerem um ataque cibernético, o que pode ter repercussões reputacionais e atingir negativamente a continuidade dos negócios, sendo extremamente prejudicial às partes envolvidas, tanto aos titulares com dados vazados, quanto à empresa.

A anonimização como ação mitigadora de riscos aos titulares

A anonimização é uma ação necessária para todas as empresas que realmente prezam pela segurança de sua base de dados, até porque tal processo auxilia na melhor organização interna da base e também porque é a forma mais efetiva de garantir a segurança dos dados pessoais dos titulares.

Os riscos aos titulares são mitigados porque a utilização a anonimização de dados desvincula a informação do indivíduo, excluindo o tratamento do âmbito da LGPD, assegurando, portanto, um tratamento legítimo e sem afetar a privacidade.

¹ BIONI, Bruno Ricardo. Compreendendo o conceito de anonimização e dado anonimizado. In: ESCOLA PAULISTA DE MAGISTRATURA. **Cadernos Jurídicos: Direito Digital e Proteção de Dados Pessoais, vol. 1, n. 1.** São Paulo: Escola Paulista da Magistratura, 2000

² SAMARATI, Pierangela; SWEENEY, Latanya. Protecting Privacy when Disclosing Information: Anonymity and Its Enforcement through Generalization and Suppression. California: Technical Report SRI-CSL, 1998.

Tokenização Ciphertrust

A plataforma de segurança de dados CipherTrust possui como uma de suas maiores vantagens o processo de tokenização, oferecendo soluções que protegem e tornam anônimos os ativos confidenciais - quer eles residam no data center, em ambientes de big data ou na nuvem.

Os melhores atributos deste processo são: (i) por meio da utilização de tokenização e anonimização dos dados pela CipherTrust, afastar a necessidade de uma adequação mais robusta do banco de dados à normas de proteção de dados como a LGPD e a PCI DSS, o que diminui bastante os gastos de adequação da empresa; (ii) garantir uma melhor e mais eficiente proteção dos dados pessoais dos titulares cujos dados estão envolvidos na atividade de tratamento; (iii) e o fato de ser totalmente “tailor made” para a realidade da empresa, apresentando soluções que são flexíveis e eficazes para a segurança informacional da empresa e dos titulares.

Assim, convidamos a todos os interessados em terem um ambiente informacional mais seguro e atento às necessidades dos titulares, que testem a Plataforma de Segurança de Dados CipherTrust da Thales para criarem um futuro digital e responsável. A Thales pode ajudar a sua organização a cumprir com muitos dos requisitos da LGPD.

Documento elaborado por Peck+ Advogados para Thales

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.



Contact us

For all office locations and contact information,
please visit cpl.thalesgroup.com/contact-us

> cpl.thalesgroup.com <



CipherTrust Vaultless Tokenization with Dynamic Data Masking

Efficiently de-identifying sensitive data and reducing
PCI DSS audit scope



White Paper

Contents

3	Introduction
5	The Challenge
5	So many security demands, so little time
6	The Solution
6	Efficiently de-identify sensitive data
6	How vaultless tokenization works
7	Deliver Robust Data Security
7	Strong access controls
7	Flexible tokenization with dynamic data masking
8	Dynamic Data Masking of Credit Cards for Customer Service
9	Tokenization options
9	Deployment and administrative simplicity
10	Efficient scalability and agility
10	Key benefits
10	Sample implementations
11	About Thales

Introduction

Today's enterprise security teams have a lot on their plates and a lot on their minds. According to the Thales 2021 Global Data Threat Report, data breaches remain a widespread challenge. In the survey for the report, respondents indicated that 56% of organizations have ever experienced some form of data breach, with 41% of them having experienced a breach in the last 12 months. Breaches are a trailing indicator of security effectiveness. Turning to forward-looking metrics, 47% of respondents reported an increase in volume, severity and/or scope of cyberattacks!

But as you read this paper, consider that 46% of respondents said that they have avoided a breach notification because leaked data was encrypted or tokenized! Attacks are becoming more sophisticated and persistent while insider threats continue to be an Achilles heel. According to the 2017 Thales Data Threat Report¹, 88% feel at least somewhat vulnerable to data threats. The report also proves those concerns are well founded: 42% of organizations experienced a data breach at some point.

Tokenization is a critical tool for enterprises to meet compliance standards, and reduce the risk of loss posed by this increasingly hostile environment. Most often, tokenization has been a great way for enterprises to reduce the cost of their Payment Card Industry Data Security Standard (PCI DSS) audits. Tokens are used to replace credit card numbers, removing the need for many servers to undergo expensive yearly audits. But use cases for tokenization are expanding today to safeguard more types of data and in more environments as the locations where this data is used proliferate.

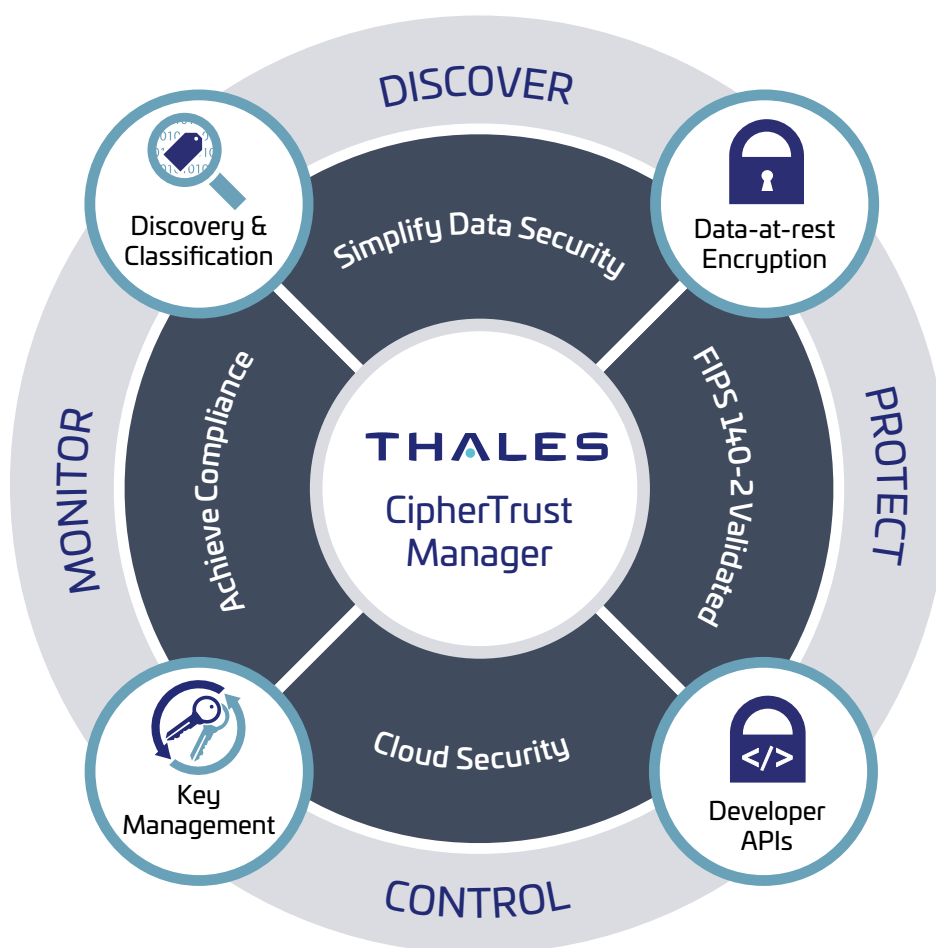
Driven by data privacy and sovereignty regulations around the world, the data types that need to be protected are no longer just credit cards, but also include U.S. Social Security numbers, Drivers Licenses, U.K. National Insurance numbers, Passport numbers, and National Identity numbers in many countries around the world among other data. And the need to use this sensitive data in advanced technology environments like cloud and big data leads to further complications.

In this white paper we'll first look at how CipherTrust Vaultless Tokenization works, key characteristics of the solution, and then examine sample implementations including PCI DSS compliance, customer contact centers, cloud, big data and more.

As noted earlier, tokenization is especially useful in PCI DSS compliance. In the past meeting requirements of the standard has been very complex and expensive, with extensive yearly audits of servers hosting credit card data required, or solutions that were limited in bandwidth and required extensive infrastructure to support.

CipherTrust Tokenization now offers a better way to meet these requirements. The solution's vaultless tokenization capabilities enable security teams to remove databases from PCI DSS scope, and so significantly reduce compliance cost and efforts. Implementations typically result in minimal disruption and administrative overhead and result in breakthroughs in operational efficiency.

CipherTrust Vaultless Tokenization leverages CipherTrust Manager as its key source and, of course, is a component of the CipherTrust Data Security Platform, which delivers a comprehensive range of data security and compliance controls that help IT teams efficiently address their requirements for encryption, privileged user access control, file access auditing, and more. With a single infrastructure, policy and encryption key management environment, the platform reduces the overall costs for organizations to implement and use data security tools while applying strong protections for data.



The Challenge

So many security demands, so little time

For today's security teams, it seems virtually everything is proliferating:

- **Threats.**

Risks posed by nation-states, cyber criminals, and malicious insiders continue to grow more sophisticated and more persistent

- **Data and repositories.**

The volume and diversity of assets and repositories that need to be protected continues to see explosive growth. Instead of just managing the traditional data center, which was challenging enough, there are now virtualized systems, big data environments, outsourced services, private clouds, public clouds, and hybrid models to contend with

- **Mandates and tools.**

As the number of repositories continues to grow, so do the number of systems that fall within the scope of PCI DSS and other regulations. Further, the scope and complexity of these regulatory mandates also continues to expand. As a result, security teams are compelled to deploy more safeguards and tools in order to establish the controls these mandates require. In particular, security teams face increased pressures to prepare for PCI DSS 3.0 audits, and adapt their security approaches as cardholder data moves into new environments—such as cloud and big data implementations—and is shared with outsourced service providers, such as analytics, development, and engineering firms

All this proliferation continues to place increasing demands on security teams—but these teams don't see their time, staffing, or budgets undergoing any commensurate expansion. To contend with these realities, many security professionals have explored the use of tokenization, which has the potential to provide a convenient way to protect sensitive assets. At a high level, tokenization is the process of replacing a sensitive record with a token, effectively a meaningless placeholder for the data that can't be exploited or reverted back to its original form

While tokenization offers the potential to address a wide range of security and compliance objectives, traditional tokenization tools and solutions have been far too costly and complex to procure and manage. These tools typically require security teams to invest in expensive appliances and contend with software agents that are difficult to deploy and manage. More than ever, security teams need to be able to leverage the benefits of tokenization—and they need to do so in a scalable, efficient, and cost effective manner.

The Solution

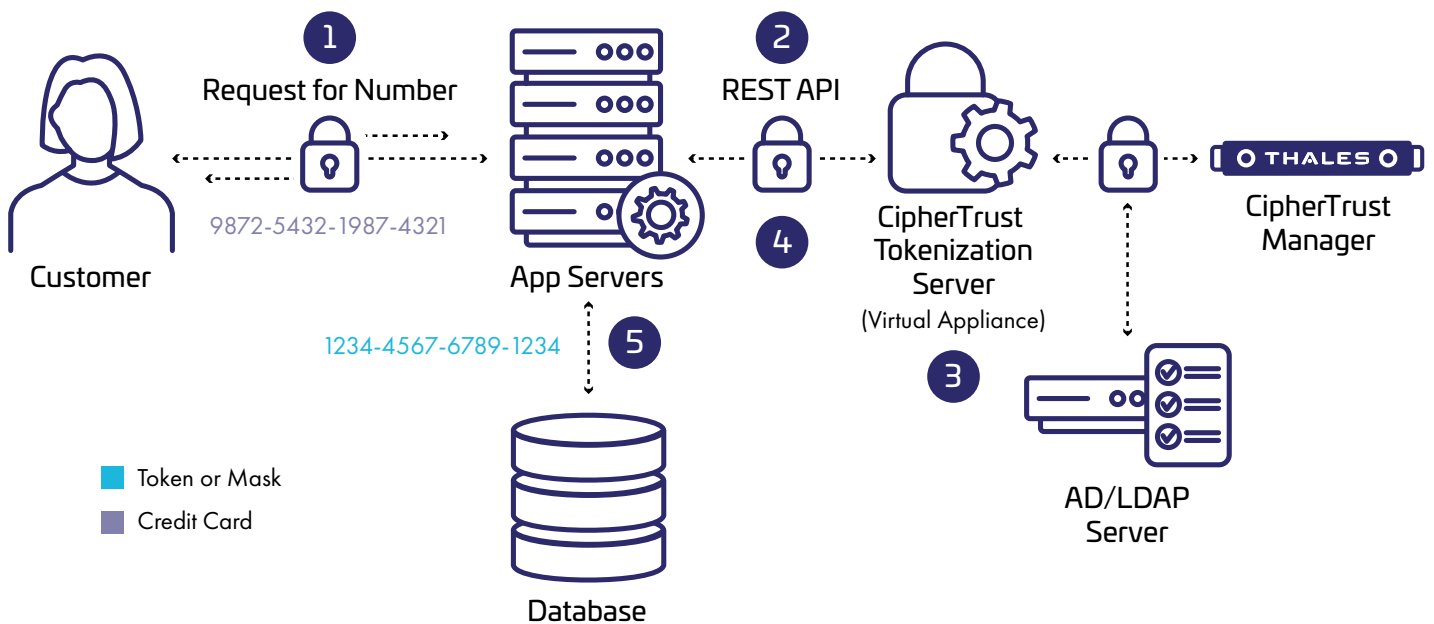
Efficiently de-identify sensitive data

The CipherTrust Data Security Platform offers tokenization capabilities that can dramatically reduce the costs and effort associated with complying with security policies and PCI DSS. With CipherTrust Vaultless Tokenization with Dynamic Data Masking, organizations can efficiently address objectives for securing sensitive assets and cardholder records— whether they reside in the data center, big data environments, or the cloud.

How vaultless tokenization works

For illustration purposes, we'll outline the process of tokenizing a credit card number. The CipherTrust Tokenization Server acts as a central hub for tokenizing records and managing access to tokens and clear-text data.

1. The customer submits a credit card number into an application.
2. The application submits the credit card number to the Tokenization Server via the REST API.
3. The Tokenization Server holds in memory a secure table created based on an AES-256 encryption key seed provided by the CipherTrust Manager. This table holds with random matches for credit card numbers – the token matching that card is identified.
4. Through the REST API, the Tokenization Server returns the token to the application.
5. The application server then submits the tokenized value into the database.



Deliver Robust Data Security

Strong access controls

Thales recommends creating at least two user accounts, or user groups from LDAP, be available on to the Tokenization Server:

- The first account or LDAP group with the permission to tokenize but not to detokenize, and the second account or LDAP group with the permission to detokenize
- The user account with tokenization privileges can then be utilized by multiple servers, including a public-facing web server, for example. This allows data to be tokenized immediately as it is brought in through applications
- Accounts or LDAP groups with permission to have detokenization privileges should be limited to a specific server that needs these permissions. For example, detokenization privileges could be restricted to a server that's responsible for submitting credit card numbers to a remote card processing service
- Any internal LDAP groups that have rights to detokenize data (Application administrators, for instance) should be strictly limited to the minimum number of members required

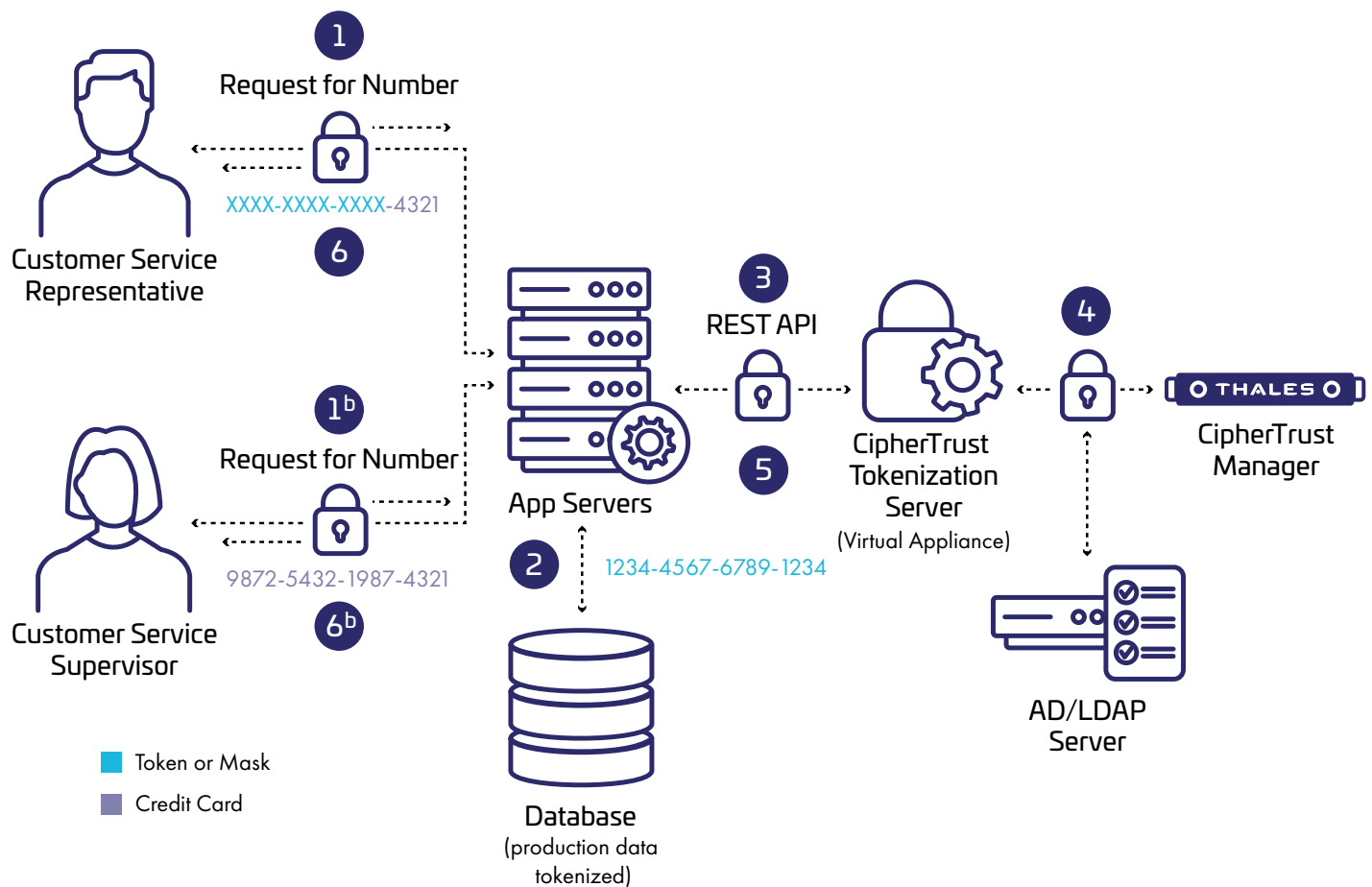
Tokenized data can be used with databases or files needed for development, QA and Test, or deployed to Big Data environments without risk to sensitive information. For credit card data, servers using tokenized information are removed from expensive and time consuming yearly audits as well. Further, Thales offers a high availability, standardsbased, FIPS 140-2 and Common Criteria validated policy management platform with the CipherTrust Manager that can provide maximum security for cryptographic keys and policies used with tokenization.

Flexible tokenization with dynamic data masking

CipherTrust Vaultless Tokenization also offers the flexibility to establish varying levels of data redaction. Security administrators can establish settings to have an entire field tokenized or to dynamically mask only part of a field, keeping the remainder in the original clear-text. When combined with LDAP-based integration to Active Directory and other directory services solutions, this capability enables uses such as call centers. One example of call center usage is limiting credit card number visibility for customer service representatives to the last four digits of a customer's credit card, while customer service supervisors can see the entire number.

The process of setting these policies is done using an intuitive administrative interface. Through the interface, administrators can choose exactly how they want data to be redacted, including specifying whether to replace a value with such characters as a pound sign, asterisk, or hyphen.

Dynamic Data Masking of Credit Cards for Customer Service



With CipherTrust Vaultless Tokenization, policies could be implemented so that a customer service supervisor could see credit card numbers in the clear, while a customer service representative would only see the last four digits.

Initial conditions:

- Credit card data in a product database is tokenized
- A policy is set for the customer services representatives group in LDAP that allows them to see only the last four digits of a credit card number
- A second policy allows members of the customer services supervisor group to see the entire number

Workflow

1. Customer service representative request to see a credit card number
2. The application calls the token from the production databases
3. The application server then passes the token and User ID information to the Tokenization Server using a RESTful API
4. The Tokenization Server calls directory services requesting group information for the User ID
5. Based on the User ID, customer services group membership, and the policy that customer services representatives can only view the last four numbers of a credit card, the token is processed, and the credit card number retrieved. Data is then dynamically masked, with the card number set to XXXX-XXXX-XXXX-4321, where 4321 are the last four digits of the actual card. XXXX-XXXX-XXXX-4321 is then returned to the application server
6. The application server displays XXXX-XXXX-XXXX-4321 to the customer services representative

Tokenization options

As well as offering the capability to dynamically mask data for display, CipherTrust Tokenization offers options for tokenizing a variety of fixed length data types – including Numeric, Text and Date formats. Options for data types to be tokenized include specific modes for:

- Numbers – Best for credit cards, U.S. Social Security numbers and other purely numeric data
- Text – All letters, all printable ASCII, All ASCII or AlphaNumeric for use with applications such as U.K. National Insurance numbers, drivers license numbers and others that include text only or mixed text and numbers
- Partial tokenization – Select only a portion of the fixed length field to be tokenized – keeping a portion of the data as the original information
- One-time use tokens – Used only when the data will never be de-tokenized
- Dates – including formats with various separators. Can also be modified with a start and end year
- Prefix – Prefixes can be appended to all tokens – for instance to designate a drivers license (DL)

Deployment and administrative simplicity

With CipherTrust Vaultless Tokenization, you can capitalize on a range of features that streamline upfront implementation and ongoing operations. CipherTrust Vaultless Tokenization

- **Enables efficient, enterprise-wide administration**

With the capabilities offered by the CipherTrust Data Security Platform, you can choose from a range of technologies and employ the mix that's optimally suited to an organizations specific projects and use cases. At the same time, enterprises gain the cost savings and operational benefits of working with solutions that can be centrally and uniformly managed. With the CipherTrust Data Security Platform from Thales, organizations can centrally manage keys for Vaultless Tokenization, as well as for CipherTrust Transparent Encryption, CipherTrust Application Data Protection, other CipherTrust products, and third-party devices.

- **Offers non-disruptive implementation**

With the solution's capabilities, you can restrict access to sensitive assets, yet at the same time, format the protected data in a way that reduces the operational impact typically associated with encryption and other obfuscation techniques. For example, organizations can tokenize a credit card field in a database, yet keep the tokenized information in a format that is compatible with associated applications. Further, tokens can appear to be real credit card numbers and pass LUHN validation, so tokenization does not break existing validation processes.

- **Eliminates manual efforts and complexity**

Tokenization occurs at the application layer, and it streamlines all the application development efforts associated with implementing tokenization in an enterprise. With the solution, developers don't have to manually institute identity management or redaction policies. CipherTrust Vaultless Tokenization offers an easy-to-use REST API for integration with the CipherTrust Tokenization Server, so that application developers can simply and quickly add tokenization and dynamic data masking to applications.

- **Provides investment protection**

Customers that have already invested in the CipherTrust Data Security Platform can get even more out of their investments. By leveraging existing CipherTrust Manager implementations (required for all CipherTrust Data Security Platform products), it is quick and cost effective to add Vaultless Tokenization through an all-software deployment.

Efficient scalability and agility

CipherTrust Vaultless Tokenization delivers the performance needed to address the operational demands of the most processing-intensive environments, enabling organizations to do millions of tokenization or detokenization operations per second. The Tokenization Server runs on virtual machines and can be quickly and efficiently scaled up and scaled down to accommodate changing workloads. For redundancy, Thales recommends deploying a cluster with at least two nodes. A three-node or four-node cluster multiplies the throughput, while providing added redundancy. Performance realized will vary depending on hardware platforms, cluster sizes, and client parallelism employed.

Key benefits

- **Reduce PCI DSS compliance effort and scope**

By leveraging CipherTrust Vaultless Tokenization, you can minimize the repositories and processes that can gain access to payment data in the clear, organizations can significantly reduce PCI DSS compliance costs and efforts.

- **Fully leverage cloud, big data, and outsourced models**

CipherTrust Vaultless Tokenization enables organizations to more fully leverage cloud services, big data models, and outsourced environments, while retaining the security controls required. For example, tokenized data can be easily migrated tokenized data to a public cloud, without exposing sensitive data to unauthorized access, even access by administrators in the cloud provider's environment.

- **Establish broad safeguards around sensitive assets**

Unlike other dynamic data masking tools, the CipherTrust solution tokenizes sensitive fields in the production database. As a result, enterprises can establish comprehensive safeguards around sensitive assets across the organization. Because the source and display data remain protected, this includes strong safeguards against cyberattacks from criminals and nation-states, as well as from insider abuse.

- **Minimize data security overhead and training**

With the CipherTrust Data Security Platform, enterprises can leverage a platform to centrally manage keys for Vaultless Tokenization and as well for the full range of the platform's encryption and access controls solutions. By leveraging a central platform that offers comprehensive capabilities, organizations can minimize the cost, effort, and training associated with managing point-products.

Sample implementations

PCI DSS 3.X Compliance

The more locations that hold primary account numbers (PANs), the more locations that fall under PCI DSS regulations. As the number of locations grow, so too do the cost and effort associated with establishing and sustaining compliance, preparing for audits, and so on. With CipherTrust Tokenization, you can tokenize PANs, and significantly restrict the number of users, processes, and repositories that can get access to this data in the clear. As a result, organizations can dramatically reduce compliance efforts and costs. Further, through such capabilities as Luhn check support, Vaultless Tokenization enables organizations to implement tokenization with minimal impact on applications and processes. In addition, the CipherTrust Data Security Platform offers many additional controls you need to address many PCI DSS requirements, including least privileged access controls, encryption, key management, and file access audit logs.

Customer contact centers

Vaultless Tokenization helps mitigate the threats that can be posed by support staff exploiting their access privileges to steal or leak sensitive data. Dynamic Data Masking can be effective both for securing sensitive data in internal contact centers and in outsourced or geographically distributed support environments. For example, enabling a support agent to see only the last 4 digits of a Social Security Number. As a result, agents can conduct any required customer verification processes, without being put in a position where they can maliciously or inadvertently leak sensitive and regulated PII.

Outsourced Development

For many organizations, application development is done by an increasingly dispersed set of teams, including geographically distributed employees as well as external contractors and service providers. While these distributed teams have helped improve the cost efficiency and agility of development efforts, they can pose significant challenges for those organizations that manage sensitive and regulated data. From initial development to testing and staging, developers need access to data repositories. If all data is left in the clear, it exposes a range of sensitive regulated data and intellectual property to any number of threats. With CipherTrust Vaultless Tokenization, organizations can quickly and efficiently provide internal and external development teams with access to corporate data repositories, while tokenizing the specific assets that pose security or compliance risks.

A complementary product, CipherTrust Batch Data Transformation, also helps organization to quickly tokenize large data sets for this purpose when needed.

Big Data

To capitalize on the opportunities presented by big data, organizations need to aggregate information from a broad range of repositories—and in the process can create massive collections of highly sensitive and proprietary data. The CipherTrust Data Security Platform enables organizations to fully leverage big data, without introducing risk to the business. Sensitive records can be tokenized before they are incorporated into big data repositories, so big data scientists, developers, and business analysts don't gain access to sensitive data that is not required for their work. At the same time, the solution's flexible tokenization options ensure that tokenized records don't disrupt big data migration, calculation, and analytics processes.

Cloud Services

When your organization moves to cloud computing models, whether infrastructure as a service (IaaS), software as a service (SaaS), platform as a service (PaaS), or hybrid approaches, you may offload a number of responsibilities and efforts, but your organization ultimately still bears responsibility for the security of its sensitive and regulated data. With CipherTrust Vaultless Tokenization, your organization can leverage the cloud models it needs, while managing tokenization policies locally. As a result, you can retain complete control and visibility over who can access sensitive assets in the clear, no matter where tokenized information ultimately resides.

Static Data Masking

Organizations can replace sensitive assets with tokens, so that outsourced application development and testing teams can run QA and analytics using realistic looking data, without having access to any sensitive assets. Once sensitive values in the database are tokenized, enterprises simply need to create a copy of the production database and give that copy to the outsourced development team. Since these teams don't have access to both the CipherTrust Tokenization Server and user credentials to detokenize, they won't be able to access sensitive data in the clear. This protects the business, maintains compliance, and enables outsource development teams to access and test realistic production data.

Just as noted above under Outsourced Development, CipherTrust Batch Data Transformation helps organization to quickly tokenize large data sets for this purpose when needed.

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.



Contact us

For all office locations and contact information,
please visit cpl.thalesgroup.com/contact-us

> cpl.thalesgroup.com <



CipherTrust Application Data Protection

Developer-Friendly APIs for Data Protection

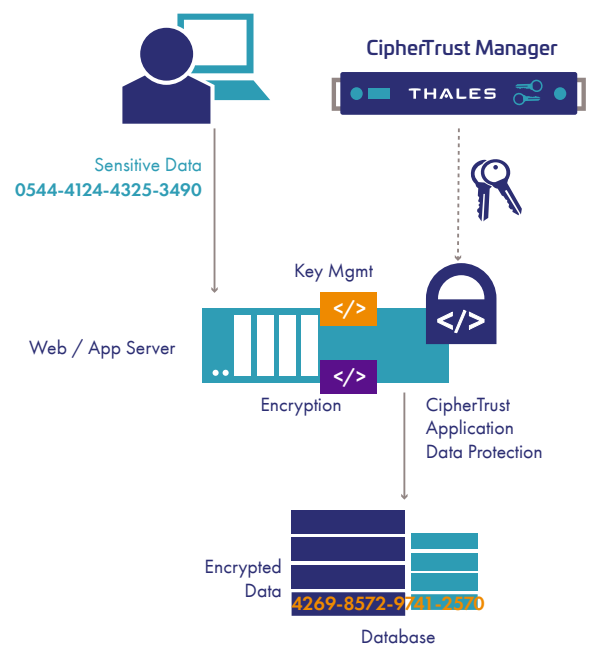


CipherTrust Application Data Protection offers developer-friendly software tools for encryption key management as well as application-level encryption of sensitive data. Protecting data at the application layer can provide the highest level of security, as it can take place immediately upon data creation or first processing and can remain encrypted regardless of its data life cycle state – during transfer, use, backup or copy. The solution is flexible enough to encrypt any type of data passing through an application. CipherTrust Application Data Protection can be deployed on premises or in private or public cloud infrastructure to secure data even when it is migrating from one environment to another, without any modifications to existing encryption or data processing policies.

Architectural Overview

CipherTrust Application Data Protection harmonizes interactions between developers and IT operations who share a common goal of data security. Developers enjoy language bindings appropriate to their projects. Operations can leverage choices among Crypto Service Providers that run on a wide range of operating systems. Further, the product includes many operational features that enhance performance and availability to ensure that security imposes a minimal to zero impact on business operations.

CipherTrust Application Data Protection operates with CipherTrust Manager, providing an architecture that centralizes encryption keys for applications. For enhanced separation of duties, CipherTrust Manager defines granular controls on both key users and key operational use.



One of several architectural choices for CipherTrust Application Data Protection

Developer-Friendly Data Protection APIs

Adding data protection to the software development process can be optimized by saving software engineers from having to master cryptography, deal with the challenge of finding high quality encryption keys as well as a safe store for them and mastering complex APIs without a comprehensive collection of sample code. CipherTrust Application Data Protection from Thales offers well-documented Crypto APIs with a wide range of language bindings, integrated with a range of industry-standard Crypto Service Providers that enable fast development of data protection for integration into mission-critical applications.

Development Libraries and APIs

Supported CipherTrust Application Data Protection language bindings and APIs include Java, C, and C# for .NET Core and .NET; XML Open Interface; RESTful APIs; KMIP client

Core Cryptographic Functions

Like a hardware security module (HSM) such as Thales Luna, CipherTrust Application Data Protection core cryptographic functions include, among many others including encrypt/decrypt, Sign, Hash (SHA) and HMAC

The solution also offers support for versioned keys for encryption and decryption, signing and HMAC. Versioned keys can reduce the risk of data loss in the event of a compromised key.

Crypto Service Providers and Supported Operating Systems

In the following table, • indicates that a CipherTrust Application Data Protection crypto service provider binary is available for the operating system.

Abbreviations key:

- CAPI = C (language) API
- CSP = Crypto Service Provider (Windows)
- CNG = Crypto Next Generation [Provider]
- JCE = Java Crypto Engine

	CAPI	PKCS#11	CSP/CNG	JCE
Linux	•	•		•
Windows Server	•	•	•	•
AIX	•	•		•
HP-UX				•
Solaris		•		•
MacOS	•			

Choosing where to encrypt

Crypto operations on the application server: For many uses cases, and, in particular for the most latency-sensitive workloads CipherTrust Application Data Protection libraries on the application server include all crypto functions. Keys are cached for use on the application server. Keys are encrypted when not in use and obfuscated in memory when in use.

Crypto operations retained on CipherTrust Manager:

CipherTrust Application Data Protection offers the possibility of retaining all encryption keys on CipherTrust Manager and forwarding encryption and decryption requests there. Retaining keys on CipherTrust Manager offers the highest key security. To support performance and availability for crypto operations on CipherTrust Manager, CipherTrust Application Data Protection libraries on the application server offer the following tools:

- Connection pooling
- Single- and multi-tiered load balancing

Crypto Service Aggregation Point

Applications can also choose to make RESTful API crypto requests to the CipherTrust Application Data Protection Web Services. This Web Services option can perform crypto operations locally or act as a centralized forwarder to CipherTrust Manager. Web Services option offers the same key protections for local operations as CipherTrust Application Data Protection libraries.

Rich Crypto Ecosystem

CipherTrust Application Data Protection has integrations for Microsoft SQL Server Always Encrypted, Microsoft Online Certificate Status Protocol (OCSP), Hashi Vault, HortonWorks, Apache HTTP and NGINX Servers, Lieberman ERPM, and many others.

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.



CipherTrust Vaultless Tokenization (CT-VL)

Version: v2.6.8

30 January 2026

Contents

Administration	4
Quick Start	7
Alternative Deployments	15
CT-VL Deployment Prerequisites	16
Installing CT-VL as a Virtual Machine on VMware vSphere	18
Installing CT-VL on Bare Metal	18
Installing CT-VL on AWS Cloud	19
Installing CT-VL on Azure Cloud	21
Installing CT-VL on Azure Stack Hub	23
Installing CT-VL on Hyper-V	24
Installing CT-VL on KVM	25
Installing CT-VL on Google Cloud Platform	26
Installing CT-VL on Alibaba Cloud	27
Tasks	31
Configuring CT-VL through the Command Line	32
Registering CT-VL VM with CipherTrust Manager	38
Configuring CT-VL Nodes	44
Configuring an AD/LDAP Server	49
Configuring ADFS and LDAP for OAuth	53
Configuring LDAP Settings.ini File	55
Working with CT-VL GUI	57
Dashboard API Monitoring	70
Administrator History Logs	71
Backing Up and Restoring CT-VL	73
Upgrading CT-VL	76
Monitoring Through SNMP	78
Examples	80
Cryptographic Operation Examples	81
Key Management Examples	87
Versioned Key Management Examples	96
Tokenization Examples	98
Advanced Topics	103
CT-VL CLI Reference	103
CT-VL Services	109
CT-VL Key Cache	110
Key Types and Input Sizes	111
Creating Static and Dynamic Keys	112
Tokenization Groups, Templates, and Masks	113
Understanding CT-VL Authentication and Authorization	116
CT-VL CLI Navigation Summary and Guidelines	121
Tokenization Setup Workflow in CT-VL GUI	122
CT-VL Tokenization REST APIs	123

CT-VL Admin REST APIs	132
CT-VL with Multiple Key Managers	138
Troubleshooting	138

Administration

Overview

CipherTrust Vaultless Tokenization (CT-VL) is a platform-independent appliance (virtual machine or bare-metal) that offers REST-API services to protect sensitive data. CT-VL offers three service categories for protecting sensitive data:

- **Tokenization Service:** Tokenization is used for replacing sensitive data with tokenized, or encrypted, data. Tokenized data retains the format of the original data while protecting it from theft or compromise. For example, if a company accepts customer credit card numbers, those numbers can be tokenized and stored as random-looking numbers, then detokenized by an employee or application with the proper permissions.

Tokenization is frequently used for sensitive data such as credit card numbers, social security numbers, drivers licenses, or other personally identifiable information (PII). Data masking can be applied to any detokenized data to hide sections of the data from different groups of users. For example, less-privileged database users might view only the last four digits of a detokenized credit card number, while a more privileged user could view the entire card number.

CT-VL uses the CipherTrust Manager (CM) to store the AES keys to tokenize and detokenize sensitive data.

- **Cryptographic Services:** CT-VL can be used to protect sensitive data using encryption. It can be used to encrypt, decrypt, sign, and verify data which can be tied to specific keys and user permissions. CT-VL uses the CipherTrust Manager to store the keys for encryption.
- **Key Management Services:** CT-VL can be used to manage keys stored on the CipherTrust Manager. It can create, modify, search, import, export, and many other operations, keys on the CM.

All the services above are available as REST-APIs which can be used to integrate into the end-user's application.

To achieve high performance and reliability (in the form of redundancy), multiple CT-VL appliances (virtual machines) can be configured to form a cluster. The cluster can operate as one coherent CT-VL system that could be served by a load balancer.

CT-VL can run as a virtual machine and is supported on many hypervisors and cloud platforms, namely, VMware vSphere, Kernel-based Virtual Machine (KVM), HyperV, Amazon Web Services (AWS), Microsoft Azure Marketplace, and Google Cloud Platform (GCP).

CT-VL can also run on a physical machine (bare-metal). The software product is available in ISO format installable on bare metal machines. All settings applicable to a virtual machine are also applicable to the physical machine.

Note

CipherTrust Vaultless Tokenization (CT-VL), formerly known as Vormetric Tokenization Server (VTS) and CipherTrust Tokenization Server (CTS), has been renamed to conform with several other former Thales Vormetric products adopting the brand name CipherTrust.

Interfaces

CT-VL has three forms of user-interface:

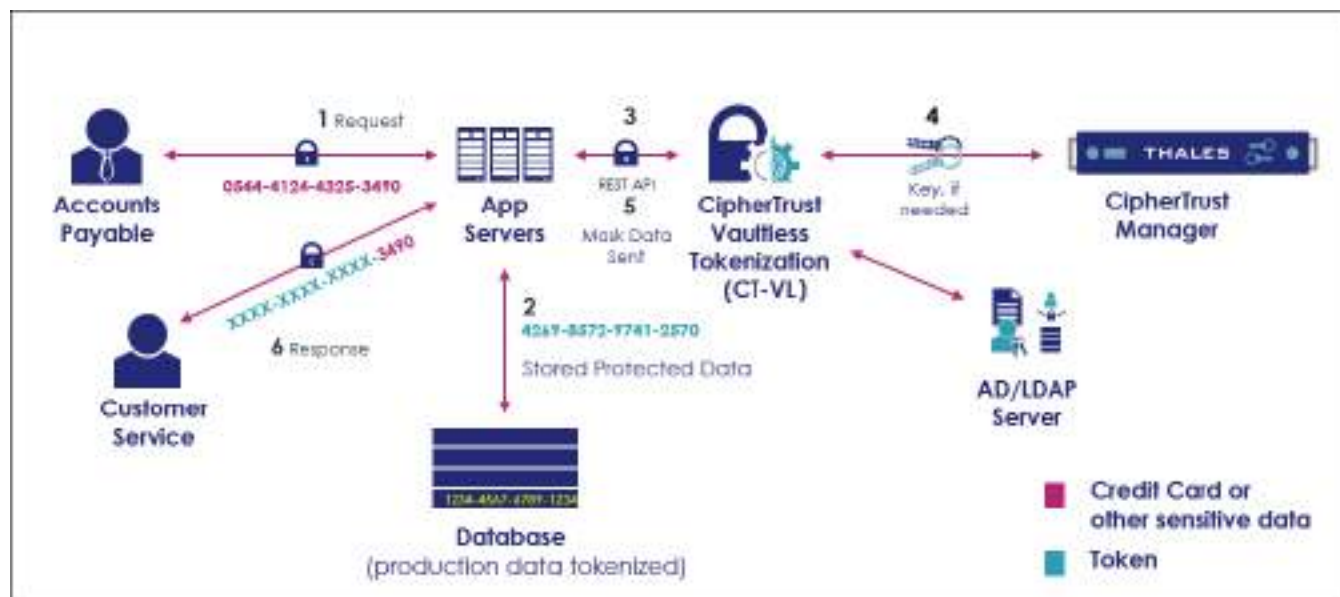
- **Command-Line Interface (CLI):** This interface is available using the SSH protocol. It is used by the administrators to install, configure, and maintain the CT-VL virtual machine and cluster.
- **Web User Interface:** This interface is used by the administrators to manage users, groups, keys, policies (templates), permissions, data masking, and other administrative functions used for tokenization.
- **REST API:** This interface is used for tokenization, cryptographic, and key management operations. Administrative operations are also available using REST APIs. These APIs are also documented in YAML files located under Documentation in the CT-VL Administration WebUI.

Component Architecture

Multiple CT-VL appliances can be set up to form a cluster to achieve high performance and availability. CT-VL in a cluster uses bi-directional replication to synchronize data across all virtual machines. In addition, CT-VL supports connection to multiple CipherTrust Managers in a cluster to achieve high availability for key access.

- A minimum of two CT-VL nodes in a cluster is recommended for redundancy.
- A maximum of 16 nodes are supported.
- A load balancer can be used to achieve higher performance by using multiple virtual machines and automatically distributing the load across all nodes in the cluster.

Below is the CT-VL architecture diagram:



Authentication and Authorization

CT-VL requires a CipherTrust Manager to obtain the keys to tokenize and encrypt sensitive data. Keys used are

not persistent and never stored on the CT-VL virtual machine. It only caches keys in memory for a brief period to achieve high performance.

In addition to locally created users and groups, CT-VL can also use a Windows AD or LDAP server to authenticate users performing tokenization, encryption, and key management. They can be used for identity management to map users and groups for tokenization policies (templates), encryption policies, and key management access.

Moreover, CT-VL supports client-certificate identity and authentication.

Refer to [Understanding CT-VL Authentication and Authorization](#) for further information.

Supported Tokenization Algorithms

CT-VL supports the following tokenization algorithms:

- FF3
- FPE-luhn
- Dates
- Random
- Random-luhn
- FF1
- FF1-luhn

Supported Character Sets

CT-VL supports the following character sets for tokenization:

- Numeric
- Alphanumeric
- Printable ASCII
- Custom character sets

Supported Cryptographic Operations

CT-VL supports the following cryptographic operations:

- Encrypt
- Decrypt
- Sign

-
- Verify

Supported Tokenization Operations

CT-VL supports the following tokenization operations:

- Tokenize
- Detokenize

Supported Key Operations

CT-VL supports the following key operations:

- Create
- Import
- Destroy
- Modify
- Find
- Export

Quick Start

This quick start guide describes the concise steps to set up and run a CT-VL appliance to perform tokenization, encryption, and key management. This guide focuses on using a virtual machine hosted on VMware vSphere ESXi. For other options to run a CT-VL appliance, see [Alternative Deployments](#).

Prerequisites for Installing CT-VL

Prerequisites for installing CT-VL as a VM are listed below:

Note

For the complete list of prerequisites for all CT-VL deployment options, see [CT-VL Deployment Prerequisites](#).

- CipherTrust Manager 2.11.1 LTS or higher.
- VMWare vSphere ESXi Server 6.0 or higher.
- CT-VL OVA image version 2.6.0 or higher.

- For optimal performance, the recommended settings for a CT-VL VM are 4 CPUs and 16GB memory.

Installing CT-VL

Steps for installing CT-VL as a VM are listed below:

1. Create a CT-VL virtual machine
2. Log in to the VM
3. Set up the VM network
4. Register VM with the Key Manager
5. Initialize the CT-VL cluster

1. Creating a CT-VL Virtual Machine

Obtain a CT-VL ISO or OVA image file, for example - `cts-2.6.0.205.ova` and deploy it to create a virtual machine. Refer to the *VMware ESX User Guide* on *how to create a virtual machine using an ISO or OVA image*. It is recommended to use Thick Provisioning for hard disks.

For a multi-node setup, you can repeat this installation process for additional machines. You would then configure each node separately and join them in a cluster. See [Configuring CT-VL Nodes](#) for details.

2. Initial Log In

Power up the CT-VL virtual machine that you have created. Open the virtual machine console and log in using the following credentials:

Note

CT-VL supports serial console for GCP and Azure Cloud platforms. You may also perform the initial login using SSH if networking is already available.

```
login: cliadmin
password: cliadmin123
```

This initial log-in will prompt you to change the password of the `cliadmin` user.

3. Network Setup

Networking for CT-VL can either use DHCP or the static IP configuration.

DHCP Network Setup

DHCP Network Setup

The CT-VL virtual machine supports DHCP to set up networking. It is enabled by default and if your network has a DHCP server, the CT-VL VM will use it automatically to obtain an IP address and configure its network at power-up.

To determine the IP address obtained by the VM from DHCP, log in to the VM console. Note: The CT-VL runs the Open VM Tools software. With this, you view the IP address used by the VM from the vSphere ESXi host or vCenter.

Static Network Setup

Static (non-DHCP) Network Setup

If your network does not have DHCP server or if DHCP is not desired, you can set up networking using a static IP address.

To do so, perform the following steps:

1. Log in to the console of the VM and run the command below to set up networking.

```
main> network setup eth0
```

The CLI command above will run interactively.

2. When prompted, supply the network configuration, that is, IP address, netmask, and gateway.
3. A non-interactive command is also available to set up the network. Supply the -h option to display usage.

```
main> network set -h
usage: set [-h] [--onboot {yes,no}] [--dhcp] [--ipaddr IPADDR]
          [--netmask NETMASK] [--gateway GATEWAY] [--defroute {yes,no}]
          [--mtu MTU] [-y]
          device
```

4. Run the 'network dns' command to configure DNS. Supply the -h option to display usage.

```
main> network dns -h
usage: dns [-h] [--show] [--add IP [IP ...]] [--remove IP [IP ...]]
          [--useDHCP {yes,no}] [-y]
```

4. Register the CT-VL VM with CipherTrust Manager (CM)

You can register the CT-VL VM with CipherTrust Manager (CM) as the key manager. For complete set of steps, see [Registering CT-VL VM with CipherTrust Manager](#).

Perform the following steps to register the CT-VL VM with CipherTrust Manager:

Log in to the CM server

1. Create an NAE interface.
2. Choose a port number for the NAE interface and set the mode to "TLS, verify client cert, user must supply password."
3. For the CipherTrust Manager 2.11 and above, enable **Allow unregistered clients**.
4. Set a local CA for the NAE interface.
5. Create a CM user for CT-VL use.
6. Create AES key(s) for CT-VL use. Preferably, use 256 bits for key size.
7. Set the AES key(s) to be exportable.
8. Set the AES key(s) to be accessible by the CM user created in #4 above.

Log in to the CT-VL VM

1. Run the CLI command below to configure the CM server.

```
icapi setup
```

This command runs an interactive setup to configure connectivity to the CM server. Follow the prompts to supply the IP address of the CM, port number of the NAE interface, CM user, and password. The CM user supplied must have access to the keys that will be used by CT-VL.

2. Run the CLI command below to register the CT-VL VM to the CM server.

```
icapi register --host <ipaddress> --user <admin>
```

where:

ipaddress - IP address of the CM server.

admin - name of a user on the CM with admin access to sign certificates.

3. Enable CM for CT-VL use.

```
icapi enable yes
```

4. Test connectivity to the CM server.

Run the CLI command below to test if CT-VL can connect and authenticate to the CM server.

```
icapi test --server_connection
```

5. Test access to a key on the CM.

Run the CLI command below to test if CT-VL has access to a key.

```
icapi test --key keyname
```

5. Initialize the CT-VL Cluster

Even if you have only one CT-VL server, you need to create and initialize a cluster. Run the command below to initialize the CT-VL cluster.

```
cluster create
```

This command is interactive. You will be prompted to supply a username, password, and email. This user will be created on the CT-VL VM and will serve as the initial administrative user to the WebUI.

Configuring CT-VL

Complete the following steps to configure CT-VL:

I. Configure CT-VL Tokenization API

Browse to the hostname or IP address `https://<ct-vl_ipaddress>/` of the VM.

Log in to the CT-VL Web user interface using the username and password created during cluster initialization.

II. Create Users

1. Navigate to the Users menu.
2. Select the 'New User' button and specify the appropriate details to create users. At a minimum, users created must have 'Active' permission to allow access to tokenization, encryption, and key management REST-APIs. To allow access to the WebUI, users must have 'Staff' or 'Superuser' permissions. Setting users with 'Staff' permission will allow WebUI access but read-only. Setting users with 'Superuser' permission will allow WebUI access with read-write.

III. Create Keys

1. Navigate to the Keys menu.
2. Select the 'Symmetric Keys' tab.

3. Create symmetric keys you will use for tokenization, encryption, or key management. The name of the keys supplied must match the name of keys on the Key Manager.

IV. Create Token Groups

1. Navigate to the **Tokenization > Token Groups** Menu.
2. Create a token group and assign a key that it will use.

V. Create Token Templates

1. Navigate to the 'Tokenization Templates' Menu.
2. Create a Tokenization Template and set a name and token group for it.
3. Select a Format algorithm, e.g., FPE.
4. Select a Character Set, e.g., All digits.

VI. Set Key Permissions

1. Navigate to the 'Permissions' Menu.
2. Select a user and key.
3. Select the permissions you will require for the user. For tokenization operations, select 'Tokenize' and/or 'Detokenize'. For cryptographic operations, options are: Encrypt, Decrypt, Sign, and Verify. For key management operations, options are: Destroy, Modify, Export, and Find.

Note

To know more about CT-VL token groups, token templates, and data masks, see the [Tokenization Groups, Templates, and Masks](#).

Run a sample tokenization, detokenization, cryptographic operation.

Complete the required operations:

- [Tokenize Data](#)
- [Detokenize Data](#)
- [Encrypt Data](#)
- [Decrypt Data](#)

Tokenize Plaintext Data

Use a REST-API client to tokenize a plaintext numeric data.

Request POST Parameters

Parameter	Type	Description
data	string	Plaintext data to be tokenized.
tokengroup	string	Name of the token group.
tokentemplate	string	Name of the token template.

Request

```
curl -k -u vtsuser -X POST 'https://ipaddress/vts/rest/v2.0/tokenize' \
-H 'content-type: application/json' \
-d '{"tokengroup": "TG1", "tokentemplate": "TT1", "data": "1111222233334444"}'
```

Response

```
{
  "token": "6300847582017130",
  "status": "Succeed"
}
```

Detokenize Tokenized Data

Use a REST-API client to detokenize a tokenized data.

Request POST Parameters

Parameter	Type	Description
token	string	Encrypted data to be detokenized.
tokengroup	string	Name of the token group.
tokentemplate	string	Name of the token template.

Request

```
curl -k -u vtsuser -X POST 'https://ipaddress/vts/rest/v2.0/detokenize' \
-H 'content-type: application/json' \
-d '{"tokengroup": "TG1", "tokentemplate": "TT1", "token": "6300847582017130"}'
```


Response

```
{
  "data": "1111222233334444",
  "status": "Succeed"
}
```

Encrypt Plaintext Data

Use a REST-API client to encrypt a plaintext data.

Request POST Parameters

Parameter	Type	Description
plaintext	string	Plaintext data to be encrypted. This needs to be encoded in base64 format.
alg	string	Encryption algorithm.
kid	string	Name of key to use.
iv	string	Initialization vector string encoded in base64 format.

Sample POST request data for encryption:

```
{
  "plaintext": "9mBirbK+VZLR/x6gDlGioQ==",
  "alg": "A256CBCPAD",
  "kid": "key1",
  "params": {
    "iv": "aKcjACEIDh0x0AX0SCQoqQ=="
  }
}
```

Request

```
curl -k -u vtsuser -X POST 'https://ipaddress/vts/crypto/v1/encrypt' \
-H 'content-type: application/json' \
--data @encrypt_data.yaml
```

Response

```
{
  "ciphertext": "WPn2Py8oC7kM1MRTtf4xYxho+UnbVYmTKmKibC9X2+o=",
  "tag": ""
}
```

Decrypt Ciphertext Data

Use a REST-API client to decrypt a ciphertext data.

Request POST Parameters

Parameter	Type	Description
ciphertext	string	Ciphertext data to be decrypted. This needs to be encoded in base64 format.
alg	string	Encryption algorithm.
kid	string	Name of key to use.
iv	string	Initialization vector string encoded in base64 format.

Sample POST request data for decryption:

```
{
  "ciphertext": "WPn2Py8oC7kM1MRTtf4xYxho+UnbVYmTKmKibC9X2+o=",
  "alg": "A256CBCPAD",
  "kid": "key1",
  "params": {
    "iv": "aKcjACEIDh0x0AX0SCQoqQ=="
  }
}
```

Request

```
curl -k -u vtsuser -X POST 'https://ipaddress/vts/crypto/v1/decrypt' \
-H 'content-type: application/json' \
--data @decrypt_data.yaml
```

Response

```
{
  "plaintext": "9mBirbK+VZLR/x6gDlGioQ=="
}
```

Alternative Deployments

CipherTrust Vaultless Tokenization can be deployed in the following ways:

Note

- For AWS and Google Cloud Platforms, contact Thales Customer Support to obtain access to CT-VL images. Refer to the cloud provider user guide to know how to create a virtual machine on the platform.
- For Microsoft Azure Cloud platform, go to Azure Marketplace and look for the latest version of CT-VL. For other platforms, download the CT-VL image from Thales Customer Support Portal.

- Installing CT-VL as a Virtual Machine on VMware vSphere
- Installing CT-VL on Bare Metal
- Installing CT-VL on AWS Cloud
- Installing CT-VL on Azure Cloud
- Installing CT-VL on Azure Stack Hub
- Installing CT-VL on Hyper-V
- Installing CT-VL on KVM
- Installing CT-VL on Google Cloud Platform
- Installing CT-VL on Alibaba Cloud

For the list of all prerequisites, see [CT-VL Deployment Prerequisites](#).

CT-VL Deployment Prerequisites

All Prerequisites

- CipherTrust Manager 2.11.1 LTS or higher.
- VMWare vSphere ESXi Server 6.0 or higher; Hyper-V; AWS, Azure, Google, and KVM cloud Platforms.
- Intel x86_64 physical machine (bare metal) can also be used in lieu of a virtual machine. Where virtual machines are mentioned in this document, it is also applicable to physical machines.
- CT-VL ISO, OVA, or cloud images with version 2.5 or higher. For physical machines, the disk size must be at least 85GB.
- For optimal performance, the recommended settings for a CT-VL VM are 4 CPUs and 16GB memory.

Active Directory/LDAP Server Requirements

An optional AD/LDAP server can be used to generate users and authenticate their credentials when they request services.

See [Understanding CT-VL Authentication and Authorization](#) for a conceptual basis on how AD/LDAP is used.

Requirements:

- Access to the AD/LDAP server containing users and groups that will have CT-VL privileges.
- For LDAPS, the LDAP server's CA certificate must be imported into the Tokenization Server.

Load Balancer Requirements

A cluster of CT-VL nodes can be installed and run as one coherent CT-VL system using a load balancer. The load balancer acts as the main entry point for tokenization and crypto services to distribute load across all nodes in the cluster. It provides scalability and reliability to the system.

CT-VL supports the following methods of load balancing:

- **Round-robin:** Requests are distributed in a round-robin fashion.
- **Least-connected:** Allows the load balancer to control the load on some CT-VL nodes.
- **IP-hash:** The client IP address is used to determine what server in a server group should be selected for the client request. This can be used for client's session sticky or persistent.

The load balancer must be able to use TLS1.1 and higher.

Port Configuration

If CT-VL must communicate with other components through a firewall, open the ports in the firewall as mentioned in the following tables.

Below table describes each port you must open for incoming communication to CT-VL:

Port No.	Protocol	Communication	Description
22	TCP	Management Console → CT-VL	Administration CLI SSH access.
443	TCP	Browser → CT-VL Requester → CT-VL CLI → CT-VL	HTTPS access for Administration GUI and REST/JSON API.
5432	TCP	CT-VL ↔ CT-VL	This port is required if a cluster of more than one node is used. This is used exclusively for communication between the CT-VL nodes in a cluster.

Below table describes each CT-VL port you must open for outgoing communication from CT-VL:

Port No.	Protocol	Communication	Description
User-defined	TCP	CTS → Log Server	Remote logging.
123	UDP	CT-VL ↔ NTP Server	Optional network time synchronization.
389	TCP	CT-VL → LDAP Server	Optional LDAP connection.
636	TCP	CT-VL → LDAPS Server	Optional LDAPS (LDAP over TLS) connection.

Installing CT-VL as a Virtual Machine on VMware vSphere

Prerequisites

- VMware ESXi hypervisor, ESXi version 6.0 or higher.
- Minimum recommended settings for the CT-VL VM: 4 CPUs, 16GB memory.

Installation

CT-VL can use a VMware ESXi hypervisor to create a virtual machine using the ISO or OVA build image. CT-VL supports ESXi version 6.0 or higher. Refer to VMware ESXi documentation on how to create a VM using an ISO or OVA image.

It is recommended to use Thick Provisioning for hard disks.

For a multi-node setup, you can repeat this installation process for additional machines. You would then configure each node separately and join them in a cluster as described in [Configuring CT-VL Nodes](#).

Configuration

To begin configuration, click the **Console** tab of the vSphere Client to log in to the CTS CLI as cliadmin, change the default password, accept the Service Level Agreement, and begin configuring CTS settings using the CTS CLI.

See [Configuring CT-VL](#) for further information.

Installing CT-VL on Bare Metal

Although typically deployed as a virtual machine, the CipherTrust Vaultless Tokenization (CT-VL) server can be installed using an ISO image and run on a bare metal machine.

A bare metal host is a computer that does not run hypervisor software. Do not install the operating system on the host before installing CT-VL, because the OS is included along with the CT-VL software in the installation package. When selecting the operating system, specify 64-bit CENTOS version 7 or higher.

Prerequisites for Installing CT-VL on Bare Metal

- Intel x86_64 physical machine.
- Number of CPU cores: 4.
- RAM: 16 GB minimum if RANDOM mode is none. 24 GB recommended if CT-VL will be used to tokenize credit card numbers longer than 16 digits. 24 GB if using numeric RANDOM mode or alphanumeric RANDOM mode, or 32 GB if using both modes.
- Hard disk space: 85 GB.

Installation Steps

To do a bare metal installation:

1. Obtain the installation file `ctvl-<release>.iso`.
2. Refer to the machine's documentation on how to boot and install from an ISO image.

Installing CT-VL on AWS Cloud

Note

Contact Thales Customer Support to obtain access to the CT-VL AMI image in Amazon Web Services Cloud.

Installation Prerequisites

Following components are necessary for running a CT-VL in the cloud on Amazon Web Services (AWS):

- **Minimum recommended settings for CT-VL VM:** 4 CPUs, 16GB memory.
- **A VPC (Amazon's Virtual Private Cloud):** If you plan to create a cluster of CT-VL machines, it is recommended that your cluster nodes use a VPC so that they communicate only with each other within the private cloud. Cluster communication does not need to be, and should not be, exposed on the Internet.
- **A Subnet:** If you choose to use a VPC, you will need to create a subnet and associate that subnet to that VPC. This subnet is where your CT-VL cluster nodes will get the IP addresses.
- **A Security Group:** Set your security group with the following minimum settings:
 - Incoming port **443** - Web and REST API access.
 - Incoming port **22** - CLI access.

-
- Incoming port **5432** - Optional. Set this only if the machines in your cluster are not using a VPC to communicate with each other.

If you will be accessing your CT-VL strictly inside the VPC, you will still need a security group but you can block all incoming ports.

- For best security, use SSH key pairs instead of passwords for SSH logins. If you choose to use an EC2 key pair when creating a CT-VL instance, the cliadmin will use that key (a private key) to log in by SSH. Password access will be disabled.
- If you choose not to use an EC2 key pair, the cliadmin will use a password to log in by SSH.
- Upgrading an EC2 instance from CT-VL v2.5 or earlier retains the use of passwords for the cliadmin. Only newly-created instances from CT-VL v2.6 and above can use the SSH key pair in AWS.

Create the CT-VL Instance

Follow this procedure to create the CT-VL instance:

1. Choose your AWS region.
2. Locate the AMI of the CT-VL image.
3. Select the AMI image and launch an instance.
4. Select Instance type: **m4.xlarge** - with recommended minimums of 4 vCPUs and 16GB memory.
5. Select the VPC to use for the installation.
6. Select the subnet.
7. Select Auto-assign public IP, if applicable.
8. Set the primary IP address. This will be the IP address of the VPC subnet, and it will be used by the VM for its cluster node IP address.
9. In the Add Storage menu, select **delete on termination**, if applicable. Do not change the size of the root volume.
10. Choose whether to have **cliadmin** log in via SSH with a password or with an EC2 key pair.
 - If you choose not to use an EC2 key pair, the **cliadmin** will use a password to log in by SSH.
 - If you choose an EC2 key pair, the **cliadmin** will use that key (private key) to log in by SSH. Password access will be disabled. You are prompted to select an existing key pair.
Select the private key file (in PEM format), and make sure to confirm your selection before launching the new instance.

Note

If an instance is configured to use an EC2 key pair, password access is blocked, and all password-related CLI commands are disabled.

Configuration Notes

To begin configuration, `ssh` into the instance as the `cliadmin` user using the public IP address or the VPC IP address, and follow the steps in [Configuring CT-VL](#).

If you will be creating a cluster of CT-VL machines, create an instance for each node in the cluster. You would configure each node separately and join them in a cluster as described in [Configuring CT-VL Nodes](#).

Note

When configuring a multi-node cluster, you must use the VPC IP address for the `node_IP_address`.

Installing CT-VL on Azure Cloud

Note

To obtain the CT-VL image for Azure, go to Azure MarketPlace, and look for "CipherTrust Vaultless Tokenization" v2.6 or higher.

Prerequisites to Install CT-VL on Azure Cloud

- A virtual machine and a Subnet.
- Minimum recommended settings for a CT-VL VM: 4 CPUs, 16GB memory.

Note

- The network security group defines the firewall rules of the CT-VL virtual machine. At a minimum, incoming ports must allow port 22 (SSH) and port 443 (HTTPS).
- If your cluster nodes are located in different resource groups, you also need to add port 5432 to the allowed incoming ports. However, if all your cluster nodes are in the same resource group, do not add

port 5432, because nodes inside a resource group do not need any firewall permissions to communicate with one another.

Create the Azure VM

1. Log in to the [Azure web portal](#).
2. Go to the *Marketplace* screen and search for **CipherTrust Vaultless Tokenization**. Select the desired version, and click the **Create** button.
3. Supply the following information to create a virtual machine:
 - **VM name:** name of the VM you want to use, for example, `my-CT-VL-2.3.0-azure-vm`.
 - **VM disk type:** SSD or HDD.
 - **User name:** Not required by CT-VL, but Azure needs this data to create a VM. This user name entry is not retained by CT-VL and is not used as a CT-VL credential.
 - **Authentication type:** Also not required by CT-VL, but required for Azure to create a VM. This authentication type entry is not retained by CT-VL and does affect the CT-VL configuration.
 - **Resource group:** You may create a new resource group or use an existing one. If you plan to create a cluster of CT-VL machines, it is recommended that all cluster nodes use the same resource group, so that they can communicate with one another without having to use a public static IP address or enable an incoming port in the firewall.
 - **Location:** For example, "West-US".
 - **Subnet and subnet address range:**
 - **Static or dynamic public IP address:** Use a static IP address if you expect your cluster to span multiple resource groups.

Configuration Notes

To begin configuration, `ssh` into the instance as the `cliadmin` user using the public IP address, and follow the steps in [Configuring the CT-VL System](#).

Note that on Azure, you will not configure the CT-VL network settings. Azure automatically configures these parameters for the CT-VL VM:

- Internal and subnet IP addresses are provided to the VM through DHCP.
- For CT-VL clusters, create an instance for each node in the cluster. Then configure each node separately, and join the nodes in a cluster.

- When creating or joining a cluster, if you have all the cluster nodes in the resource group, you must use the same internal or subnet IP address used by the VM.
- When adding a node to the cluster, you must always use the internal or subnet IP address of the new CT-VL VM node.

Installing CT-VL on Azure Stack Hub

Note

To obtain the CT-VL image for Azure Stack Hub, download the CT-VL VHD image from Thales Customer Support Portal.

Prerequisites to Install CT-VL on Azure Stack Hub

- A virtual machine and a subnet.
- Minimum recommended settings for a CT-VL VM: 4 CPUs, 16GB memory.

Note

- The network security group defines the firewall rules of the CT-VL virtual machine. At a minimum, incoming ports must allow port 22 (SSH) and port 443 (HTTPS).
- If your cluster nodes are located in different resource groups, you also need to add port 5432 to the allowed incoming ports. However, if all your cluster nodes are in the same resource group, do not add port 5432, because nodes inside a resource group do not need any firewall permissions to communicate with one another.

Create the Azure VM

1. Log in to Azure Stack Hub.
2. Upload the CT-VL VHD image into the Azure Stack Hub storage container. The CT-VL VHD image is of dynamic format. To upload a VHD image to Azure Stack Hub, the VHD image must be in fixed format. Convert the VHD image to fixed format before uploading to the Azure Stack Hub storage container. If you use the 'azure-vhd-utils' tool, you can use the dynamic VHD image to upload and it will convert automatically to fixed format when it saves to the Azure Stack Hub.
3. Navigate to Virtual Machines menu. Supply the following information to create a virtual machine:

-
- **VM name:** name of the VM you want to use, for example, `my-CT-VL-2.3.0-azure-vm`.
 - **VM disk type:** SSD or HDD.
 - **User name:** Not required by CT-VL, but Azure needs this data to create a VM. This user name entry is not retained by CT-VL and is not used as a CT-VL credential.
 - **Authentication type:** Also not required by CT-VL, but required for Azure to create a VM. This authentication type entry is not retained by CT-VL and does affect the CT-VL configuration.
 - **Resource group:** You may create a new resource group or use an existing one. If you plan to create a cluster of CT-VL machines, it is recommended that all cluster nodes use the same resource group, so that they can communicate with one another without having to use a public static IP address or enable an incoming port in the firewall.
 - **Location:** For example, "West-US".
 - **Subnet and subnet address range:**
 - **Static or dynamic public IP address:** Use a static IP address if you expect your cluster to span multiple resource groups.

Configuration Notes

To begin configuration, `ssh` into the instance as the `cliadmin` user using the public IP address, and follow the steps in [Configuring the CT-VL System](#).

Note that on Azure, you will not configure the CT-VL network settings. Azure automatically configures these parameters for the CT-VL VM:

- Internal and subnet IP addresses are provided to the VM through DHCP.
- For CT-VL clusters, create an instance for each node in the cluster. Then configure each node separately, and join the nodes in a cluster.
- When creating or joining a cluster, if you have all the cluster nodes in the resource group, you must use the same internal or subnet IP address used by the VM.
- When adding a node to the cluster, you must always use the internal or subnet IP address of the new CT-VL VM node.

Installing CT-VL on Hyper-V

Prerequisites for Installing CT-VL on Hyper-V

- The `.iso` image or `VHD` file.
- Minimum recommended settings for the CT-VL VM: 4 CPUs, 16GB memory.

Installation Steps

Note

- Install the CT-VL instance on a Hyper-V server using an ISO or VHD image file.
- Download the `ctvl-<release>.iso` or `ctvl-<release>.vhd` file to the Hyper-V server.

Use the following steps to install the CT-VL on a Hyper-V:

1. Open Hyper-V Manager.
2. Select **Action** > **New** > **Virtual Machine**.
3. Choose **Generation 1**.
4. Set the start up memory to 16GB.
5. Choose a network adapter.
6. For the VM, choose one of the following based on the type of image file to be used:
 - **VHD image:** Use an existing hard disk: Select CT-VL VHD image file you have downloaded or copied.
 - **ISO image:** Install an operating system from a bootable CD/DVD-ROM. Navigate to and select the downloaded ISO file.
7. Select **Finish** to create the Virtual Machine.
8. For ISO installation: Select and start the Virtual Machine. The VM will boot the mounted ISO image file and install the CT-VL.

Installing CT-VL on KVM

Prerequisites to Install CT-VL on KVM

Minimum recommended settings for the CT-VL VM: 4 CPUs, 16GB memory.

Installation Steps

Below are the steps to install CT-VL on a Linux Kernel-based Virtual Machine (KVM):

1. Download the CT-VL image in `qcow2` format. Place this image file into your KVM images directory.
2. Create a KVM virtual machine by importing the CT-VL `qcow2` image. Use the following KVM parameters:

- **virt-type:** kvm
- **os-type:** linux
- **os-variant:** centos7.0
- **vcpus:** 4
- **ram:** 16000
- **controller:** type=scsi,model=virtio-scsi
- **disk:** /var/lib/libvirt/images/cts-2.x.n.nnn.qcow2,bus=scsi

3. Connect to the console, and log in as user `cliadmin`.

Installing CT-VL on Google Cloud Platform

Note

Contact Thales Customer Support to obtain access to the CT-VL disk image in Google Cloud Platform.

Prerequisites for Installing CT-VL on Google Cloud Platform

Machine configuration recommended settings:

- General-purpose
- N1 Series
- 4 CPUs, 16GB memory

Installation Steps

To install the CT-VL on Google Cloud Platform (GCP) using the image:

1. Contact Thales Customer Support, and provide your GCP account and region details for access to the CT-VL disk image.

Note

CT-VL is also available as a tar image file. Thales Customer Support places the image Tar file in the `gs://cts-release` support folder. Refer to the cloud provider user guide to know how to convert the tar file

to a disk image and create a virtual machine on the platform.

2. Log in to the GCP console and navigate to **Compute Engine > VM Instances > Create Instance**.
3. Set a name, region, and zone for your instance.
4. Select your machine configuration as per the prerequisites.
5. Select **Boot disk**, and navigate to **Change > Custom Images**.
6. In the Show Images tab, look for the project name that Thales Support has provided and select that project.
7. Select the Image tab and look for the latest CT-VL GA release and select it. For example, ctvl-240-216.
8. Choose a boot disk type, for example, Standard or SSD.

Note

Do not change the disk size.

9. Close the Boot disk page by clicking Select.
10. In the Firewall setting, select **Allow HTTPS traffic**.

Note

Do not use SSH keys.

11. Click **Create** to begin creating the CT-VL instance.
12. Connect to the instance using SSH to begin the CT-VL setup.

Installing CT-VL on Alibaba Cloud

Installation Prerequisites

Following are the prerequisites for installing CT-VL on Alibaba Cloud:

- Minimum recommended settings for CT-VL instance are 4 CPUs and 16GB memory.

-
- If you want to create a cluster of CT-VL machines, it is recommended that cluster nodes should use a Virtual Private Cloud (VPC) so that they communicate only with each other within the private cloud. Cluster communication does not need to be exposed to the Internet.
 - If you choose to use a VPC, you will need to create a subnet and associate that subnet to that VPC. This subnet is where your CT-VL cluster nodes will get the IP addresses.
 - Set your security group with the following minimum settings:
 - Incoming port 443: Web and REST API access.
 - Incoming port 22: CLI access.
 - Incoming port 5432: Optional. Set this only if the nodes in CT-VL cluster are not using a VPC to communicate with each other.

Optional: Create your own CT-VL Alibaba Image

You may create your own CT-VL Alibaba image and use that to create a CT-VL instance (VM). To create your own CT-VL image, upload **CT-VL QCOW2** image file to your OSS storage, and then launch **Import Image** task.

CT-VL Alibaba image can be created:

- [Using WebUI](#)
- [Using CLI](#)

Using WebUI

Following are the steps to create CT-VL Alibaba image using WebUI:

1. Log in to [OSS Console](#).
2. Create a bucket repository if you have not done so.
3. Navigate to your OSS bucket.
4. Select **Upload** to upload **CT-VL QCOW2** image file.
5. Log in to [ECS Console](#) and navigate to **Instances > Images**.
6. Select **Import Image Manually**.
7. Supply the following image information:
 - OSS Object Address: File URL from OSS storage.
 - Image Name: ctvl-2.6.7.21.
 - Operating System: Linux.

- System Disk Size: 85GB.
- System Architecture: x86_64.
- Platform: Other Linux.
- Image Format: QCOW2.
- License Type: BYOL.
- Image Description: CipherTrust Vaultless Tokenization.

Note

- Do not select platform other than "**Other Linux**".
- Do not change the System Disk Size (85GB).

Using CLI

Following are the steps to create CT-VL Alibaba image using CLI:

1. Upload CT-VL QCOW2 image file to Alibaba Object Storage.

Example:

```
$ ossutil64 cp ctvl-2.6.7.21.qcow2 oss://ct-vl/builds/
```

2. Import image.

Example:

```
$ aliyun ecs ImportImage \  
--RegionId us-west-1 \  
--Description "CipherTrust Vaultless Tokenization" \  
--ImageName "ctvl-2.6.7.21" \  
--LicenseType BYOL \  
--DiskDeviceMapping.1.Format QCOW2 \  
--DiskDeviceMapping.1.DiskImageSize 85 \  
--DiskDeviceMapping.1.OSSObject "builds/ctvl-2.6.7.21.qcow2" \  
--DiskDeviceMapping.1.OSSBucket "ct-vl" \  
--Architecture x86_64 \  
--OSType Linux \  
--Platform 'Others Linux'
```


Note

Do not change the System Disk Size (85GB).

Sample Output Response:

```
{
  "ImageId": "m-rj9dwoc45p5gxvyfw812",
  "RegionId": "us-west-1",
  "RequestId": "9EFC096A-AB3F-3373-AC67-91C19915C003",
  "TaskId": "t-rj9dwoc45p5gxvybupew"
}
```

Create a CT-VL Instance (VM)

Following are the steps to create a CT-VL instance (VM) using a CT-VL image shared by Thales Customer Service Support or your own imported image from **CT-VL QCOW2** image file:

1. Log in to Alibaba Cloud and go to Elastic Compute Service.
2. Select a Region. For example: Singapore, US (Silicon Valley).
3. Select Create Instance.
4. Select a Billing Method. For example: Subscription or Pay-As-You-Go.
5. Select Zone. For example: Zone A or Zone B.
6. Select an Instance Type:
 - Architecture: x86-Architecture.
 - Category: General Purpose or Compute Optimized.
 - Instance Type:
 - Minimum: 4 CPUs, 16GB Memory.
 - Recommended: 8 CPUs, 32GB Memory.
7. Select Quantity.
 - Minimum: Two instances for a 2-node cluster.
8. Select Image.

-
- For image you created from **CT - VL QCOW2** image file , select **Custom**.
 - For image shared by Thales Customer Service Support, select **Shared Image** and look for CT-VL.

9. Select Storage.

- Enhanced SSD or Standard SSD.
- Use 85GB as shown (DO NOT CHANGE).
- Optional: Release with Instance or Disk Encryption.

10. Select Network Type: VPC.

11. Optional: Select Public IP Address: Assign Public IPv4 Address.

12. Select Security Group. Security group must allow SSH and HTTPS access.

13. Select Logon Credentials: Set Later.

Note

Key Pair is currently not supported. If you select Key Pair, it will be ignored.

14. Provide Instance Name.

15. Select Create Instance.

Tasks

This section contains the following articles:

- [Configuring CT-VL through the Command Line](#)
- [Registering CT-VL VM with Key Manager](#)
- [Configuring CT-VL Nodes](#)
- [Configuring an AD/LDAP Server](#)
- [Configuring ADFS and LDAP for OAuth](#)
- [Configuring LDAP Settings.ini File](#)
- [Working with CT-VL GUI](#)
- [Dashboard API Monitoring](#)

-
- Administrator History Logs
 - Backing Up and Restoring CT-VL
 - Upgrading CT-VL
 - Monitoring Through SNMP

Configuring CT-VL through the Command Line

This topic describes how the network administrator uses the CT-VL Command Line Interface (CLI) to configure the CT-VL environment, establish connections between the various components, and other system-level tasks.

Accessing the CLI

The network administrator uses the CT-VL CLI to configure the CT-VL environment as described below:

1. Access the CLI through a terminal emulator like PuTTY.
2. On first access, enter the default login and password:

login: `cliadmin`

Password: `cliadmin123`

3. On first access, you will be prompted to change the password. **Save and store your new password securely.**
4. On first access, you will be prompted to review and sign the License Agreement.

Refer to [CT-VL CLI Navigation Summary and Guidelines](#) for description about CT-VL CLI commands.

Basic Configuration Steps

Note

- The configuration steps in the following sections employ a subset of the CLI commands. See [CT-VL CLI Reference](#) for a summary of CLI commands. For complete usage details on any CLI command, run the command with the `--help` option.
- For multi-node installations, the basic configuration steps in this section must be completed on each node before you [Configure CT-VL Nodes](#).

Configure the CT-VL Network Settings

Configure the network settings for VMware and bare metal installations with a static IP implementation.

Note

Network settings are required only for VMware and bare metal installations, with a static IP implementation. Do not attempt to set the IP address and gateway of the network interface if you are using CTS in an Azure, Amazon, or Google Cloud Platform. Cloud providers assign the IP addresses to the VM through DHCP.

To configure the CT-VL network settings, complete the following tasks:

- Set the IP address for the CT-VL VM.
- Configure the CT-VL DNS server or use the `/etc/hosts` file.

Check [Deployment Prerequisites](#) for the IP address, gateway address, and DNS values you will need.

Set the IP Address for the CT-VL VM

1. Using the CT-VL CLI, navigate to the *network* commands menu.
2. There are two ways to set the IP address for the CT-VL VM: Interactive (with the `setup` command) or Non-interactive (with the `set` command).

Interactive

To set the IP address using the interactive mode, use the `setup` command followed by the network device name:

```
network> setup eth0
```

Using the `setup` command prompts you for the IP address, as well as other information required to set up the specified network device, such as subnet mask, gateway, and so on.

Non-interactive

To set up the IP address with the non-interactive mode, use the `set` command with the `--ipaddr IPADDR` option and argument value. With the same `set` command invocation, you can also configure other network and device parameters with additional options:

```
ct-vl> network set --help
usage: set [-h] [--onboot {yes,no}] [--dhcp] [--ipaddr IPADDR]
[--netmask NETMASK] [--gateway GATEWAY] [--defroute {yes,no}]
[--mtu MTU] [-y]
device
Configure Network Interface
[. . .]
```

Options and Arguments to the Non-interactive `set` Command:

Option	Argument	Description
<code>--onboot</code>	yes	no
<code>--dhcp</code>		Use DHCP
<code>--ipaddr</code>	IPADDR	Set the IPv4 address of this device to IPADDR
<code>--netmask</code>	NETMASK	Sets the netmask of this device to NETMASK (for example, 255.255.0.0)
<code>--gateway</code>	GATEWAY	Specify the gateway IP address as GATEWAY.
<code>--defroute</code>	yes	no
<code>--mtu</code>	MTU	Set the maximum transmission unit (MTU) value to MTU
<code>-y</code>		Do not prompt for confirmation

3) Verify the IP address configuration using the `show` command followed by the device name:

```
network> show eth0
```

Configure the CT-VL DNS Server

If your network environment uses DNS, use the `dns` command to configure the nameservers:

```
network> dns --add IP [IP ...]
```

You can configure a maximum of three nameservers.

Show the DNS settings with the `--show` option:

```
network> dns --show
DNS resolver is using DHCP
Generated by DHCP:
nameserver 192.168.100.199
nameserver 192.0.0.299
```

Note

If you are using DHCP, using the `dns` command overrides the nameservers provided by the DHCP.

The `/etc/hosts` file

If you have hostnames that do not use DNS, you can use the local `/etc/hosts` file to resolve IP addresses. To

update this file, run the CLI `system hosts` command, which includes options to add or remove IP addresses, and to show the `hosts` file contents.

```
main> system hosts
usage: hosts [-h] [--show] [--add IP [HOSTNAME ...]] [--remove IP]
Update hosts file
optional arguments:
-h, --help show this help message and exit
--show Show hosts file entries
--add IP [HOSTNAME ...]
Add/update hosts entry
--remove IP Remove hosts entry
```

For example:

```
main> system hosts --add 192.168.99.99 rgrimes
192.168.99.99 rgrimes
10.3.110.187 cts-builder2
main> system hosts --show
192.168.99.99 rgrimes
10.3.110.187 cts-builder2
```

Note

You must do this on each CT-VL, since entries in the hosts file are not replicated across CipherTrust Vaultless Tokenization Server.

Setting the CT-VL Hostname

Set a hostname for the CT-VL VM. This hostname will be used to identify the CT-VL when the CipherTrust Manager issues a client-certificate.

To set the hostname of the CT-VL VM, run the `system hostname` command with the `--set` option in the CT-VL CLI:

```
main> system hostname --set your_cts_hostname
```

Importing a Server Certificate for TLS

Use the `ct-vl server_certificate` command in the CT-VL CLI to import a server certificate.

Enter the command without any options to see a detailed help message that explains all arguments to the command (some of which are deleted from the example shown below to save space):

```
main> ct-vl server_certificate
usage: server_certificate [-h] [--import {certificate,key}]
[--create {csr,selfsigned,key}] [--size SIZE]
[--show {key,certificate}] [-y] [--country C]
```

```
[--state ST] [--locality L] [--organization O]
[--organizational_unit OU] [--common_name HOSTNAME]
[--email EMAIL] [--dns [DNS [DNS ...]]] [--days DAYS]
Server Certificate Utility
optional arguments:
-h, --help show this help message and exit
--import {certificate,key}
import a certificate or private key
--create {csr,selfsigned,key}
create a certificate signing request, a self-signed certificate, or a new private key.
a certificate signing request is created using [ . . . ]
```

Certificates in PEM Format

All certificates used in CT-VL must be in PEM format. The following is an example of a certificate in PEM format:

```

-----BEGIN CERTIFICATE-----
MIIDrTCCAxagAwIBAgIBADANBgkqhkiG9w0BAQQFADCBnDEbMBkGA1UEChMSVGHl
IFNhbXBsZSBDb2lwYW55MRQwEgYDVQQLEwtdQSBEaXZpc2lvbWJlcmBoGCSqSgSIb3
DQEJARYNY2FAC2FtcGxLLmNvbTETMBEGA1UEBxMKTWV0cm9wb2xpczERMA8GA1UE
CBMITmV3IFlvcmsxCzAJBgNVBAYTALVTRMRQwEgYDVQQDEwttUU0MgUm9vdCBDQTAE
Fw0wMTEyMDgwNDI3MDVaFw0wMjEyMDgwNDI3MDVaMIGcMRswGQYDVQQKEhJUAUgUg
U2FtcGxLIENvbXBhbGknFDASBgNVBASTC0NBIERpdmlzaW9uMRwwGgYJKoZIhvcN
AQkBFg1jYUBzYW1wbGUuY29tMRMwEQYDVQQHEwptZXRYb3BvbGZlMREwDwYDVQI
EwhOZXcgwW9yazELMAKGA1UEBhMCMVVMxvFDASBgNVBAMTC1RTQyBSb290IENBMIGf
MA0GCSqGSIsb3DQEBAQUAA4GNADCBiQKBgQDaiAwfKB6ZBtntRTIo6ddomt0S9ec0
NcuvtJogt0s9dXpHowh98FCDjnLtCi8du6LDTZLuhl0tTFARPlV/LVnpsbyMCXMs
G2qpdjJop+XIBdvoCz2HpGXjUym8wLqt+coWwJqUSwiEba74JG93v7TU+Xcvc00
5MWhxmKZzD/R3QIDAQABo4H8MIH5MAwGA1UdEwQFMAMBAf8wHQYDVR00BBYEFg/v
yytrBtEquMX2dreysix/MLPMIHIJBgNVHSMEGcEwgb6AFG/vyytrBtEquMX2drey
six/MLPMoYGipIGfMIGcMRswGQYDVQQKEhJUAUgUgU2FtcGxLIENvbXBhbGknFDAS
BgNVBASTC0NBIERpdmlzaW9uMRwwGgYJKoZIhvcNAQkBFg1jYUBzYW1wbGUuY29t
MRMwEQYDVQQHEwptZXRYb3BvbGZlMREwDwYDVQI
EwhOZXcgwW9yazELMAKGA1UE
BhMCMVVMxvFDASBgNVBAMTC1RTQyBSb290IENBggEAMA0GCSqGSIsb3DQEBAQUAA4GB
ABcLymJfsPOuazNQ08aIaxwVbXWS+8AFekMMRx6068ICAMubQBvs8Buz3ALXhqYe
FS5G13pw2ZnALsdTkSTKkE5wGZ1RYSfyiEKXb+u0KhdN9LnajDzaMPkNDU2NDXDz
Sghk9ZiE1boQamZjNlu+KabTLpmL9uXvFA/i+gdenFHV
-----END CERTIFICATE-----

```

Option 1: Regenerating a Self-signed Server Certificate

To generate a self-signed certificate, run the `ct-vl server_certificate` command with the `--create selfsigned` option:

```
main> ct-vl server_certificate --create selfsigned
```

Enter answers to the informational questions, and restart the web server when asked.

Option 2: Importing an Authenticated Third-party Server Certificate (Recommended)

To import an authenticated third-party CA certificate to the CT-VL, follow the steps below:

1. In the CT-VL CLI, go to the `ct-vl` category, and enter `server_certificate --create csr` to generate a certificate request:

```
main> ct-vl
ct-vl> server_certificate --create csr
```

2. Follow the system prompts. A certificate request displays at the command line:

```
----- BEGIN CERTIFICATE REQUEST -----
. . .
----- END CERTIFICATE REQUEST -----
```

3. Copy and paste the certificate request including the "REQUEST" lines onto a third-party certificate request form, such as GoDaddy, Thawte, or Verisign.
4. After receiving the certificate, at the `ct-vl` command category, run `server_certificate --import certificate` at the command line:

```
ct-vl> server_certificate --import certificate
```

5. When prompted to continue, enter `yes`.
6. Note that the certificate must be in PEM format. Copy and paste the entire certificate including the CERTIFICATE lines, as shown in the example below:

```
-----BEGIN CERTIFICATE-----
MIID1TCCAr2gAwIBAgIJAMf9x1/DhPfPMA0GCSqGSIb3DQEBCwUAMIGAMQswCQYD
VQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcn5pYTERMA8GA1UEBwwIU2FuIEpvc2Ux
. . .
9hZV7uAAnRzqaodIPQYAQBwbj58/eo8hMw4JflfJwTavZuFt/hpnbxEECynd+s4
cLSgBd7x4aeDKt0vinSvBv2niyZyL0v1eg==
-----END CERTIFICATE-----
```

7. Enter `Ctrl-d` to end the input.
8. The CT-VL imports the certificate and tries to verify that the imported certificate matches the private key generated at the time the certificate was generated.
9. The system prompts you to restart the web server. Type `yes`.
10. The web server component of the CT-VL is restarted, and the new certificate takes effect. The system is set up using the third-party server certificate for TLS connections.

See Next

Registering CT-VL with CM
Configuring CT-VL Nodes
Configuring an AD/LDAP Server
Configuring ADFS and LDAP for OAuth
Configuring the settings.ini file for LDAP
Restarting CT-VL server

Restarting CT-VL

After all CLI configuration is completed:

1. Navigate to the ct-vl commands menu:

```
main > **ct-vl**
ct-vl
```

2. Restart the web server component of CT-VL:

```
ct-vl> service --restart
Restarting ct-vl (via systemctl)   [OK]
ct-vl>
```

Registering CT-VL VM with CipherTrust Manager

Complete the following procedure to register CT-VL with the CipherTrust Manager (CM) server.

1. Create a user and set a password for it.

Besides client certificate authentication, CM servers require user/password authentication for key access.

2. **Optional:** Create a user-group on the CM server.

In addition to granting key access to a user, key access can be granted to a group. Users belonging to that group have access to the group keys.

3. Create symmetric AES keys for CT-VL use on the CM server using the properties below:

- Key type: **AES**
- Key size: **256 bits**
- Exportable: **yes**
- Prevent keys from being deleted: **yes**

- Key properties: **Encrypt** and **Decrypt**
- Owner: **CT-VL user in Step #1 above**
- Group: **Group in Step #2 above**

Note

Check for [Recommendations for Secure Initialization Vector](#) in encryption requests.

4. If not already created, create a local CA on the CM server.

This will be used by CT-VL for client certificate authentication.

5. Create an NAE interface on the CM server using the following properties:

- Type: **NAE**
- Port number: **9000**
- Mode: **TLS**; user must supply password; verify client certificate
- Username Location in Certificate: **CN**
- Local CA: (select)
- Network interface: (select)

6. Set up CT-VL with IP(s), port, and user credentials to the CM server.

- a. Log in to the CT-VL CLI and run the `icapi --set` command:

```
icapi set --IP <CM_ip> --port 9000 --user ctvluser
```

- b. If the `ctvluser` is under a domain, supply the domain name as shown below. Take note of the double bar (pipe) character separating the `domainname` and `ctvluser`.

```
icapi set --IP <CM_ip> --port 9000 --user domainname|ctvluser
```

- c. When prompted, enter the password of the `ctvluser` on the CM server.
- d. If the key manager is a cluster of servers, supply all the IP addresses of the servers in the cluster. For example:

```
icapi set --IP ip1 ip2 ip3 --port 9000 --user <ct-vl-user>
```

- e. Run the **icapi show --properties** command to inspect the settings made. For example:

```
main> icapi show --properties
Client_Cert_Authentication yes
KEYSTORE_PASSWORD *****
KEYSTORE_USER ctvluser
NAE_IP ['10.3.211.79']
NAE_Port 9000
Protocol ssl
Version 3.1
```

7. Register CT-VL to the CM server with either an automated registration or a manual registration.

- To perform automated registration:

- a. Log in to the CT-VL CLI and run the **icapi register** command. For example:

```
main> icapi register --host 10.3.211.79 --user <adminuser>
```

This command requires the user credentials on the CM server that has admin rights to the local CA. It creates a client certificate on the CT-VL that is signed by the CM server. This serves as the client certificate authentication whenever CT-VL connects to the CM server.

This command prompts for the password of the CM admin user and the information to create the certificate.

For example:

```
main> icapi register --host 10.3.211.79 --user admin password:
Running registration will delete any existing CA and client certificates.
Do you wish to continue? <y/n> [n] y
Country Name (2 letter code) [US]:
State or Province Name [California]:
Locality Name (e.g., city) [San Francisco]:
Organization Name (e.g., company):
Organizational Unit Name:
Common Name (eg, server's hostname) [ctvl-250-263-202]:
Created Private-Key: (2048 bit)
CA certificate updated
Client certificate updated
Registration complete
CM signed certificate:
```

```

id : f22e722f-4135-44c7-9e78-c03b1acc83da
ca : kylo:kylo:naboo:localca:8d2889d6-45a4-4981-95e8-d1589238deb8
issuer : /C=US/ST=MD/L=Belcamp/O=Gemalto/CN=CM Root CA
notAfter : 2030-02-09T21:05:22Z
serialNumber : 195829222921448800841697297203339489128
sha1Fingerprint : 4C5B96B0F0C3D6FCF06C905E49965E4BCF370824
sha256Fingerprint :
EB7E39AD001EF9B63F9DF1370569869813869B1D592F0130CF6CADB501E625F1
subject : /C=US/ST=California/L=San Francisco/CN=ctvl-250-263-202

```

A successful registration will display the client certificate information and its fingerprint.

- b. Log in to the CM server and verify that the certificate information and fingerprint match

Notes:

- If the CM server has more than one Local CA, the automated registration uses the first Local CA found. If the desired Local CA is not the first on the list, use the **--caid** option to supply the ID of the desired Local CA. You can use the **show --LocalCAs** command to view the ID of all the local CAs on the CM server:

```
main> icapi show --LocalCAs --user <adminuser>
```

- By default, the certificate is set to expire in 10 years. Use the **--days** option to set the desired expiration.
- The registration command can be run in non-interactive mode by supplying all parameters to the command. Run the **icapi register --help** command for details.
- To perform manual CM registration:

Use the following procedure to register CT-VL to a CM server. You can also use this manual procedure if you do not want to automatically register, or for some other reason cannot automatically register.

Some reasons for using manual registration include:

- Network connectivity from CT-VL to CM is not yet in place.
- The user performing auto-registration does not have administration credentials to the CM server. The user credentials to perform registration must include permission to run REST APIs on the server, permission to download the Local CA, and permission to sign a certificate request.
- Access to the REST API (port 443) of the CM server is blocked.
- An imported private key is desired instead of letting CT-VL generate a random key internally.
- A common client certificate is desired for several CT-VL servers, for example, with CT-VL servers in a

cluster.

Steps:

- a. Create a certificate private key on the CT-VL:

```
main> icapi create --client_key
```

You may import a private key instead of letting CT-VL create one. Use the **icapi upload** command to import a private key.

- b. Create a certificate signing request:

```
main> icapi create --csr
```

This command creates a certificate signing request using the private key you have created or imported.

- c. Using the CSR you now generated, log in to the CM server and locate the Local CA to sign the CSR.

The signed CSR produces a certificate in PEM format.

- d. Import the signed certificate into the CT-VL using the upload command:

```
main> icapi upload --client_certificate
```

- e. Log in to the CM server and download the Local CA. Import this certificate into the CT-VL using the **icapi upload --CA_certificate** command:

```
main> icapi upload --CA_certificate
```

Note

When a self-signed certificate has been added, chain construction stops. In this case, it must fully match a trust anchor, otherwise chain building fails. If you are using a self-signed NAE server certificate, add the issuer of the self-signed certificate to the Trusted CAs list on the client even if the root CA is already added to this list.

The NAE server certificate will be considered a self-signed certificate if the **Issuer** and **Subject** fields are the same in the certificate.

8. Enable ICAPI.

Run **icapi enable yes** command to enable CT-VL access to the CM server. For example:

```
main> icapi enable yes
Restarting vts (via systemctl):      [ OK ]
ICAPI properties updated
```

9. Test connectivity to the CM server.

Use the **icapi test --server_connection** command to test server connectivity to the CM server. For example:

```
main> icapi test --server_connection
Test server connection using user: vts-user1
10.3.211.79: OK
```

This command tests the connection and authentication to the CM server. A successful authentication confirms correct CM user credentials and a valid client certificate.

10. Test access to an AES key.

Use the **icapi test --key** command to test access to an AES key on the CM server. For example, to test access to a CM key named vts-key:

```
main> icapi test --key vts-key
vts-key: OK
```

This command tests if the saved user credentials provide proper access to the specified key. This test requires correct user and client certificate authentication to the CM server.

Troubleshooting and Automation Tips

- Make sure the **icapi test** CLI commands pass before setting up CT-VL for tokenization use.
- Use the **icapi show** CLI command to inspect if the icapi agent is enabled or not, if certificates are created or uploaded, and if server settings are correct.
- All CLI commands can be run in non-interactive mode. This mode is ideal for automating setup.

Security Tips

Consider the following guidelines when configuring CT-VL to ensure maximum security:

- When setting up the NAE interface, it is highly recommended to use the following mode for best security:
 - TLS, user must supply password, verify client cert.
 - Although other modes are possible, anything other than the above would be less secure.
- When creating a user for CT-VL to use on the CM server, do not use a user with admin privileges or access to

any keys that CT-VL will not use. The Keystore user should only have the following:

- login access.
- key access privileges.
- access to the keys that CT-VL will use.

Note

Do not add the Keystore user to the "Key Users" group as this may allow the user access to keys that CT-VL does not use. Preferably, create a group for CT-VL, assign to that group all the keys that CT-VL will use, and then add the Keystore user to that group.

Configuring CT-VL Nodes

To set up a CT-VL system, a single Virtual Machine (VM) or multiple VMs can be used to form a cluster of nodes operating as one coherent CT-VL system.

A cluster of nodes is useful to achieve scalability and high availability. You can set up a CT-VL cluster manually, or using an automated process.

It is highly recommended to use the automated process. It is easier to follow and less error-prone, especially when setting up a cluster of more than two nodes.

A CT-VL system is set up by creating the first node in the cluster. At a minimum, a CT-VL system can run with just one node. It is still considered a cluster, but with only one node. Additional nodes can then be added to the cluster.

Note

CT-VL supports up to 16 nodes in a cluster.

Automated Cluster Setup

To automatically set up a cluster:

1. Log into the CLI and enable the `apiadmin` user on all nodes will join the CT-VL cluster. Enabling the `apiadmin` user requires setting a password. By enabling the `apiadmin` user, the primary node will be able to access a remote node and join it to the cluster:

```
main> cluster apiadmin [--enable] [--setpasswd PASSWORD]
```

2. Choose a node to start your cluster. Log into the CLI and run the command below. This `create` command

will initialize the node as the first node of the cluster:

```
main> cluster create
```

3. On the same node, start joining other nodes to the cluster, by running the `remotejoin` command specifying the IP addresses of the other nodes you wish to join the cluster:

```
main> cluster remotejoin NODE_IP [NODE_IP ...]
```

4. It is recommended to disable the `apiadmin` user after you have finished the automated cluster setup. Run the command below on all nodes to disable `apiadmin` access:

```
main> cluster apiadmin --disable
```

5. To add a new node to the cluster at a later time, re-enable the `apiadmin` user on all nodes of the cluster. Run the `remotejoin` command as before to add more nodes to that cluster:

Note

The `remotejoin` command can be run from any active node in the cluster.

Warning

Since a node joining a cluster will mirror the data of the cluster, any existing data in that node will be destroyed.

```
main> cluster remotejoin NODE_IP [NODE_IP ...]
```

Manual Cluster Setup

Similar to automated setup above, choose a node to start your cluster. Run the `cluster create` command to initialize the node as the first node of the cluster. The command requires the following parameters:

- **IP** - IP address of this node. It is highly recommended to use the internal IP, for example, the VPC IP address of this node. The IP address is optional if creating a cluster node using existing data (with the `--preserve` option).
- **USERNAME** - WebUI Admin username.

- **PASSWORD** - WebUI Admin password.
- **EMAIL** - WebUI Admin email.

The `cluster create` command supports both interactive and non-interactive modes. See the `--help` option output for details on how to use:

```
main> cluster create --help
usage: create [-h] [-u USERNAME] [-p PASSWORD] [-e EMAIL] [--preserve] [-y]
[IP]
usage: create [-h] [-u USERNAME] [-p PASSWORD] [-e EMAIL] [--preserve] [-y]
[IP]
```

Once this command completes, you can log in to the CT-VL Web UI using the credentials entered above. Load the Web UI by loading the node IP address or the external IP address of the VM:

- `https://node_IP`
- `https://external_IP`

Manually Adding Nodes to a Cluster

Unlike the automated process above, manually adding nodes to a cluster is more complicated. This process does not require `apiadmin` access. A node joining a cluster must be initiated from the node that is joining.

Although the node joining will only contact one active node in the cluster, all other active nodes in the cluster must be informed of the new node that is joining.

Similarly, the node that is joining must be informed of all the active nodes in the cluster. This aspect is crucial since data from each node in the cluster must be able to replicate on all nodes in the cluster. Informing a node is done by using the `cluster add` command.

Similar to the automated process, a cluster is created by choosing the first node of the cluster. This is done by running the `cluster create` command. This will create a cluster but with only one node.

Note

Before running the `join` command, the `add` command must be run on the first node. This is a way of telling the first node that an additional node will be joining the cluster. Simply running the `join` command on the new node will not work. The first node must know about the new node by running the `add` command.

To add a second node to a cluster:

1. Log in to the CLI of the first node and run the cluster add command:

```
main> **cluster add NODE2_IP**
```

where `NODE2_IP` is the IP address of the new (second) node joining the cluster.

2. Log in to the CLI of the second node. Run the **add** command for the first node, and then the **join** command:

```
main> **cluster add NODE1_IP**  
main> **cluster join NODE2_IP NODE1_IP**
```

The **join** command requires two parameters: **NODE2_IP** (the current, new node IP) and **NODE1_IP** (the existing cluster node IP).

The **add** command is optional and not necessary since the **join** command will take care of it. However, if you are joining a cluster already having more than one node in the cluster, you must run the **add** command for all nodes already in the cluster.

The **join** command only lets you select one node_ip to join for connectivity purposes, but you will need to tell the system all the other nodes in the cluster you are joining and that is accomplished by running the **add** command.

Example: Manually Setting up a Four-node CT-VL Cluster

The following procedure shows how to manually set up a CT-VL system with four nodes, with IPs **NODE1_IP**, **NODE2_IP**, **NODE3_IP**, and **NODE4_IP**.

1. On node1, run the **create** command:

```
main> **cluster create NODE1_IP**
```

2. On node1, run the **add** command for all nodes:

```
main> **cluster add NODE1_IP**  
main> **cluster add NODE2_IP**  
main> **cluster add NODE3_IP**  
main> **cluster add NODE4_IP**
```

Note

The **add NODE1_IP** command above is redundant since **NODE1** is already joined to the cluster. But for clarity, this repetition shows that the system must be told of all nodes joining the cluster.

3. Repeat the same **add** commands shown in Step 2 on nodes **NODE2**, **NODE3**, and **NODE4**.
4. Log in to **NODE2** and run the **join** command:

```
main> **cluster join NODE2_IP NODE1_IP**
```

5. Log in to **NODE3** and run the **join** command:

```
main> **cluster join NODE3_IP NODE1_IP**
```

Note

NODE3 can also be joined to the cluster using NODE2_IP as the second parameter instead of NODE1_IP. This is because NODE2 is already joined to the cluster. Any node can be used as the second parameter as long as that node is already joined to the cluster. **For example**, the following is equivalent to the preceding example above:

```
main> **cluster join NODE3_IP NODE2_IP**
```

6. Repeat the `join` command for NODE4.

```
main> **cluster join NODE4_IP NODE2_IP**
```

Log in to the GUI on all nodes and verify that they are all replicating. Changing a value in the GUI of one node should be reflected in the same changes on all other nodes.

Note

To add a new node to a cluster, all nodes in the cluster must be online. No new nodes can be added if not all nodes in the cluster are online.

Remove a Node from a Cluster

The process of removing a cluster node is the same for a manually-created cluster as it is for an automatically-created one.

To remove a node from a cluster, run the `remove` command from one node -- any node except the node that will be removed. You only need to run the `remove` command on one node. All other nodes see this command and remove the node from their system.

As with adding a node to a cluster, to remove a node from the cluster, all nodes must be online so that they will know that a node is being removed.

Note

- The order of the list of the CipherTrust Managers used by CT-VL does not matter. CT-VL picks a CipherTrust Manager to be used in a round-robin manner.

- The CipherTrust Manager used to register CT-VL does not have preference for usage. After a CT-VL is registered, it only retains the certificate of the CipherTrust Manager when registered. That certificate is present on all the CipherTrust Managers in the cluster (due to replication), thus CT-VL thinks all of them are the same.

CT-VL may continue to add more CipherTrust Managers to the CipherTrust Manager cluster without registering. It may even remove the original CipherTrust Manager from the cluster when registered without having any impact.

Configuring an AD/LDAP Server

The Active Directory/LDAP server is optional. However, you can use it to import CT-VL users and CT-VL GUI Administrators. See [Active Directory/LDAP Server Requirements](#) before configuring. You must configure the AD/LDAP server on every node in your CT-VL HA cluster.

1. Go to the `auth` command category in the CT-VL CLI to see the `ldap` command.

```
main> auth
auth$ ?
      Command      Description
      =====
      ldap          LDAP Setup
      oauth2        OAuth2 Setup
```

2. Like all CLI commands, you can type the command without a parameter and a usage example is displayed:

```
auth> ldap
usage: ldap [-h] [--show] [--setup] [--enable {true,false}]
           [--server SERVER_URI] [--bind_dn BIND_DN]
           [--bind_password PASSWORD] [--user_prefix PREFIX]
           [--user_search_filter FILTER]
           [--user_search_scope SCOPE] [--group_search_scope SCOPE]
           [--active_user_group GROUP] [--staff_user_group GROUP]
           [--super_user_group GROUP]
           [--ca_cert {import,remove,show}] [--test {bind_dn}]
           [--query_user NAME] [-y]

LDAP Setup

optional arguments:
-h, --help            show this help message and exit
--show               show LDAP settings
```

```
[ . . . ]
```

3. Disable, then enable LDAP support in the Tokenization Server:

```
auth> ldap --enable false
LDAP settings have been saved
auth> ldap --enable true
Restarting cts (via systemctl):      [ OK ]
LDAP settings have been saved
auth>
```

Note

If you disable LDAP support, then you can use only the CT-VL GUI to create users and groups.

4. Set the AD/LDAP server URI with the `--server` command (where the default port number for an LPDAS server is 636):

```
auth> ldap --server ldaps://<hostname_or_IP_address>:<Port>
```

or (where the default port number for an LDAP server is 389):

```
auth> ldap --server ldap://<hostname_or_IP_address>:<Port>
```

Example:

```
auth> ldap --server ldap://192.168.118.99:389
LDAP settings have been saved
auth>
```

5. Set the binding DN of the user who has access to the AD/LDAP server with password. Example:

```
ldap --bind_dn <FQDN of user with access to LDAP Server>
```

Example:

```
auth> ldap --bind_dn cts0@cts.vormetric.com
Enter Password :
```

```
Enter Password again :  
Set LDAP Bind User DN SUCCESS  
auth>
```

6. Set user search scope using `user_search_scope` command:

```
ldap --user_search_scope <User search scope>
```

Example:

```
auth> ldap --user_scope dc=cts,dc=vormetric,dc=com  
LDAP settings have been saved  
auth>
```

7. Set group search scope using the `group_search_scope` command:

```
ldap --group_search_scope <Group search scope>
```

Example:

```
auth> ldap --group_search_scope CN=Users, dc=cts,dc=vormetric,dc=com  
LDAP settings have been saved  
auth>
```

8. Set attribute field for user search filter using the `user_search_filter` command:

```
ldap --user_search_filter <Name of user field in AD/LDAP>
```

Example:

```
auth> ldap --user_search_filter sAMAccountName  
LDAP settings have been saved  
auth>
```

9. Set user prefix name using `user_prefix` command. The user prefix is the attribute that should be used to prefix the user name when searching for the user name in the LDAP database.

```
ldap --user_prefix <User prefix name. Example: uid or cn or sn>
```

Example:

```
auth$ user_prefix uid
Set LDAP USER PREFIX SUCCESS
```

10. Set the Tokenization Server active user group using the `active_user_group` command:

```
auth> ldap --active_user_group <Group whose members can access the RESTful AP
I>
```

In this example, any user who wants access to the CTS RESTful API, must belong to the AD group `ctsUsers` within Users of the domain `cts.vormetric.com`:

```
auth> ldap --active_user_group cn=ctsUsers,cn=Users,dc=cts,dc=vormetric,dc=com
LDAP settings have been saved
auth>
```

11. Set the CT-VL superuser group using the `--super_user_group` command:

```
ldap --super_user_group <Group that can access the Tokenization Server GUI & A
PIs>
```

In this example, any user who requires access to the CTS GUI and REST APIs, must belong to the AD group `ctsSuperuser` within Users of domain `cts.vormetric.com`:

```
auth> ldap --super_user_group cn=ctsSuperuser,cn=Users,dc=cts,dc=vormetric,d
c=com
LDAP settings have been saved
auth>
```

12. **For LDAPS Support Only:** Import the CA certificate chain of the LDAP server to the built-in LDAPS client of CT-VL. A Certificate Management tool like DigiCert or Certificate Manager tool is recommended to request the certificate, after which use the `--ca_cert` option to import the certificate into the CT-VL.

Note

The `--ca_cert` option uses an underscore (`_`) instead of a hyphen (`-`).

- a. Using a Certificate Management tool like DigiCert or Certificate Manager tool, download your intermediate and root certificates.

Note

CT-VL requires all certificates to be in PEM format.

- b. Run `--ca_cert` import to import the certificate trust chain. Paste the entire body of each certificate in the order shown in the example, then terminate with Ctrl-d when finished.

```
auth> ldap --ca_cert import
Cut-and-paste below a CA Certificate in pem format. Press Ctrl-d when done.
-----BEGIN CERTIFICATE-----
(Your Intermediate certificate: DigiCertCA.crt)
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
(Your Root certificate: TrustedRoot.crt)
-----END CERTIFICATE-----
Restarting cts (via systemctl): [ OK ]
LDAP CA Certificate has been uploaded.
auth>
```

13. Execute the `--show` option to list the LDAP configuration:

```
auth> ldap --show
ENABLE true
SERVER_URI ldap://192.168.118.99:389
BIND_DN cts0@cts.vormetric.com
BIND_PASSWORD *****
USER_SEARCH_SCOPE dc=cts,dc=vormetric,dc=com
GROUP_SEARCH_SCOPE CN=Users,dc=cts,dc=vormetric,dc=com
USER_SEARCH_FILTER sAMAccountName
ACTIVE_USER_GROUP cn=ctsUsers,cn=Users,dc=cts,dc=vormetric,dc=com
SUPER_USER_GROUP
auth>
```

Configuring ADFS and LDAP for OAuth

Follow these procedures to configure Active Directory Federated Services (ADFS) and LDAP for an OAuth authentication option.

Export Certificate

Below are the steps to export a certificate for CT-VL:

-
1. Open Server Manager.
 2. Click **Tools** and select **ADFS Management**.
 3. Click **Certificates**.
 4. Double-click on **Token-signing certificate**.
 5. In the *certificate* window, select **Details** tab.
 6. Click **Copy to File...**
 7. Click **Next**.
 8. Select **Base64 format** and click **Next**.
 9. Enter file name.

Add an Application Group

Below are the steps to add an application group:

1. Open Server Manager.
2. Click **Tools** and select **ADFS Management**.
3. Right click **Application Groups** and select **Add Application Group** wizard.
4. Enter name and select **Native application accessing a web API**.
5. Copy **Client Identifier** to a template for later use.
6. Enter your CT-VL login URL in **Redirect URI field** (<https://login>) and click **Add**.
7. Click **Next**.
8. Enter the same CT-VL URI as entered in previous step in the **Identifier** field and click **Add**.
9. Click **Next**.
10. Use default (**Permit everyone**) and click **Next**.
11. Click **Next** on the review page.
12. Click **Close**.

Add Properties to a New Group

Below are the steps to add properties to a new group:

1. Double-click on the desired new group.
2. Select - **Web API** and click **Edit**.
3. Select **Issurance Transform Rules** and click **Add Rules**.
4. Select **Claim rule template: Send LDAP Attributes as Claims**.
5. Click **Next**.
6. Enter a rule name.
7. Add the following rules:
 - Token-Groups - Unqualified Names : Group
 - User-Principal-Name : UPN
 - SAM-Account-Name : Name
 - Surname : Surname
 - Given-Name : Given Name
8. Click **Finish**.
9. Click **OK**.

Test ADFS Authentication

To test ADFS authentication:

```
curl -k -X POST https://<adfs server>/adfs/oauth2/token -d  
'grant_type=password&username=<user>&password=<password>&client_id=<client_id>5&resource=https%3A%2F%2Fmickeymouse'
```

The expected response is:

```
{"access_token": "eyJ0eXA... mVSpzmW-FBDlg"}
```

Configuring LDAP Settings.ini File

Set up the `settings.ini` file according to which LDAP scenario you implement:

- AD Email
- AD DN
- Linux LDAP DN

To test your settings, use the CLI option `--query_user` to the `ldap` command:

```
auth> ldap --query_user NAME
```

AD Email

The following is an example `settings.ini` configuration for an LDAP setup with AD email:

```
auth> ldap --show
SERVER_URI ldap://10.3.118.98:389
BIND_DN cts0@cts.vormetric.com
BIND_PASSWORD *****
USER_PREFIX
USER_SEARCH_FILTER sAMAccountName
USER_SEARCH_SCOPE CN=Users,DC=cts,DC=vormetric,DC=com
GROUP_SEARCH_SCOPE CN=Users,DC=cts,DC=vormetric,DC=com
ACTIVE_USER_GROUP CN=ctsUsers,CN=Users,DC=cts,DC=vormetric,DC=com
STAFF_USER_GROUP CN=ctsManager,CN=Users,DC=cts,DC=vormetric,DC=com
SUPER_USER_GROUP CN=ctsAdmin,CN=Users,DC=cts,DC=vormetric,DC=com
ENABLE true
```

AD DN

The following is an example `settings.ini` configuration for an LDAP setup with AD DN:

```
auth> ldap --show
SERVER_URI ldap://10.3.118.98:389
BIND_DN CN=cts User0,CN=Users,DC=cts,DC=vormetric,DC=com
BIND_PASSWORD *****
USER_PREFIX CN
USER_SEARCH_FILTER cn
USER_SEARCH_SCOPE CN=Users,DC=cts,DC=vormetric,DC=com
GROUP_SEARCH_SCOPE CN=Users,DC=cts,DC=vormetric,DC=com
ACTIVE_USER_GROUP CN=ctsUsers,CN=Users,DC=cts,DC=vormetric,DC=com
STAFF_USER_GROUP CN=ctsManager,CN=Users,DC=cts,DC=vormetric,DC=com
SUPER_USER_GROUP CN=ctsAdmin,CN=Users,DC=cts,DC=vormetric,DC=com
ENABLE true
```

Linux LDAP DN

The following is an example `settings.ini` configuration for setup with LDAP DN on Linux:

```
auth> ldap --show
SERVER_URI ldap://10.10.62.132:389
BIND_DN uid=tuser1,ou=People,dc=i,dc=vormetric,dc=com
BIND_PASSWORD *****
USER_PREFIX uid
USER_SEARCH_FILTER uid
USER_SEARCH_SCOPE ou=People,dc=i,dc=vormetric,dc=com
GROUP_SEARCH_SCOPE ou=Groups,dc=i,dc=vormetric,dc=com
ACTIVE_USER_GROUP cn=ctsUsers,ou=Groups,dc=i,dc=vormetric,dc=com
```

```
STAFF_USER_GROUP cn=ctsManager,ou=Groups,dc=i,dc=vormetric,dc=com  
SUPER_USER_GROUP cn=ctsAdmin,ou=Groups,dc=i,dc=vormetric,dc=com  
ENABLE true
```

Working with CT-VL GUI

Use the CT-VL administration GUI to:

- Create tokenization users and groups.
- Specify permissions for users and groups.
- Create token templates.
- Create token groups.

Logging into the CT-VL GUI

To log in to the CT-VL administration GUI:

1. Point your browser to the Tokenization Server:

```
https://<CT-VL name or IP address>/
```

2. Log in using the Admin user name and password you defined when you created the CT-VL cluster.
3. If you have implemented OAuth authentication, log in with those credentials.

The CT-VL GUI Home page is displayed.

Troubleshooting

The CipherTrust Vaultless Tokenization Server Administration GUI login failure is indicated by the following error message:

```
nginx 504 Gateway Time-out error
```

In this scenario, consider the following problem/solution option:

- **Problem:** The Tokenization Server is not connected to the LDAP server.
- **Solution:** Restart the CT-VL service. Open the CT-VL CLI, and run the following command:

```
main> vts service --restart
```

Creating and Managing Passwords

CT-VL uses a dynamic password policy control mechanism to ensure system security.

CT-VL leverages a password checker library. If you try to create a password that does not pass the proactive password checking provided by this library, you are prompted to try another string. The policy can change dynamically based on several factors.

The dynamic nature of the checking algorithm provides the most secure possible password policy options.

When creating a system password, use the following guidelines:

- Use at least 1 digit, 1 uppercase letter, 1 lowercase letter, 1 special character, and use at least 8 characters in total.
- Do not use your name, the company name, or other familiar names in the password string.
- Do not use your birthday or any other personally or professionally meaningful dates in the password string.
- Include a combination of the following character types in the password string:
 - Lowercase letters
 - Uppercase letters
 - Numbers
 - Special characters

Managing CT-VL User Groups

User groups enable you to assign the same permissions to multiple users.

Refer to [CT-VL Roles](#) and [Understanding CT-VL User Groups and Permissions](#) to familiarize yourself with user groups in CT-VL.

If you are using AD/LDAP users, create at least three user groups, to correspond to the CT-VL roles:

- Superuser
- Staff
- Active

User groups must be defined on the LDAP system. The user group names must be entered into the CT-VL GUI before importing users.

Creating User Groups

To create a user group:

- 1) Log in to the CT-VL GUI and select **User Groups** from the navigation bar. **All User Groups** list is displayed.

2) Select **New User Group** and enter a name on the resulting page. Click **Create**. The new group name appears in the **All User Groups** list, and is selectable from user pages and the *Permissions* page.

Assigning Masks

You can assign a mask to a group when initially creating the user group (at the *New User Group* dialog), or any time later to an existing user group, by clicking the user group and adding a mask assignment at the *Update User Group* dialog.

Adding Users to User Groups

To add a user to an existing user group:

- 1) From the CT-VL GUI, select **User Groups**.
- 2) Click on a group from the list.
- 3) Select users to add to the group, and click **Update**.

Deleting a User Group

To delete a user group:

1. Log in to the CT-VL GUI and select **User Groups** from the navigation bar.
2. Select the check box by a group.
3. Click **Delete**. A dialog box appears asking you to confirm your action.

Note

If you delete a user group to which users and/or permissions have been associated, the user group is removed from any user belonging to it, and will delete any permissions assigned to it.

Managing CT-VL Users

Refer to [Understanding CT-VL Authentication and Authorization](#) for details on CT-VL users and CT-VL authentication and authorization basics.

You can perform the following to manage CT-VL users:

- Create a user manually.
- Import users from an AD/LDAP server.

Creating a User Manually

Creating Basic User Information

1. Log in to the CT-VL GUI and select **Users** from the navigation bar. The list of **All Users** is displayed.
2. Select **New User**. In the **New User** dialog, fill in the required information and click **Create**.

Note

- The username must contain only English letters, numbers, or the following special characters: “@” (at-sign), “.” (period), “+” (plus-sign), “-” (hyphen), or “_” (underscore).
- Usernames and passwords are case sensitive. Application code that calls the CT-VL RESTful APIs must pass usernames and passwords in the same case that are defined in the CT-VL GUI.

Updating User Information

After you complete the steps described in [Creating Basic User Information](#), you can add more attributes to the basic user, including changing the CT-VL role and assigning to a user group.

1. Log in to the CT-VL GUI, and select **Users** from the navigation bar.
2. Click on the username of interest.
3. Change CT-VL role, User Groups assignment, Mask, or Email, as needed.
4. Click **Change password here** to update the user’s password.
5. Click **Update**.

Deleting a User Manually

If you delete a user to which permissions have been associated, all permissions linked to the user are also deleted.

Note

It is generally recommended to uncheck the **Active** status for a user rather than completely delete the user.

1. Log in to the CT-VL GUI, and select **Users** from the navigation bar.
2. Select the check box by a user, click **Delete**, and confirm the deletion.

Importing Users from an AD/LDAP Server

Refer to [CT-VL Roles](#) and [Understanding CT-VL User Groups and Permissions](#) to familiarize yourself with user groups in CT-VL.

AD/LDAP User Import Prerequisites

Before you import users, complete the following tasks:

- Set up your AD/LDAP connection to CT-VL, and configure it using the CT-VL CLI. See [Configuring an AD/LDAP Server](#) for instructions.
- Create the appropriate user groups in the CT-VL GUI as described in [Creating User Groups](#).
- Set up the AD/LDAP server to import and authenticate users and user groups.

To import CipherTrust Tokenization Server users from an AD/LDAP server:

1. Create three AD/LDAP server groups, one for REST API users (Tokenization/Crypto/Key Management) and the other two for CT-VL administrators who require GUI access.
The group that has been designated as the “Superuser” group in the CLI will have read/write access in the GUI, whereas the group that has been designated as the “Staff” group in the CLI will have read-only access in the GUI.
2. Populate each group with the desired REST API (“active”) users and CT-VL administrators.

Note

- The name with permission to access the REST APIs must match the name specified with the `active_user` command under the `auth` category of the CT-VL CLI. See [Configuring an AD/LDAP Server](#) for further information.
- The name of the CT-VL users with read-only access to the GUI must match what was specified in the `staff_user` command of the `auth` category of the CLI.
- The name of the CT-VL administrators' group must match the name of the group that was specified in the `super_user` command of the `auth` category of the CT-VL CLI.

Example of AD/LDAP server group names:

- `ct-vlUsers` - Users in this group can create and/or access data through the RESTful API.
- `ct-vlStaff` - Users in this group can access the CT-VL GUI (read only).
- `ct-vlAdmin` - Users in this group can access the CT-VL GUI (read/write)

Note

- `staff_user` is valid only if the user also has `active_user` permission.
- `super_user` is valid only if the user has both `active_user` and `staff_user` permissions.
- Use corporate policy to define whether or not `staff_user` and `super_` users should actually be granted REST API permissions, and disallow by policy when necessary to maintain corporate standards or separation of duties.

3. Create three group permissions in the CT-VL GUI with the same names as the LDAP groups (example: `ct-vlUsers` , `ct-vlStaff`, `ct-vlAdmin`). Members of these AD/LDAP groups inherit these permissions.
 - An application program that uses Basic Auth (username/password) will try to authenticate to LDAP using these credentials.
 - An application program that uses a client certificate for authentication will use the Bind user's username and password for looking up the client certificate's common name (CN) in LDAP.
 - If the credentials are authorized, the LDAP user is imported into the user table of the CT-VL. Thus, there is no need to manually create the LDAP users in the GUI.
 - Once the user has been added to the CT-VL, permissions can be modified with another group permission or individually.

Note

User names and passwords are case sensitive. Application code that calls the REST APIs must pass usernames and passwords in the same case that are defined in the AD/LDAP server.

Unlocking User Accounts

As a security feature, users are locked out after six failed login attempts.

If a user fails to successfully login after six attempts, the account is locked for 15 minutes (to prevent brute force attacks). The maximum number of attempts and cool-off time duration can be modified in the CLI.

To unlock a user account:

1. Click **Logs** in the navigation menu.
2. Select the **Locked Accounts** tab. The locked accounts are displayed.

3. Click **Unlock** for the desired account(s).

Note

All successful and failed login attempts are displayed under the **Access Logs** tab in the **Logs** menu. Select **Success** or **Failed** from the pull-down menu to filter the list so that only the selected access type is shown.

Specifying CT-VL Keys

Refer to [Understanding CT-VL Keys](#) to understand the definition and usage of symmetric keys, asymmetric keys, and opaque objects.

Creating a Key Name

Use the CT-VL GUI to enter a key name that corresponds to a key that has been created in the CipherTrust Manager, and is to be used for CT-VL functions.

After the name has been entered into the CT-VL local database, the key is available to associate with user and user group permissions. Symmetric keys are also be available to associate with tokenization groups.

Note

The very first symmetric key entered into CT-VL needs to be entered from the GUI of the first cluster node. Enter this key as soon as possible, it will be used to create large lookup tables for the “RANDOM” tokenization mode.

1. After [Logging into the CT-VL GUI](#), select **Keys** from the navigation bar. The Keys List is displayed, with the **Symmetric Keys**, **Asymmetric Keys**, and **Opaque Objects** tabs in the top navigation.
2. Select the tab for the type of key you want (for example, **Symmetric Keys**), and click the **New Symmetric Key** button.

Note

In CT-VL, the very first key generated cannot be renamed, edited, or destroyed. There is a warning when creating the key. Once this initial setup is complete, you can create subsequent keys with the normal **New Symmetric Key UI**, for application FPE.

3. Enter the key name that was used in the CipherTrust Manager and click **Create**. The key name is displayed on the Keys List and will show as available in the **Permissions matrix** and **Tokenization Group** pages.

Deleting a Key Name

Deleting a key name from the CT-VL local database does not delete the key material (nor the key name) from the CipherTrust Manager. Deleting a key name deletes all tokenization groups, tokenization templates, and permissions associated to it.

1. Select **Keys** from the navigation bar. The Keys List is displayed, with the **Symmetric Key**, **Asymmetric Keys**, and **Opaque Objects** tabs in the top navigation.
2. Select the tab for the type of key you want (for example, **Symmetric Keys**), click the check box by the desired key, and click **Delete**.
3. Confirm and click **Delete** again.

Note

The very first symmetric key entered into the CT-VL can serve as a regular key for all encryption, signing, and tokenization operations, but it is also used to build internal cryptographic tables and hence must not (and cannot) be deleted.

Tokenization Setup

Refer to [Tokenization](#), [Templates](#), and [Masks](#) for the concepts behind setting up tokenization in the CT-VL GUI.

Note

Before setting up tokenization, make sure you have specified the CM keys.

Creating Tokenization Groups

Have at least one symmetric key created in the CM and entered in the CT-VL GUI, as described in [Specifying CT-VL Keys](#).

1. Select **Tokenization > Tokenization Groups**. The *All Tokenization Groups* list is displayed.
2. Click the **New Tokenization Group** button. The *New Tokenization Group* creation page is displayed.
3. Enter a Tokenization Group **Name**.

Note

Tokenization group names are case sensitive. Application code must pass tokenization group names in the same case that are defined in the CT-VL GUI.

4. Select a **Symmetric Key** from the drop-down list.

Warning

- Do not delete keys used for tokenization. If you delete a key, you will have to recreate the corresponding tokenization group to regain access to your tokens.
- Do not use auto-key-rotation features with tokenization.

5. Click **Create**.

Deleting Tokenization Groups

Before deleting a tokenization group, make sure that tokens created with the symmetric key associated with this tokenization group are not stored anywhere within (or outside) of the organization.

Warning

Once the tokenization group used to create these tokens has been deleted, they can no longer be detokenized.

To delete a tokenization group:

1. Log in to the CT-VL administration interface.
2. Select **Tokenization > Tokenization Groups**.
3. Select the group and click **Delete**.

Managing Character Sets

CT-VL now includes enhanced support for tokenizing international characters. In earlier versions, CT-VL supported three default character sets:

- All digits

-
- Alphanumeric
 - All printable ASCII (not available for RANDOM Mode)

For FPE (FF3) and FF1 modes, you can also define a custom character set, which allows you to tokenize and detokenize UTF strings.

Creating Custom Character Sets

To define a custom character set, you need to know the Unicode character range for the language(s) you want to support. For example, the range for Arabic is 0600-06FF, while the range for Thai is 0E00 - 0E7F.

To create a custom character set:

1. Select **Tokenization > Character sets**.
2. Click **New Character set**.
3. Enter a **Name** and **Range** or set of ranges for your set. The maximum size of a custom UTF character set definition string is 1536. This implies either 100 ranges, or 200 individual characters, or some combination thereof—for instance, 50 ranges plus 100 individual characters.
4. Click **Create**. The new set appears in the **All Character Sets** list, and is available to apply to tokenization templates.

Editing Character Sets

To edit a character set, click on its name in the list. Predefined character sets cannot be edited and should not be deleted.

Applying Character Sets

The three predefined character sets always appear in the drop-down list of character sets when you create a tokenization template. Once a custom set has been defined, it appears in the tokenization template creation menu as well.

See [Tokenization Template](#) for information about CT-VL tokenization templates.

Deleting Character Sets

Before deleting a character set, make sure that tokens created with this character set are not stored anywhere within (or outside) of the organization.

Caution

After the character set used to create these tokens has been deleted, they can no longer be detokenized, unless the character set is recreated very precisely. Even the order of the characters in the character set matters.

To delete a character set:

1. Navigate to **Tokenization > Character Sets**.
2. Select the character set from the list, and click **Delete**.

Prototyping Tokenization

CT-VL supports UTF-8 inside the `valuenam` field for "data" and "token".

Generate a file that has the JSON request in it containing this UTF-8 code, and then submit it via cURL.

A file can be submitted in cURL as shown here:

```
--data-binary @filename option.curl --tlsv1.2 -k -X POST -u
superuser:ssl12345 https://127.0.0.1/vts/rest/v2.0/tokenize --
data-binary @tokenizerequest.txt
```

Managing Tokenization Templates

Creating Tokenization Templates

1. In the CT-VL GUI, select **Tokenization > Tokenization Templates**.
2. Click **New Tokenization Template**.
3. Enter the values described in [Managing Tokenization Templates](#) field table.

Sample: Use a Tokenization Template in the REST API

After you specify a tokenization template, it can be used in the RESTful API application code. This is shown in the following example:

```
curl --tlsv1.2 -X POST -u EdYee:EdYee_password -d '{"tokengroup" :
"store1" , "data" : "9453677629008564", "tokenTemplate" :
"ctsUsersTemplate" }' https://192.168.12.88/vts/rest/v2.0/tokenize
```

Deleting Tokenization Templates

Before deleting a tokenization template, make sure that tokens created with this tokenization template are not stored anywhere within (or outside) of the organization. Once the tokenization template used to create these tokens has been deleted, they can no longer be detokenized, unless the tokenization template is recreated very precisely.

Caution

After the character set used to create these tokens has been deleted, they can no longer be detokenized,

unless the character set is recreated very precisely. Even the order of the characters in the character set matters.

To delete a tokenization template:

1. Navigate to **Tokenization > Tokenization Templates**.
2. Select the tokenization template from the list, and click **Delete**.

Managing Data Masks

Tokenization data masks hide specified parts of detokenized data. A mask is required for detokenization, even if the user is authorized to view the entire unencrypted string of data.

See [Data Masks](#) for more information on data masks used in CT-VL.

Creating a Data Mask

To create a data mask:

1. Select **Tokenization > Masks**. The **All Masks list** is displayed.
2. Click **New Mask** and enter the information on the **New Mask** dialog.

Some examples of data masks you could create:

- A mask called SHOW_LAST_4 that shows the last four digits of a tokenized credit card: XXXX-XXXX-XXXX-7897.
- A mask called SHOW_FIRST_3 that shows only the first three digits of a social security number: 565-XX-XXXX.
- A mask that shows everything (ALL_CLEAR): set Show Left and Show Right values to 999999.

3. Click **Create**.

Deleting a Data Mask

When a data mask is deleted, all user and user group permissions associated with that mask will be deleted.

Warning

This could result in production down time if a user is using the system and their permissions are deleted. They will be denied tokenization and detokenization rights. Be VERY sure these permissions are not in use before deleting the mask.

Applying Permissions to Users and User Groups

With CT-VL, users are authenticated, granted permissions (to tokenize, detokenize, encrypt, hash, etc.), and associated with particular keys, using a three-way permissions matrix.

Assigning Key-based Permissions

The CT-VL GUI Administrator assigns permissions based on user (or user group), action, and key, similar to the English sentence:

"User ctsuser1 may encrypt, decrypt, tokenize and detokenize with symmetric key key1."

Before permissions can be assigned, you must:

- Create user groups and users.
- Specify CT-VL Keys

Use the following steps to assign key-based permissions:

1. Log in to the CT-VL GUI and select **Permissions**. The User Permissions list is displayed. If necessary, click **User Group** to see the User Group Permissions list.
You can click the tabs to see the Symmetric Keys, Asymmetric Keys, and Opaque Objects assigned to a user (or group).
2. Click a user or group name to see a user permissions summary page. You can review and change permissions there or from the list page. To continue from the list page:
3. Click the permission symbol for a particular user's key. The user permission matrix is displayed.
4. Assign permissions as needed and click **Apply**.

Assigning Other Permissions

The permissions in the "Other Permissions" category are those that are not tied to specific keys. They are assigned only once per user group, and apply to all key groups. Actions that are not supported in the API are not displayed.

Creating and Importing Keys

To grant a user or user group permission to create keys on the CT-VL or to import keys to the CT-VL:

1. Log in to the CT-VL GUI and click **Permissions**. The User Permissions list is displayed. (If necessary, click **User Groups** to access the User Group Permissions list.)
2. Click the permission symbol for a user in the **Other Permissions** column. The Other Permissions assignment window is displayed.
3. Check **Create** and **Import**, as desired, and click **Apply**.

Note

- The **Create** permission is assigned only once per user and applies to all type of keys.
- The **Import** permission is assigned only once per user and applies to symmetric keys and opaque objects.

Other Permissions: Tokenize/Detokenize

Select **Tokenize/Detokenize** to apply privileges at a global level, on all symmetric keys in the system.

If your system is upgraded from an earlier version, any users who had been granted tokenize/detokenize privileges have these boxes checked automatically.

Remember to de-select the global check boxes and reassign privileges on a per-key basis to take advantage of the more granular structure of CT-VL configuration options.

Dashboard API Monitoring

The CT-VL Dashboard provides graphical and tabular monitoring for:

- CipherTrust Vaultless Tokenization activity
- CipherTrust Key Management activity
- CipherTrust Cryptography activity

Activity data is updated hourly, and can be sorted and filtered by:

- User (or all users)
- Date range
- Hour
- Day
- Week
- Month

The tabular view of the API data can be filtered by user or by user IP address (which can be used to check for access by unusual IP addresses). Clicking a user or IP address row in the table filters the graph view above it by that selection.

Exporting Data

Click the **Export** button to export into a CSV file the data as currently filtered and shown in the Dashboard.

Administrator History Logs

Most tokenization and detokenization log messages are in the file `tokenization.log`. You can get this via web download with credentials from the file `tokenization.log`.

Example:

`https://ctshost.vormetric.com/log/tokenization.log`

These log messages also go to the syslog with the facility code `daemon`.

Other log files:

- `django.request.log` - Logs any user request coming from the CipherTrust Vaultless Tokenization Server GUI.
- `uwsgi_cts.log` - Debug log.
- `clish.log` - CLI activities and events.

Note

Log data added by the server or server framework are not included in the log files.

View the CipherTrust Vaultless Tokenization Server logs by clicking the **Logs** icon in the CipherTrust Vaultless Tokenization Server GUI. (Audit log messages also go to the syslog with the facility code `user`).

From here you can:

- View audit logs, access logs (login and logout attempts), active sessions, locked accounts, and server CPU usage.
- Export the logs.

Viewing Audit Logs

To view the audit logs:

1. Log in to the CT-VL GUI.
2. Click **Logs** in the navigation pane.
3. Click the **Audit Logs** tab.

Details are shown for each audited event, including:

- **Date** specifies when the event occurred.
- **User** is the CipherTrust Vaultless Tokenization Server user.

-
- **Content Type** is the database table that was modified. *Tenant* involves Group Tokens, *Key* is encryption keys, *user* is for users, *groups* for groups, and so on.
 - **Object Name** is the specific database object that was created or changed.
 - **Modification** is the action that was performed.

Viewing Access Logs

The access log shows information about all attempts to log in to the server, and all logouts from it. The related information includes the access time, the IP address, username, and user agent (typically, a web browser). By default, all login attempts are listed, but the list can be filtered (by Status) to show only successful or only failed attempts.

To view the access logs:

1. Log in to the CT-VL GUI.
2. Navigate to **Logs**.
3. Click the **Access Logs** tab.
4. Filter the listed results with the **Type** menu. When viewing the **Logins** access type, use the additional **Status** pull-down menu to filter by Successful or Failed login attempts.

Downloading (Export) Access or Audit Logs

To download the history logs for an administrator:

1. Log in to the CT-VL GUI.
2. Navigate to **Logs**.
3. Click the **Audit Logs** or **Access Logs** tab.
4. Click the **Export** button for the Audit or Access Logs.

Showing Active Sessions

To see a list of active sessions of users currently logged in to the server:

1. Log in to the CT-VL GUI.
2. Click **Logs** in the navigation pane.
3. Click the **Active Session** tab.
4. To end a user's session on the GUI, click the **End Session** button in the row for that user.

Showing Locked Accounts

If a user fails to enter valid credentials more than five times, the user is locked from accessing the CT-VL for 15 minutes. View the details of locked accounts, including when the lock was activated and when it was released. Superusers can unlock an account before the 15 minute time limit is completed.

To view and manage locked accounts:

1. Log in to the CT-VL GUI.
2. Click **Logs** in the navigation pane.
3. Click the **Locked Accounts** tab.
4. Unlock an account by clicking **Unlock** in the row of the desired user.

System Logging

Audit log messages go to the syslog with the facility code `user`.

`tokenization.log` messages go to the syslog with the facility code `daemon`.

CLI log messages go to the syslog with the facility code `local3`.

Configuring Syslog Servers

Configure remote syslog servers with the CLI, with the command under the CT-VL category. You can also set up multiple syslog servers with this command, with the `--add` and `--remove` commands.

The `remotelog` command has utilities to check the connection status to the servers, and to send a sample log message to them.

Backing Up and Restoring CT-VL

Create regular backups of the server to restore CT-VL in the event of failure. Even in a clustered scenario, system failure can occur. To minimize this risk, backup your CT-VL implementation regularly using the administration interface or the REST API.

Use the CT-VL GUI to create a backup before upgrading your CT-VL version and before any major server configuration changes. Maintain a regular backup schedule using the CT-VL REST API.

Restore the state of your CT-VL implementation using the administration interface. Backups restore the CT-VL to the saved version in the following scenarios:

- Restoring the application after system failure.
- Restoring the application after upgrade failure.
- Restoring the application due to unrecoverable system configuration errors.

Note

A backup cannot restore a system using a CT-VL version older than 2.2.2. A backup cannot be applied to any version of the CT-VL software that is older than the version in use when the backup was created.

Backing up CT-VL in the GUI

Use the administration interface to create and save a backup.

Use the following steps to create a system archive:

1. Select **Backup** from the administration interface left hand navigation.
2. Click the **Create Backup** tab.
3. Accept the default values, and click **Backup now**.

The archive generates and is saved to your default download folder. The file name is created with the following syntax:

`backup_year-month-day`

For example:

`backup_2018-10-20`

Move the backup to your storage directory.

Managing Backup Tasks

Use the REST API to create a backup and automate system backup tasks.

You can use the Backup API to generate backups via any valid HTTP request. The syntax requires authentication using user credentials.

Restoring the CT-VL from a Backup

If a complete system restore is required, make sure the CT-VL installation is complete before beginning this task.

Note

Make sure the CT-VL version leveraged for the backup is the same as the backup or newer than the backup and is at least version 2.2.2.

Use the following steps to restore the CT-VL to its last recorded working state using a backup:

1. Select **Backup** from the administration interface left hand navigation.

2. Click the **Restore** tab.
3. Click **Choose File** and navigate to the required backup archive.
4. Confirm the following back up details:
 - CT-VL version
 - Backup date
 - Encryption key name
5. Click **Restore**.

Note

Depending on the size of the backup, the restore process may take some time.

If this node is a member of an existing cluster, it is removed from the cluster. Any existing data on this node is destroyed.

A new cluster is created with this node as the first node but with data restored from the backup.

When the restore completes, this node has all data from the backup including login credentials.

Note

If the user forgets the login credentials in the backup, run the `createsuperuser` CLI command to create a new user.

This node becomes a new node of a new cluster. It is no longer a member of any previous cluster.

Note

If this node was a member of an existing cluster prior to the restore, it must be removed from that cluster. Failure to remove this node from its previous cluster prevents new nodes from joining that cluster.

The CT-VL is restored to the state of archive. When the task is complete, a banner appears, indicating that the restore is complete. The CT-VL reboots.

When the login screen appears, enter your credentials to access the restored application.

Key Rotation

Regular encryption key rotation can increase security and also satisfy some PCI DSS requirements. To rotate CT-VL keys, ask your CT-VL Administrator to run the CT-VL CLI command `security gencert` on the CM that creates and stores the CT-VL keys.

Versioned keys are not permitted for tokenization.

Upgrading CT-VL

You can upgrade with the standard upgrade ZIP file only if you are upgrading from CT-VL v2.3 or higher. If you are upgrading from any CT-VL version earlier than v2.3, refer to the release notes for specific details based on the current version you are upgrading from.

If you are upgrading from CT-VL version 2.2.x or earlier, you must perform a backup and restore procedure as part of your upgrade. The backup includes users and groups, permissions, and settings. You can perform this backup either through the CT-VL GUI or REST APIs.

Refer to [Backing Up and Restoring CT-VL](#) for instructions on this backup step, and on the subsequent restore step.

Notes for Upgrading a Cluster

- All nodes in the cluster must be upgraded to the same CT-VL version. A CT-VL cluster must not operate nodes using different CT-VL versions.
- When upgrading nodes in a cluster, all CT-VL operations must be suspended. If this is not possible, you must perform a rolling upgrade. The nodes to be upgraded must be taken offline, for example, taken out of the load balancer before an upgrade is performed.
When the load balancer is ready to use the upgraded nodes, all other nodes not yet upgraded must be taken offline.
At any one time, the load balancer must not serve nodes running different CT-VL versions.
- While a node upgrade is in progress, it must not be interrupted. If it is interrupted or does not complete successfully, you must discard the node by removing it from the cluster.
Create a new VM, upgrade it to the desired version, and rejoin it to the cluster. When joining the node to the cluster, it must have the same version as all other nodes in the cluster.
- All nodes in the cluster must be upgraded, and must complete the upgrade before resuming operations.

Upgrades to AWS Instances Cannot Use SSH Key Pairs

Upgrading an EC2 instance from CT-VL v2.5 or earlier retains the use of passwords for the cliadmin instead of SSH key pairs. Only newly-created instances from CT-VL v2.6 and above can use the SSH key pair in AWS.

Upgrade Steps

If you have multiple nodes in a cluster, you can upgrade all nodes at the same time.

1. Obtain the upgrade ZIP file from your Thales sales or support representative, and place the file on a server

that is accessible from CT-VL over either HTTP or FTP.

- Although the upgrade ZIP file is encrypted and digitally signed, the HTTP server can use SSL (HTTPS) and basic authentication. The HTTPS server can also use a self-signed certificate.
- Log in as `cliadmin` to the CT-VL virtual machine that you want to upgrade. In the following example command, substitute your own CT-VL hostname.

```
$ ssh -l cliadmin <Tokenization Server hostname>
cliadmin@<Tokenization_Server_IP_address> password: <Enter the cliadmin
password>
[...]
CipherTrust Tokenization Server Main Menu
main> ?
Command                Description
=====
auth                    Authentication Setup
cluster                 Cluster Setup
icapi                   ProtectApp Setup
network                 Network Setup
system                  System Setup
vae                     VAE Configuration
vts                     Tokenization Setup
quit|q|up|<ctrl-d>      Quit or return to previous menu
exit                    Exit application

Enter <command> to display usage
main>
```

- At the main menu, enter:

```
main> vts upgrade --url <upgrade_zip_url>
```

Another option is to navigate to the vts submenu and enter the upgrade command directly:

```
main> vts
vts> upgrade --url <upgrade_zip_url>
```

Examples:

```
upgrade https://example.com/ctvl-upgrade-<version>.zip
upgrade https://username:password@example.com/ctvl-upgrade-<version>.zip
upgrade ftp://example.com/ctvl-upgrade-<version>.zip
```


Watch the output of the **upgrade** command while the upgrade is underway. After several progress messages, you should see a success message like the following:

```
Upgrade complete.
```

Depending on the type of upgrade, the system will be rebooted at the end for changes to take effect. You are prompted at the beginning of the upgrade for confirmation.

There are other options to perform an upgrade. Run **upgrade --help** for details.

Monitoring Through SNMP

You can monitor disk partitions, files, load average, and processes by setting up SNMP polling. Configure SNMP at the CLI in the **network** category and the **snmp** command.

When enabling SNMP, you must specify a whitelist of IP addresses for SNMP managers. CIDR control access is supported.

CT-VL supports both SNMP v3 and v2c. SNMP v3 supports password authentication, encryption, and a choice of encryption and password hashing algorithms.

Note

SNMP traps are not supported.

Supported MIB OIDs

Below are some MIB OIDs that CT-VL supports:

CPU Load Average Monitoring

```
1 minute load average  .1.3.6.1.4.1.2021.10.1.3.1
5 minute load average  .1.3.6.1.4.1.2021.10.1.3.2
15 minute load average .1.3.6.1.4.1.2021.10.1.3.3
```

System Uptime

```
.1.3.6.1.2.1.1.3.0
```

Memory Statistics

```
Total Memory      .1.3.6.1.4.1.2021.4.5.0
Total Memory Used  .1.3.6.1.4.1.2021.4.6.0
Total Memory Free  .1.3.6.1.4.1.2021.4.11.0
Total Swap Space   .1.3.6.1.4.1.2021.4.3.0
```

```
Total Availabe Swap .1.3.6.1.4.1.2021.4.4.0
```

Disk Statistics:

```
Total size of disk partition .1.3.6.1.4.1.2021.9.1.6.1
Total disk space used .1.3.6.1.4.1.2021.9.1.8.1
Total available disk space .1.3.6.1.4.1.2021.9.1.7.1
```

Examples

Below are examples using the `snmpwalk` command.

Get Total Memory on CT-VL VM

```
snmpwalk -v 2c -c mystring ctvl-ip .1.3.6.1.4.1.2021.4.5.0
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 16431432 kB
```

Get System Uptime

```
snmpwalk -v 2c -c mystring ctvl-ip .1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (22329) 0:03:43.29
```

Note

- SNMP on CT-VL is disabled by default. Use the CLI to enable access.
- SNMP manager(s) must be set to allow access to CT-VL.

Configuring SNMP

1. To configure SNMP, use the `network snmp` CLI command.

```
main> network snmp
usage: snmp command [options]SNMP Configurationcommands:
  user          SNMPv3 user setup
  community     SNMPv2c community access. Do not set if you don't want to enable SNMPv2
  monitor       Monitor Services Setup (processes, load average, disk space, and file size)
  manager       SNMP Managers Setup
  service       Enable, disable, and show SNMP Service
```

system SNMP System Information Setup

2. Create a user.

```
main> network snmp user
usage: user [-h] [--show] [--add USERNAME] [--delete USERNAME [USERNAME ...]]
          [--authpass PASSWORD] [--privpass PASSWORD] [-x {DES,AES}]
          [-a {MD5,SHA}]
```

3. Create a community string.

```
main> network snmp community
usage: community [-h] [--show] [--add STRING] [--delete STRING] [-y]
```

4. Enable SNMP as SNMP is disabled by default.

```
main> network snmp service --enable
```

5. Set SNMP managers who will access CT-VL.

```
main> network snmp manager
usage: server [-h] [--show] [--add IP [IP ...]] [--remove IP [IP ...]]
```

6. Monitor disk, file, load, and processes.

```
main> network snmp monitor
usage: monitor [-h] [--show] [--set {disk,file,load,proc}]
          [--delete {disk,file,load,proc}] [--value VALUE]
```

Examples

This section contains the following pages:

- [Cryptographic Operation Examples](#)
- [Key Management Examples](#)
- [Versioned Key Management Examples](#)
- [Tokenization Examples](#)

Cryptographic Operation Examples

This section shows the examples of the CipherTrust cryptographic operation along with the examples related to encryption, decryption, encrypting and decrypting large files, sign, verify, and random.

A user or group may perform each of the following cryptographic operations using a specific key if he or she has permissions to perform such as configured for each individual key from inside the CT-VL Administration UI.

The allowed cryptographic operations available are:

- **Encrypt** - Encryption using a key.
- **Decrypt** - Decryption using a key.
- **Sign** - Sign with a given key.
- **Verify** - Verify a signature with a given key.

Cryptographic Headers

Cryptographic headers is a convenient feature that allows a ciphertext to be decrypted without knowledge of the specific key name and/or version that was used to encrypt it. When encrypting with headers enabled, the ciphertext includes bytes containing key information inserted before the encrypted data.

To enable cryptographic headers, the `header` attribute must appear in the POST body for sign, verify, encrypt, and decrypt calls.

Verify and decrypt also take an `all` header attribute value to let the service figure out the header version in a more transparent way for the caller.

CipherTrust Cryptographic Operation Examples

Simple SHA-256 Digest Example

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/crypto/v1/digest' \
-d '{"alg": "S256", "payload": "Ef+4MkU6ukzQSMbHxH4rxoVF/xKHyg=="}' \
-H 'content-type: application/json'
```

Response

```
{
  "digest": "RfdwuYKwBn0l03ZMGB7r59S+P+T9/s8ufPJfcSQXNfE="
}
```

Encryption Examples

AES-256 CBC Encryption Example

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/crypto/v1/encrypt' \
-d '{"plaintext": "JLns9BI4Pa7TTrCaeJIjQQ==", "alg": "A256CBC", "params": {"iv":
"tler+lrCnchFRJDCFs1F/A=="}, "kid": "mykey123"}' \
-H 'content-type: application/json'
```

Response

```
{
  "ciphertext": "SZASu6Ts+01IMtb14+djhg==",
  "params": {}
}
```

Simple AES-256 CTR Encryption with Headers Example

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/crypto/v1/encrypt' \
-d '{"plaintext": "JLns9BI4Pa7TTrCaeJIjQQ==", "alg": "A256CTR", "params":{"iv":
"tler+lrCnchFRJDCFs1F/A=="}, "kid": "myversionedkey", "headers"="hdr_v2.7"}' \
-H 'content-type: application/json'
```

Response

```
{
  "ciphertext": "UktNQzIxMAD/////AAAABXV1awQAAAAAEDkKfAiGhjeLkp50NDgjpQT/////AA
AAA2l2AAAAABhYmMxMjNkZWZnaDY3ODkw/////wAAAAVjc3VtAAAAACAbtDVcSesXu7SYqoZxNR+hC6C
q8H3gE5xv5vvSzRhUrKcwnBu8",
  "params": {}
}
```

Decryption Examples

Simple AES-256 CBC Encryption Example

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/crypto/v1/decrypt' \
-d '{"ciphertext": "wInPTVRugkxP0KBm46yCo8lpQMaDDRo15bEWt9Mz629MKh7IqB9roh36ZkyIu"
```

```
mLi", "alg": "A256CBC", "params": {"iv": "ek5Lmcw0JATvux5PoSaZ6w=="}, "kid": "foo
key"}' \
-H 'content-type: application/json'
```

Response

```
{
  "plaintext": "40idw01cBKKcF1tK4ZBdZg7yDe9KbvBP1npfs382HT+yz0Fb3CeCh/eBIhH7a1r
b",
  "params": {}
}
```

Simple AES-256 CTR Decryption with Headers Example

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/crypto/v1/decrypt' \
-d '{"ciphertext": "UktNQzIxMAD/////AAAAABXV1awQAAAAAEDkKfAiGhjeLkp50NDgjPQT/////A
AAAA2l2AAAAABBBhYmMxMjNkZWZnaDY3ODkw////wAAAAVjc3VtAAAAACAbtDVcSesXu7SYqoZxNR+hC6
Cq8H3gE5xv5vvSzRhUrKcwnBu8", \
"alg": "A256CTR", "params":{"iv": "tler+lrCnchFRJDCFs1F/A=="}, "headers"="all"}' \
-H 'content-type: application/json'
```

Response

```
{
  "plaintext": "JLns9BI4Pa7TTrCaeJIjQQ==",
  "params": {}
}
```

Encrypting and Decrypting Large Files

To handle files that are larger than the payload limit of 5 MB, use the following script to automatically break down a large file into a chain of smaller files for encryption, and to join these files and decrypt them:

```
#!/bin/bash
URL=http://localhost:8000/crypto/v1
KEY=cc01
IV="MTAwMDIwMDAzMDAwNDAwMA=="
CHUNKSIZE=256
FILE="/etc/hosts"
ALG="A256CTR"
BASE=`basename $FILE`
OIV=$IV
if [ ! -e ./ $BASE ]
then
cp $FILE .
```

```

fi
rm -f ${BASE}.* *-${BASE}.*
split -b $CHUNKSIZE ./${BASE} ${BASE}-
NOMORE=0
echo "===== Encryption ====="
for i in ${BASE}.*
do
if [ $NOMORE -eq 1 ]
then
break
fi
PTEXT=`cat $i | base64 -w 0`
PAYLOAD=$(jq -n -r \
--arg kid $KEY \
--arg iv $IV \
--arg pt "$PTEXT" \
--arg alg $ALG \
'{ plaintext: $pt, alg: $alg, kid: $kid, params: {iv: $iv} }'
)
RESULT=`echo $PAYLOAD | curl -Ss -X POST -k -d @- -H "Accept: application/json"
$URL/encrypt`
NEWIV=`echo $RESULT | jq -r '.params.iv'`
CTEXT=`echo $RESULT | jq -r '.ciphertext'`
if [ -z "$NEWIV" ] || [ "$NEWIV" == "null" ]
then
NOMORE=1
fi
IV=$NEWIV
echo -n $CTEXT | base64 -d -w0 > ${i}.enc
done
cat ${BASE}.*.enc > ./${BASE}.enc
split -b $CHUNKSIZE ./${BASE}.enc enc-${BASE}-
IV=$0IV
NOMORE=0
echo "===== Decryption ====="
for i in enc-${BASE}.*
do
if [ $NOMORE -eq 1 ]
then
break
fi
CTEXT=`cat $i | base64 -w0`
PAYLOAD=$(jq -n -r \
--arg kid $KEY \
--arg iv $IV \
--arg ct "$CTEXT" \
--arg alg $ALG \
'{ ciphertext: $ct, alg: $alg, kid: $kid, params: {iv: $iv} }'
)
RESULT=`echo $PAYLOAD | curl -Ss -X POST -k -d @- -H "Accept: application/json"
$URL/decrypt`
NEWIV=`echo $RESULT | jq -r '.params.iv'`
PTEXT=`echo $RESULT | jq -r '.plaintext'`
if [ -z "$NEWIV" ] || [ "$NEWIV" == "null" ]
then
NOMORE=1
fi
IV=$NEWIV
echo -n $PTEXT | base64 -d -w0 > ${i}.dec

```

```
done
cat enc-`${BASE}`-*.dec > `${BASE}`.dec
diff `${BASE}`.dec `${BASE}`
echo diff status = $?
```

The large file is broken into chunks of size whose modulo 16 equals 0 (that is, into chunk sizes that are multiples of 16 bytes).

When submitting the chunks to the encryption endpoint, the response includes an IV that is used to encrypt the next chunk. If the payload length is not 16-byte aligned, the response does not return the next IV, assuming that chaining is complete.

For decryption, the same rules apply. The first call must use the same IV as the call used for encrypting. The ciphertext chunk size can be a different size from the original plaintext encryption request, as long as the chunks are 16-byte aligned.

For example a 1030 byte file can be encrypted in 8 chunks of 128 bytes each and a last one of 6 bytes. The same file once encrypted can be split in 3 chunks of 320 bytes and a last one of 70 bytes.

Sign

Simple HMAC-SHA256 Sign Example

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/crypto/v1/sign' \
-d '{"alg": "HS256", "payload": "aGVsbG8gd29ybGQh", "kid": "mykey"}' \
-H 'content-type: application/json'
```

Response

```
{
  "signature": "13pCxjIXjY8MNhsx69szyU0VraQc6Tn0C0irqCut0Ls="
}
```

Verify

Simple HMAC-SHA256 Sign Example

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/crypto/v1/verify' \
-d '{"alg": "HS256", "signature": "13pCxjIXjY8MNhsx69szyU0VraQc6Tn0C0irqCut0Ls=",
  "payload": "aGVsbG8gd29ybGQh", "kid": "mykey"}' \
-H 'content-type: application/json'
```


Response

```
{
  "valid": 1
}
```

Random

Simple Random Example

The following example shows a simple request to generate an 8-byte random number:

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/crypto/v1/random' \
-d '{"seed": "q8qqyw==", "size": 8}' \
-H 'content-type: application/json'
```

Response

```
{
  "random": "152LM5GzL8U="
}
```

Authorization Operations

Below table shows the possible authorization operations when a key is / was created or imported via the API. The owner is the user who creates the key.

Operation	Owner	Group	Other User	Other Group
Create	Yes, if owner has create permission Owner can specify on create the permissions allowed though the usage attribute If usage attribute is not supplied, then key will be created with all permissions	Yes, if owner has create permission granted through a group Owner can specify on create the permissions allowed though the usage attribute If usage attribute is not supplied, then key will be created with all permissions	N/A	N/A
Import	Yes, if owner has import permission	Yes, if owner has import permission	N/A	N/A

Operation	Owner	Group	Other User	Other Group
	Owner can specify on import the permissions allowed though the usage attribute If usage attribute is not supplied, then key will be imported with all permissions	granted through a group Owner can specify on import the permissions allowed though the usage attribute If usage attribute is not supplied, then key will be imported with all permissions		
Destroy	Yes, if owner gave permission on creation	No	No	No
Export	Yes, if owner gave permission on import or create	No	No	No
Find	Yes, if owner gave permission on import or create	No	No	No
Modify	Yes, if owner gave permission on import or create	No	No	No
Encrypt	Yes, if owner gave permission on import or create	No	No	No
Decrypt	Yes, if owner gave permission on import or create	No	No	No
Sign	Yes, if owner gave permission on import or create	No	No	No
Verify	Yes, if owner gave permission on import or create	No	No	No

Key Management Examples

This section shows the examples of the CipherTrust key management as listed below:

- Key States
- Creating and Importing Keys
- Managing Keys

-
- Valid Key Operations
 - Key Creation
 - Key Import
 - Key Delete
 - Key Update
 - Key Retrieval
 - Key Export
 - Wrapping Keys
 - Importing with Wrapping Key
 - Exporting with Wrapping Key
 - Importing and Exporting Using Clear Bytes

Key States

Symmetric and asymmetric keys support the following states:

- **Preactive** - When a key was created to have a preactive state.
- **Active** - Default when a key is created.
- **Suspended** - A key has been suspended.
- **Compromised** - Key is compromised.
- **Deactivated** - Key is deactivated.
- **Destroyed** - Key is destroyed.

Note

- To set a key to the suspended, compromised, or deactivated state, call the modify endpoint (**PATCH**).
- A key can be set to the active state again only if it is in the suspended state.

Creating and Importing Keys

This category covers both **key import** and **key creation**. A user or group that has import, create, or both

permissions is allowed to add keys to CT-VL.

The difference between import from create is in the ability to add a wrapped key representing the key itself to the key creation request. This ability to create and import keys is set for each user through the CT-VL Administration UI.

Note

- If a user or group has create and/or import permissions, invoking the corresponding API endpoint will create a key with all the permissions for CipherTrust key management and cryptographic operations.
- Any other user needing to perform operations on such keys, can only be added through the CT-VL Administration UI.

If the user does not have permission to create or import, the corresponding creation endpoint will return an error.

Managing Keys

A user or group may perform each of the following key management operations if he or she has permissions to perform such. This can be configured for each individual key from inside the CT-VL Administration UI.

The allowed key management operations are:

- **Destroy** - Destroy a key.
- **Modify** - Modify a key's attributes.
- **Find** - Look up or retrieve a specific key.
- **Export** - Export a specific key. This only applies to symmetric keys.

Valid Key Operations

Below table lists the valid key operations from the API. It does not include permissions related to authorization. See the [Authorization Operations](#) section for more details.

Operation	Symmetric Key	Versioned Symmetric Key	Asymmetric Key	Opaque Object
Create	yes	yes	yes	yes
Import	yes	yes	no	yes
Destroy	Key delete	Delete key by UUID or MUID only	Key delete	Key delete
	custom attribute delete	Delete key of version 0 deletes entire set	custom attribute delete	custom attribute delete

Operation	Symmetric Key	Versioned Symmetric Key	Asymmetric Key	Opaque Object
		Delete custom attribute		
Modify	custom attributes add alias key states key migration to versioned key	custom attributes add alias key states key rotation lifespan interval	custom attributes	custom attributes add alias
Find	By label/name By UUID By MUID By alias	By label/name - last version By UUID By MUID By alias	public key only By label/name	By label/name By UUID By MUID By alias
Export	clear_key kid for plain bytes wrapping key with A256CBC or A256CBCPAD	clear_key kid for plain bytes wrapping key with A256CBC or A256CBCPAD	no	no
Encrypt	Standard encryption cryptographic headers	Standard encryption cryptographic headers	Standard encryption only	no
Decrypt	Standard decryption cryptographic headers	Standard decryption cryptographic headers	Standard decryption only	no
Sign	HMAC signing cryptographic headers	HMAC signing cryptographic headers	RSA signing only	HMAC signing cryptographic headers
Verify	HMAC verifying cryptographic headers	HMAC verifying cryptographic headers	RSA verifying only	HMAC verifying cryptographic headers

Key Creation

Key creation allows to create a key whose contents (key material or bytes) is generated by the CM.

Note

- You cannot create a key via the REST API and make it a seed key in one step.
- You must create the key and add it via the CT-VL GUI in separate steps.
- You cannot just pass in **seedkey** as a keyword to the create key REST API. If you do, the key is created, but it is not a seed key.

This example shows how to create an AES 128-bit key named foo123key:

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/km/v1/keys' \
-d '{"size": 128, "name": "foo123key", "kty": "oct"}' \
-H 'content-type: application/json'
```

Response

```
{
  "attributes": {
    "usage": [
      "encrypt",
      "decrypt",
      "sign",
      "verify",
      "destroy",
      "modify",
      "find",
      "export"
    ],
    "size": 16
  },
  "metadata": {
    "key_management": {
      "uuid": "9f3c2775-405b-33e8-91af-904faa0e551f",
      "caching_duration": 44640,
      "state": "active",
      "lifespan_unit": "days",
      "muid": "9f3c2775-405b-33e8-91af-904faa0e551f39b6aee0-6821-39f9-a186-9a9904abff32",
      "caching_enabled": true,
      "name": "foo123key"
    }
  },
  "kid": "tesrn:vts::label:foo123key"
}
```

Key Import

Key import is basically the same as a key create, however the request contains information on the **wrapped** key and the **wrapping** key.

This request imports an AES 128-bit key named foo1234key with key bytes equal to: **"HelloWorldAlways"**.

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/km/v1/keys' \
-d '{"size": 128, "name": "foo123key", "kty": "oct", "wrappedkey": "SGVsbG9Xb3JsZEFsd2F5cw=="}' \
-H 'content-type: application/json'
```

Response

```
{
  "attributes": {
    "usage": [
      "encrypt",
      "decrypt",
      "sign",
      "verify"
    ],
    "size": 16
  },
  "metadata": {
    "key_management": {
      "uuid": "9f3c2775-405b-33e8-91af-904faa0e551f",
      "caching_duration": 44640,
      "state": "active",
      "lifespan_unit": "days",
      "muid": "9f3c2775-405b-33e8-91af-904faa0e551f39b6aee0-6821-39f9-a186-9a9904abff32",
      "caching_enabled": true,
      "name": "foo123key"
    }
  },
  "kid": "tesrn:cts::label:foo1234key"
}
```

Key Delete

This example deletes a key named foo123key.

Note

Upon success, the HTTP status code must be 200.

Request

```
curl -k -u vtsuser -X DELETE 'https://127.0.0.1/vts/km/v1/keys/foo123key'
```

Response

```
200 OK
```

Key Update

Request

```
curl -k -u vtsuser -X PATCH 'https://127.0.0.1/vts/km/v1/keys/fookey' \
-d '{"state": "suspended"}' \
-H 'content-type: application/json'
```

Response

```
{
  "attributes": {
    "usage": [
      "encrypt",
      "decrypt",
      "sign",
      "verify",
      "destroy",
      "find"
    ],
    "size": 16
  },
  "metadata": {
    "key_management": {
      "uuid": "bfd2a7b7-20db-31d7-b22f-01e625a54961",
      "caching_duration": 44640,
      "state": "suspended",
      "versioning_enabled": 1,
      "lifespan_unit": "days",
      "muid": "bfd2a7b7-20db-31d7-b22f-01e625a549618b0ce745-fcb2-3718-97b3-d4a0b415ac5d",
      "caching_enabled": 1,
      "name": "fookey"
    }
  },
  "kid": "tesrn:cts::label:fookey"
}
```


Key Retrieval

Request

```
curl -k -u vtsuser -X GET 'https://127.0.0.1/vts/km/v1/keys/fookey'
```

Response

```
{
  "attributes": {
    "usage": [
      "encrypt",
      "decrypt",
      "sign",
      "verify",
      "destroy",
      "find"
    ],
    "size": 16
  },
  "metadata": {
    "key_management": {
      "uuid": "bfd2a7b7-20db-31d7-b22f-01e625a54961",
      "caching_duration": 44640,
      "state": "suspended",
      "versioning_enabled": 1,
      "lifespan_unit": "days",
      "muid": "bfd2a7b7-20db-31d7-b22f-01e625a549618b0ce745-fcb2-3718-97b3-d4a0b415ac5d",
      "caching_enabled": 1,
      "name": "fookey"
    }
  },
  "kid": "tesrn:cts::label:fookey"
}
```

Key Export

Request

```
curl -k -u vtsuser -X POST 'https://127.0.0.1/vts/km/v1/keys/export/fookey' \
-d '{"alg": "A256CBCPAD", "params": {"iv": "RERERERERERERERERERERA=="}, "kid": "tesrn:cts::label:foo123key"}' \
-H 'content-type: application/json'
```

Response

```
{
  "exported_key": {
```

```

    "attributes": {
      "usage": [
        "encrypt",
        "decrypt",
        "sign",
        "verify",
        "destroy",
        "find"
      ],
      "size": 16
    },
    "metadata": {
      "key_management": {
        "uuid": "a0415d15-74d5-3d90-b755-53d88b6b2c31",
        "caching_duration": 44640,
        "state": "active",
        "lifespan_unit": "days",
        "muid": "a0415d15-74d5-3d90-b755-53d88b6b2c315276f6c8-ecde-30aa-8
7b9-17d63a432d94",
        "caching_enabled": 1,
        "name": "fookey"
      }
    },
    "kid": "tesrn:cts::label:fookey"
  },
  "wrappedkey": "pXSyQ1eGDEucWA1Mvmwix5tTGezs73Hr92hecKtn5Bc="
}

```

Wrapping Keys

When importing or exporting a key, it is recommended to use a wrapping key in order not to compromise the key itself.

Note

In the current version of the CT-VL Key Management module, a wrapping key must be an AES 256-bit key, and only the CBC or CBCPAD ciphers may be used.

The ideal method for wrapping a key on both import and export is to use asymmetric keys.

Importing and exporting key requires that the user has a wrapping key, and such key is available on both client and the CT-VL.

Importing with Wrapping Key

When importing with a wrapping key, the following steps need to be taken:

1. Create a 256-bit AES key that is known to the caller and the CT-VL. This is the wrapping key.
2. Encrypt the key bytes with the wrapping key using an AES-CBC or AES-CBCPAD cipher. This is the wrapped key.

3. Call the create endpoint and supply the base64-encoded wrapped key and the wrapping key ID.

Exporting with Wrapping Key

When exporting with a wrapping key, the following steps need to be taken:

1. Create a 256-bit AES key that is known to the caller and the CT-VL. This is the wrapping key.
2. Call the export endpoint, supplying a wrapping key ID. This will return a wrapped key.
3. Decrypt the base64-encoded wrapped key using the wrapping key on the client side.

Importing and Exporting Using Clear Bytes

The API offers the ability to perform import and export functions using raw key bytes, however such method is not recommended.

To import a key using clear bytes, omit the wrapping key ID or give the `clear_key` label. For exporting, give the `clear_key` label as the wrapping key ID. In both cases, the `alg` and the `iv` attributes must be omitted.

Versioned Key Management Examples

This section shows how to use versioned keys within the CipherTrust Cryptographic and Key Management set of APIs.

Versioned Key Creation Example

Creation of an AES 256-bit versioned key named `vkey256` with each version of the key with a lifespan of 7 days.

Request

```
POST /vts/km/v1/keys
{"size": 256, "name": "vkey256", "kty": "oct", "versioning_enabled": true, "version": 0, "version_action": "create", "lifespan_interval": 7}
```

Response

```
200 OK
{"attributes": {"usage": ["encrypt", "decrypt", "sign", "verify", "destroy", "modify", "find", "export"], "size": 16}, "metadata": {"key_management": {"uuid": "9f3c2775-405b-33e8-91af-904faa0e551f", "caching_duration": 44640, "state": "active", "lifespan_unit": "days", "muid": "9f3c2775-405b-33e8-91af-904faa0e551f39b6aee0-6821-39f9-a186-9a9904abff32",
```

```
"caching_enabled": True,
"name": "vkey256"}},
"kid": "tesrn:vts::label:vkey256"}
```

Versioned Key Import Example

Creation of an AES 256-bit versioned key named `vkey256` with each version of the key with a lifespan of 7 days.

Request

```
POST /vts/km/v1/keys
{"size": 256, "name": "vkey256", "kty": "oct", "versioning_enabled":
true, "version": 0, "lifespan_interval": 7, "wrappedkey":
"SGVsbG9Xb3JsZEFsd2F5cw=="}
```

Response

```
200 OK
{"attributes": {"usage": ["encrypt", "decrypt", "sign", "verify",
"destroy", "modify", "find", "export"],
"size": 16},
"metadata": {"key_management": {"uuid": "9f3c2775-405b-33e8-91af-
904faa0e551f",
"caching_duration": 44640,
"state": "active",
"lifespan_unit": "days",
"moid": "9f3c2775-405b-33e8-91af-904faa0e551f39b6aee0-6821-39f9-
a186-9a9904abff32",
"caching_enabled": True,
"name": "vkey256"}},
"kid": "tesrn:cts::label:vkey256"}
```

Standard Key to Version Key Migration Example

Creation of an AES 256-bit versioned key named `vkey256` with each version of the key with a lifespan of 7 days.

Request

```
PATCH /vts/km/v1/keys/existingkey {"version_action": "migrate"}
```

Response

```
200 OK
{
"attributes": {
"size": 32
},
}
```

```

"metadata": {
  "key_management": {
    "uuid": "7f460ca4-04e2-37d9-8d57-b2c36fbed55e",
    "caching_duration": 44640,
    "state": "active",
    "versioning_enabled": true,
    "muid": "7f460ca4-04e2-37d9-8d57-b2c36fbed55ecd222707-123a-3df5-baef-c9b1f77ac3fb",
    "caching_enabled": 1,
    "name": "existingkey"
  }
},
"kid": "tesrn:cts::uuid:7f460ca4-04e2-37d9-8d57-b2c36fbed55e"
}

```

Version Key Rotation Example

Rotation of an existing versioned key named `vkey256` with each version of the key with a lifespan of 7 days.

Request

```
PATCH /vts/km/v1/keys {"version_action": "rotate"}
```

Response

```

200 OK
{"attributes": {"usage": ["encrypt", "decrypt", "sign", "verify",
"destroy", "modify", "find", "export"],
"size": 16},
"metadata": {"key_management": {"uuid": "d0a031de-5102-3bba-80f4-7019306da03e",
"caching_duration": 44640,
"state": "active",
"lifespan_unit": "days",
"muid": "d0a031de-5102-3bba-80f4-7019306da03ed0a031de-6821-39f9-a186-9a9904abff32",
"caching_enabled": True,
"name": "vkey256"}},
"kid": "tesrn:cts::label:vkey256"}

```

Tokenization Examples

Java Client Example

```

package com.vormetric.app;

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;

```

```

import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;

import org.apache.commons.codec.binary.Base64;

import com.jayway.jsonpath.*;

import javax.net.ssl.HttpURLConnection;
import javax.net.ssl.SSLSession;
import javax.net.ssl.SSLSocket;
import javax.net.ssl.SSLSocketFactory;

public class CTSCClientSample {

    public static void main(String[] args) {
        new CTSCClientSample().DoIt();
    }
    @SuppressWarnings("null")
    private void DoIt(){
        String credential = Base64.encodeBase64String("ctsuser1:Panda45".getBytes());
        String ccn[] = {
            "5471949763376677", "5545127221796024", "5353800637453171", "5139918930790601",
            "5267003853942275"
        };
        try {

            // Tokenize request
            String https_url = "https://ipaddress/vts/rest/v2.0/tokenize/";
            URL myurl = new URL(https_url);
            HttpURLConnection con = (HttpURLConnection)myurl.openConnection();
            String ccNum = "9453677629008564";
            String jStr =
                "{\"data\": \""+ccNum+"\", \"tokengroup\": \"t1\", \"tokentemplate\": \"Credit Card\"}";
            con.setRequestProperty("Content-length", String.valueOf(jStr.length()));
            con.setRequestProperty("Content-Type", "application/json");
            con.setRequestProperty("Authorization", "Basic "+credential);
            con.setRequestMethod("POST");
            con.setDoOutput(true);
            con.setDoInput(true);
            DataOutputStream output = new DataOutputStream(con.getOutputStream());
            output.writeBytes(jStr);
            output.close();
            BufferedReader rd=new BufferedReader(new InputStreamReader(con.getInputStream()));
            String line = "";
            String strResponse = "";
            while ((line = rd.readLine()) != null) {
                strResponse=strResponse+line;
            }
            rd.close();
            String token = JsonPath.read(strResponse, "$.token").toString();
            con.disconnect();
            System.out.println("Tokenize server: "+https_url);
            System.out.println("Tokenize request: "+jStr);
            System.out.println("Tokenize response: "+strResponse);

            // Bulk tokenize

```

```

String jStrArray = "[";
for (int i=0;i<ccn.length;i++) {
jStrArray = jStrArray +
"{\"data\": \"\"+ccn[i]+\"\", \"tokengroup\": \"t1\", \"tokenemplate\": \"Credit
Card\"}";
if (i<ccn.length-1) {
jStrArray = jStrArray + ",";
}
}
jStrArray=jStrArray+"]";

con = (HttpsURLConnection)myurl.openConnection();
con.setRequestProperty("Content-length", String.valueOf(jStrArray.length()));
con.setRequestProperty("Content-Type", "application/json");
con.setRequestProperty("Authorization", "Basic "+credential);
con.setRequestMethod("POST");
con.setDoOutput(true);
con.setDoInput(true);
output = new DataOutputStream(con.getOutputStream());
output.writeBytes(jStrArray);
output.close();
rd = new BufferedReader(new InputStreamReader(con.getInputStream() ));
line = "";
strResponse = "";
while ((line = rd.readLine()) != null) {
strResponse=strResponse+line;
}
rd.close();
System.out.println("Bulk Tokenize request: "+jStrArray);
System.out.println("Bulk Tokenize response: "+strResponse);
String[] tokens = new String[5];
for (int i=0;i<5;i++) {
tokens[i]=JsonPath.read(strResponse, "$["+i+"].token").toString();
}
// Detokenize request
https_url = "https://ipaddress/vts/rest/v2.0/detokenize/";
myurl = new URL(https_url);
con = (HttpsURLConnection)myurl.openConnection();
jStr="{\"token\": \"\"+token+\"\", \"tokengroup\" :
\"t1\", \"tokenemplate\": \"Credit Card\"}";
con.setRequestProperty("Content-length", String.valueOf(jStr.length()));
con.setRequestProperty("Content-Type", "application/json");
con.setRequestProperty("Authorization", "Basic "+credential);
con.setRequestMethod("POST");
con.setDoOutput(true);
con.setDoInput(true);
output = new DataOutputStream(con.getOutputStream());
output.writeBytes(jStr);
output.close();
rd = new BufferedReader(new InputStreamReader(con.getInputStream() ));
line = "";
strResponse = "";
while ((line = rd.readLine()) != null) {
strResponse=strResponse+line;
}
rd.close();
con.disconnect();
System.out.println("Detokenize server: "+https_url);
System.out.println("Detokenize request: "+jStr);

```

```

System.out.println("Detokenize response: "+strResponse);

// Bulk DeTokenize
jStrArray = "[";
for (int i=0;i<ccn.length;i++) {
jStrArray = jStrArray + "{\"token\":\""+tokens[i]+"\", \"tokengroup\" : \"t1\", \"tokentemplate\": \"Credit Card\"}";
if (i<ccn.length-1) {
jStrArray = jStrArray + ",";
}
}
jStrArray=jStrArray+"]";

con = (HttpsURLConnection)myurl.openConnection();
con.setRequestProperty("Content-length", String.valueOf(jStrArray.length()));
con.setRequestProperty("Content-Type", "application/json");
con.setRequestProperty("Authorization", "Basic "+credential);
con.setRequestMethod("POST");
con.setDoOutput(true);
con.setDoInput(true);
output = new DataOutputStream(con.getOutputStream());
output.writeBytes(jStrArray);
output.close();
rd = new BufferedReader(new InputStreamReader(con.getInputStream() ));
line = "";
strResponse = "";
while ((line = rd.readLine()) != null) {
strResponse=strResponse+line;
}
rd.close();
con.disconnect();
System.out.println("Bulk Detokenize request: "+jStrArray);
System.out.println("Bulk Detokenize response: "+strResponse);
} catch (MalformedURLException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
}
}
}

```

C# Client Example

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Linq;
using System.Text;
using System.Net;
using System.IO;
using System.Threading.Tasks;
using RestSharp;
using RestSharp.Authenticators;
using System.Windows.Forms;

```



```

using System.Security.Cryptography.X509Certificates;
using System.Net.Security;

namespace CTS2ClientSample
{
    class Program
    {
        static public bool MyRemoteCertificateValidationCallback(System.Object sender,
            X509Certificate certificate, X509Chain chain, SslPolicyErrors sslPolicyErrors)
        {
            return true; // unsafe! Do not do this in production!
        }

        static void tokenize(RestClient client)
        {
            var request = new RestRequest("vts/rest/v2.0/tokenize", Method.POST);
            request.RequestFormat = DataFormat.Json;
            request.AddBody(new { tokengroup = "t1", tokentemplate = "Credit Card",
                data = "1234567812345670" });
            // execute the request
            IRestResponse response = client.Execute(request);
            var responsestatus = response.ResponseStatus;
            var content = response.Content; // raw content as string

            MessageBox.Show("Tokenize API returns: " + content,
                responsestatus.ToString(), MessageBoxButtons.OK);
        }

        static void detokenize(RestClient client)
        {
            var request = new RestRequest("vts/rest/v2.0/detokenize", Method.POST);
            request.RequestFormat = DataFormat.Json;
            request.AddBody(new { tokengroup = "t1", tokentemplate = "Credit Card",
                token = "1234567812345670" });
            // execute the request
            IRestResponse response = client.Execute(request);
            var responsestatus = response.ResponseStatus;
            var content = response.Content; // raw content as string

            MessageBox.Show("Detokenize API returns: " + content, responsestatus.ToString(),
                MessageBoxButtons.OK);
        }

        static void Main(string[] args)
        {
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
            ServicePointManager.ServerCertificateValidationCallback =
                MyRemoteCertificateValidationCallback; // unsafe! Do not do this in production!

            var client = new RestClient("https://10.10.10.10");
            client.Authenticator = new HttpBasicAuthenticator("username",
                "password");

            tokenize(client);

            detokenize(client);
        }
    }
}

```

Advanced Topics

This section describes advanced CT-VL topics.

- [CT-VL CLI Reference](#)
- [CT-VL Key Cache](#)
- [CT-VL Services](#)
- [Key Types and Input Sizes](#)
- [Creating Static and Dynamic Keys](#)
- [Tokenization Groups, Templates, and Masks](#)
- [Understanding CT-VL Authentication and Authorization](#)
- [CT-VL CLI Navigation Summary and Guidelines](#)
- [Tokenization Setup Workflow in CT-VL GUI](#)
- [CT-VL Tokenization REST APIs](#)
- [CT-VL Admin REST APIs](#)
- [CT-VL with Multiple Key Managers](#)

CT-VL CLI Reference

The CipherTrust Vaultless Tokenization (CT-VL) Command Line Interface (CLI) is used to configure the CT-VL network and do other system-level tasks.

Listed below are the CT-VL CLI categories and the commands within each category:

- [Authentication Setup Category](#)
- [Cluster Management Category](#)
- [CipherTrust Manager Setup](#)
- [Network Management Category](#)
- [System Configuration Category](#)
- [CT-VL Configuration Commands](#)

Note

- This article provides a high-level overview of each command category. For complete usage details on each command and its options, run the command with the `--help` option.
- For tips on navigating the CLI, see [CLI Navigation Cheat Sheet](#).
- To access the CLI, see [Access the CLI through a Terminal Emulator like PuTTY](#).

Authentication Setup Category

The `auth` category contains authentication setup commands -- for OAuth2 (`oauth2` command), for setting and reviewing optional OAuth2 settings, and the `ldap` command, whose options configure and review the LDAP Server settings.

As an alternative to specifying the various LDAP settings as options to the `ldap` command (for example, `auth ldap --server ldap://hostname:636`), you can run an interactive wizard that prompts you for each setting. Run this wizard with the `ldap --setup` command.

Cluster Management Category

The `cluster` category is a subset of CT-VL commands for use with cluster management.

Below is the cluster management category commands table:

Command	Description
<code>add</code>	Add a node that can later create or be joined to an existing cluster. This command must be used on a node before that node can join a cluster. NOTE: Do not use with <code>remotejoin</code> .
<code>apiadmin</code>	API admin user setup, for automated cluster set up.
<code>create</code>	Create a CT-VL cluster.
<code>join</code>	Join a node to an existing CT-VL cluster. NOTE: Do not use with <code>remotejoin</code> .
<code>remotejoin</code>	Remotely join nodes to a cluster. Use in automated cluster setups.
<code>remove</code>	Remove a node from a cluster.
<code>show</code>	Show cluster settings.

CipherTrust Manager Setup

Use the `icapi` category of CT-VL commands to register the CipherTrust Vaultless Tokenization Server to a CipherTrust Manager (CM).

Below is the CT-VL setup Commands table:

Command	Description
<code>create</code>	Create a private key, or certificate signing request.
<code>enable</code>	Enable or disable the ICAPI interface to CT-VL servers.
<code>register</code>	Register to a CM host.
<code>set</code>	Set ICAPI properties, such as IP addresses, ports, and user credentials to a CM server.
<code>setup</code>	Perform an interactive setup of ICAPI properties (instead of using <code>set</code>).
<code>show</code>	Show ICAPI and CT-VL status and settings.
<code>test</code>	Test connection to a CM server or access to a specified key on the CM server.
<code>upload</code>	Upload a private key, signed certificate, or local CA certificate.

Network Management Category

Use the `network` category to set, modify, or delete system IP addresses, and to set up CipherTrust Vaultless Tokenization Servers. The `network` category also contains commands for showing and testing network configurations.

Note

Each server is assigned a unique IP address.

Below is the network management category commands table:

Command	Description
<code>arp</code>	Manipulate the system ARP cache.
<code>arping</code>	Send ARP requests to a neighbor host.
<code>checkport</code>	Check port connection status of a host
<code>dns</code>	Show or configure settings for the CT-VL DNS server(s).
<code>ethtool</code>	Query network driver and hardware settings.
<code>ifconfig</code>	Show or configure a network interface.
<code>ip</code>	Show or configure the CT-VL network interface settings.
<code>netstat</code>	Print network connections.

Command	Description
nslookup	Query internet name servers.
ntpservice	NTP service.
ping	Pings an IP address, host name, or FQDN.
route	Set static route.
service	Start/stop/restart network service.
set	Configure network device interface.
setup	Network setup wizard.
show	Show network device configuration.
snmp	Show, set, or change SNMP (v2c or v3) server configuration, which can be used to monitor processes, load average, disk space, and file size.
traceroute	Traces route to IP address or host name.

Example - Request:

```
network> ip address
```

Example - Response:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0c:29:47:22:72 brd ff:ff:ff:ff:ff:ff
inet 10.3.201.140/16 brd 10.3.255.255 scope global dynamic eth0
valid_lft 216sec preferred_lft 216sec
inet6 fe80::20c:29ff:fe47:2272/64 scope link
valid_lft forever preferred_lft forever

WARNING! Changes made using this command are not persistent. Use the setup command
to change network settings permanently.
```

System Configuration Category

The system configuration category enables you to configure the CT-VL host settings (such as, hostname, console port, or timezone), and view and manage other common system administration tasks for the server (such as, reboot or shut down, or view disk space usage or currently logged-in users).

Below is the system configuration category commands table:

Command	Description
<code>banner</code>	Configure SSH banner.
<code>cpuid</code>	Show detailed CPUid information.
<code>cpuinfo</code>	Show CPU information.
<code>daemon</code>	Show daemon services that are running.
<code>date</code>	Print or Set the system date and time.
<code>df</code>	Report file system disk space usage.
<code>free</code>	Display amount of free and used memory in the system.
<code>host</code>	DNS lookup utility.
<code>hostname</code>	Set/Show hostname.
<code>hosts</code>	Update hosts file.
<code>lastlogin</code>	Show last logged-in users.
<code>meminfo</code>	Show memory information.
<code>reboot</code>	Reboot system.
<code>shutdown</code>	Shut down system.
<code>terminal</code>	Show or terminate terminal (CLI) sessions.
<code>timezone</code>	Set or show timezone.
<code>top</code>	Display Linux processes.
<code>uname</code>	Print system information.
<code>uptime</code>	Show how long the system has been running.
<code>vmstat</code>	Report virtual memory statistics.
<code>who</code>	Show who is logged into the system.

CT-VL Configuration Commands

The `vts` category is used to configure the CT-VL, set time parameters, configure GUI login behavior, and upgrade the CipherTrust Vaultless Tokenization Server software.

Note

The CT-VL configuration commands still retain the `vts` category name to ensure backward-compatibility.

The `vts` category consists of the following commands:

Command	Description
<code>batch_size_limit</code>	Tokenization Batch size limit.
<code>cliadmin</code>	Cliadmin user setup. NOTE: If CT-VL runs on an AWS EC2 and has been set to use an EC2 key pair, password access is disabled, and <code>cliadmin password</code> commands will fail with an error message. For example, a password cannot be disabled or unlocked. To view if the <code>cliadmin</code> is using an AWS SSH key pair, run <code>vt cliadmin keys --list</code> CLI command. If the instance uses an EC2 key pair, an entry in the SSH keys will show <code>[ec2]</code>.
<code>client_certificate</code>	Enables client authentication by requiring clients to pass a client key and CA certificate to CT-VL. NOTE: For best security, enable client certificates with identities (the <code>--enable-with-id</code> option) instead of without identities.
<code>cors</code>	Class-Origin Resource Sharing (CORS).
<code>create_superuser</code>	Create a superuser local account for Web administrative GUI access.
<code>logfile</code>	List, view, or tail log files.
<code>loglevel</code>	Set log level.
<code>randommode</code>	Random Mode utility.
<code>rate_limiting</code>	Limit the amount of tokenization requests.
<code>remotelog</code>	Set remote logging.
<code>restore</code>	Restore system from backup.
<code>server_certificate</code>	Creates the self-signed certificate or certificate signing request and import signed certificate.
<code>service</code>	CT-VL Service utility.
<code>show</code>	Show the security settings.
<code>smtp</code>	Configure, view, or test email notifications and underlying SMTP setup.
<code>ssl_ciphers</code>	Show or set SSL cipher suites. WARNING: Non-FIPS approved SSL ciphers will be deprecated in the future. It is highly recommended you configure CT-VL to use the default SSL cipher suite (with the <code>vt ssl_ciphers --setdefault</code> command), otherwise the server can be open to security vulnerabilities.
<code>ssl_protocol</code>	Specifies what version of the SSL (TLS) is supported by the CT-VL.
<code>upgrade</code>	Upgrade CT-VL.
<code>weblogin</code>	Web login setup.

CT-VL Services

CipherTrust Vaultless Tokenization (CT-VL) is a platform-independent appliance (virtual machine or bare-metal) that offers REST-API services to protect sensitive data.

CT-VL Core Services

CT-VL offers the following service category for handling sensitive data:

- **Tokenization service:** Tokenization is frequently used for sensitive data such as credit card numbers, social security numbers, drivers licenses, or other personally identifiable information (PII). Data masking can be applied to any detokenized data to hide sections of the data from different groups of users.

For example, less-privileged database users might view only the last four digits of a detokenized credit card number, while a more privileged user could view the entire card number.

The CipherTrust Tokenization REST APIs are used to integrate this functionality into a developer's application.

Components in a CT-VL Deployment

CT-VL deployment consists of the following components:

- CipherTrust Vaultless Tokenization Server
- CipherTrust Manager
- LDAP/Active Directory Server (optional)
- Remote Logging Server (optional)
- Load Balancer (optional)
- SNMP Server (optional)

Tokenization

- Replaces sensitive data in databases with tokens. This reduces the number of places in which plain text credit card numbers reside, and thus reduces the scope of complying with the Payment Card Industry Data Security Standard (PCI DSS) and corporate security policies.
- Preserves the format of data in a way that reduces the operational impact associated with encryption and other obfuscation techniques. For example, you can tokenize a credit card field in a database, yet keep the tokenized information in a format that is compatible with associated applications.
- Enables outsourcing application testing and running analytics without giving access to sensitive assets because the format of the data has been preserved. To outsource, you can create a copy of the production database and give that copy to the outsourced development team.

- Creates strong separation of duties between privileged administrators and data owners. In this way, IT administrators, such as hypervisor, cloud, storage, and system administrators can perform their tasks without access to the sensitive data residing on those systems.
- Enables dynamic data masking—the ability to establish varying levels of data redaction for different database users. For example, you can enable customer service personnel to access the last four digits of a customer's credit card number, while an accounts payable representative can access the full credit card number.
- Integrates tokenization users with existing LDAP-based identity directories. Security teams can efficiently set granular tokenization policies for specific users and groups.
- Provides high-performance cryptographic operations to protect sensitive data: encrypt, decrypt, sign, and verify. CT-VL allows batch encryption and decryption of data in bulk to achieve high performance.
- Provides an alternative way of managing keys on the CipherTrust Manager.

CT-VL Key Cache

Keys are cached in CT-VL for efficiency and to minimize network latency.

If a key expires (that is, deactivated, compromised, and so on), there is a trade-off that each installation needs to take into account.

- **Integrity** - if this is more important, performance slows down considerably given that network transmission is orders of magnitude slower than accessing data in memory.
- **Performance** - if this is more important, there will be a small integrity sacrifice.

Determining Key Cache Expiration Time

CT-VL key cache has a default expiration time of 300 seconds. However, the expiration time also depends on whether the key is in use for a session or not.

Note

- If a key id has the `:refresh` attribute appended to it, it will clear both **local** and **cryptod** caches. But calling it on every request would degrade its performance, although importantly integrity is preserved.
- The strategy that can be used to find out remaining expiration time is to look at the `deactivation_date` attribute and figure how much time is available until the next rotation.

Automatic vs Manual Key Rotation Process

Automatic Key Rotation

A way to ensure that the key isn't expired is to do a **GET** on the key, and look at the `deactivation_date` attribute value. When you are close to expiration time, append the `:refresh` attribute until you see a different date in the future.

Manual Key Rotation

The `creation_date` can be used to calculate a deactivation date if you have a manual rotation process, and you know how long the key will last. As it is manual, you may want to do refreshes for a grater period of time until you see a new creation date.

Once out of the rotation risk zone, drop the `:refresh`, and you will get better throughput at these times.

There are no performance numbers for key operations, since the original assumption is that keys and their states do not change that often. CipherTrust Cryptographic operations do perform a key fetch internally if not cached.

Key Types and Input Sizes

User permissions (to encrypt, decrypt, sign, verify, hash, or tokenize/detokenize) must be tied to particular keys. The CipherTrust Key Management APIs provide an array of static key management functions, as well as the ability dynamically to create or import keys to the keystore.

Key Type

Key Type	Description
Symmetric	Oct or octet sequence. Used to: • tokenize • detokenize • encrypt • decrypt Symmetric keys are also used for tokenization, and while the symmetric keys are typical AES keys with 128 or 256 bits of key material, the algorithms are FPE (FF3) and FF1. Note: Versioned keys may not be used for tokenization, only non-versioned keys.

Key Input Sizes

For CBC and ECB modes of encryption, the input size should be a multiple of the block size.

Note

The AES block size is 16 bytes (128 bits). If you do not use an input length that is a multiple of the block size, use CTR or CBC-PAD encryption.

Refer to [Creating Static and Dynamic Keys](#) for keys creation.

Creating Static and Dynamic Keys

Keys can be created with either a static or dynamic work flow:

Static Work Flow

Note

It is recommended to use a static key model with tokenization.

In most cases, it is recommended to implement your application using static keys, that is, keys that will change their key material rarely, and thus are rarely created or destroyed in the keystore.

The implementation tasks for static key creation are:

- Decide which actions will be performed (tokenize/detokenize), and which types of keys will be needed to support those actions.
- **CipherTrust Manager:** Create CT-VL-specific keys in the keystore, giving them meaningful names.
- **CT-VL GUI Administrator:** Enter the appropriate key names into the CT-VL GUI, to be stored in the local CT-VL database, and used for assigning encryption and tokenization permissions.
- **Application Developer:** Use the CipherTrust Key Management, Cryptography, and Tokenization APIs to perform the necessary functions in the application being integrated with CT-VL.

Warning

Do not use the REST API *create* or *import* key functions.

Dynamic Work Flow

In some cases, an organization may decide to automate key creation, rotation, or key import in the keystore, using the appropriate CipherTrust Key Management APIs.

Warning

If a key is deleted, all of the ciphertext the key was instrumental in creating is rendered unreadable.

In this case, the implementation team would need to:

- Decide which actions will be performed (tokenize/detokenize), and which types of keys will be needed to support those actions.
- **Application Developer:** Use the CipherTrust Key Management and CipherTrust Cryptography APIs to perform the necessary functions in the application being integrated into CT-VL, including *create* and/or *import* key functions.
- **CT-VL GUI Administrator:** Designate a user (possibly a non-human user), and grant the user permission to create and/or import keys.

Tokenization Groups, Templates, and Masks

In tokenization, sensitive data such as credit card numbers or social security numbers, are replaced with an encrypted *token* which retains the format of the original data. When detokenizing, different users can be authorized to view the entire detokenized/unencrypted value, or authorized to see only a part of the value, with the rest hidden by a *data mask*.

CT-VL includes tokenization groups, templates, and data masks as convenience features to make implementing tokenization easier.

Tokenization Groups

A *tokenization group* is a way to compartmentalize CT-VL at a department level or site level. Each tokenization group has a symmetric key associated with it. While the CipherTrust key management and CipherTrust encryption services refer to keys directly, the tokenize/detokenize services refer to a key indirectly – via the tokenization group.

Managing Tokenization Groups

A *tokenization group* is used as a parameter in the Tokenization API calls, tokenize and detokenize. To use tokenization, you must create at least one tokenization group.

Note

- Have at least one symmetric key created when you create a tokenization group.
- Remember that all symmetric keys used for tokenization must be unversioned keys.

Tokenization Template

Tokenization templates are used to supply parameters to the tokenization/detokenization operation which otherwise would have to be supplied individually in the REST API call, thus making the REST API call both easier to use and shorter.

The CT-VL GUI administrator collaborates with the application developer to create tokenization templates with the appropriate settings, and to give the template a name that developers will use in their code.

Managing Tokenization Templates

Tokenization templates allow you to define the specifics of how you want your data tokenized and detokenized. For example, you can specify that your data be Luhn checked before tokenizing, or that the last four digits of a data string be left unencrypted, or that an identifying prefix be attached to all tokens.

To use tokenization, at least one tokenization template must be created.

Template Value	Description
Name	Name of the template. This name will be used in the API POST commands tokenize and detokenize.
Token Group	The tokenization group for which this template can be used. You must create at least one template for each tokenization group.
Format	Formats to be used to tokenize data: • Format-preserving encryption (FPE) with or without a Luhn check. A Luhn check ensures that a submitted number is a valid credit card number. Note: FPE is an alternate name for FF3. • Random or Random with Luhn check. This format is optimized for use with numbers from 9 digits (without a Luhn digit, for instance SSNs) to 19 digits (including a Luhn digit), like credit card numbers or tax ID numbers. If the data includes more free-form text, such as names and addresses, either the FPE or FF1 format is preferable. Note: In a random or random-Luhn format, the template uses a pseudo-random number generator (PRNG) instead of an encryption algorithm. In addition, the very first symmetric key configured in the CT-VL GUI is not a true encryption key, but instead acts as a seed value for the PRNG. • One of the supported date formats. Use this to anonymize dates. Between the month, date, and year portions of the date format, the input data can include one of the following separators: slash (/), comma (,), hyphen(-), or space (). Date formats are used along with the Start Year and End Year fields. When a user enters a date in YYYY format but the token format requires YY, the system uses only the first two digits of the year for tokenization and maps them to the current century. The last two digits of the year are preserved

Template Value	Description
	exactly as entered during both tokenization and detokenization. Note: In a date format, if a two-digit year format is selected (for example, DDMMYY), the CT-VL assumes that the date is in the twenty-first century; that is, it assumes the first two digits of the year are 20.
Character Set	The character set to use with tokenization.
Prefix	(Optional) An optional prefix to be added to all tokens. Example: CC-1234-1234-1234-1234 or SSN:123-23-1234. Note that the format is no longer preserved with this feature.
Keep Left	The number left digits to leave unencrypted. You need at least a total of two digits to tokenize. Cannot be left blank.
Keep Right	The number right digits to leave unencrypted. You need at least a total of two digits to tokenize. Cannot be left blank.
Irreversible	(Optional) Check this box if you never want the token to be detokenized. For example, if you have production data that you want to tokenize for test or development, and you know you'll never want to decrypt it
Allow empty or 1 character inputs	Enables null, empty, and one-character inputs, which are passed through. The token value returned is the original input value. The single-digit input supports both characters and numerals, and is available for all formats except RANDOM.

Minimum Input Characters with Keepleft or Keepright

If you configure using keepleft or keepright, note the following minimum numbers of input characters, according to tokenization format:

- For FPE: 2 characters plus the number of keepleft or keepright characters.
- For RANDOM Numeric: 9 characters plus the number of keepleft or keepright characters.
- For RANDOM Alphaumeric: 5 characters plus the number of keepleft or keepright characters.

Data Masks

When a user or user group is given detokenization privileges, a data mask must be applied that defines how much of the unencrypted data they are allowed to see.

To create a mask that shows the entire data string, the CT-VL GUI Administrator would enter very large values in the Show Left and Show Right fields (999999).

Understanding CT-VL Authentication and Authorization

CT-VL Users

CT-VL makes no official distinctions between “human user accounts,” for real people, and “service user accounts,” representing daemons, scripts, processes, applications, or groups within applications with programmatic access to CT-VL services. Both are referred to as “users.”

From the point of view of the CT-VL system, the CT-VL REST APIs and the application developer who wishes to incorporate CT-VL services into an application, a basic user account consists of:

- **user name/identifier**
- **password or client certificate** (If authentication is done from a client certificate, the password is not used.)

These attributes form the basis of user authentication.

Note

As a best practice in CT-VL, it is recommended to collect individual users into logical user groups and assign permissions at the group level. See [Understanding CT-VL user groups and permissions](#) for more information.

User Profiles

Users can be human or servers.

Human users

Human users include the administrators, staff, and developers who use the CT-VL interface and/or the CT-VL APIs to install, configure, integrate, and maintain its services.

There are also human “end-users,” such as employees in a customer service department who may be assigned to a CT-VL user group authorized to detokenize credit cards, and to view the last four digits of the card, based on the data mask that was applied to their user group.

Service users

However, most CT-VL usage consists of service users, where various servers and applications pass authentication, tokenization, encryption, and key information between themselves to complete a transaction of some kind.

For example:

A company may set up a structure to assign a financial application as **User1** with a certain number of keys to perform certain operations. Within that same application, **User2** might be a process that uses **key2** to tokenize/detokenize credit cards, and **User3** might be used to encrypt/decrypt social security numbers with **key3**.

CT-VL Authentication Overview

CT-VL first seeks to validate whether the user credentials match entries in its local database. If not, and if AD/

LDAP is configured, then it will check against AD/LDAP settings. When authentication is successful, the system will also check for CT-VL role assignment, any key association, and any active permissions.

AD/LDAP and User Authentication

Companies can choose to use Active Directory/LDAP to store both human and service user accounts for CT-VL, to have a centralized way to manage them. If this approach is chosen, then the implementation involves cooperation between:

- The network administrator, to configure the connection between AD/LDAP and CT-VL.
- The AD/LDAP administrator and other organizational decision makers who decide which users should be imported into the CT-VL database, how to group them, what user group names to assign them in AD/LDAP, and what types of permissions should be granted on a group and/or individual basis.
- The CT-VL GUI administrator, who must manually enter the designated user group names into the CT-VL GUI before the user accounts are imported.

Note

It is also possible to create local user accounts directly in CT-VL. This is useful in test or demo installations of CT-VL or might be used for enhanced security as no AD/LDAP administrator would have access to the local user credentials.

Client Certificates and User Authentication

Client certificates can be used for authentication. The certificate itself is a credential and can be safer than a password, as it cannot be easily shared, duplicated, written on a sticky note, or otherwise carelessly exposed.

Companies might enable simple client cert authentication, to verify that a machine has a valid cert to connect with CT-VL. Or they may choose a more sophisticated implementation.

With CT-VL, it is possible to use the common name (CN) inside a client cert and enter it into CT-VL as a user name, thereby taking advantage of the ability to assign granular permissions and/or associate particular keys to a client certificate. This is known as using a client certificate “with identities.”

Note

Thales highly recommends using client certificates with identities. If you use without identities, you must pass in a username/password along with the certificate, which can be a security vulnerability that client certificates are intended to prevent. The option to use certificates without identities is provided to maintain backward compatibility with the earlier basic authentication flow.

To implement client certs with granular permissions, an organization would need to:

- Have established an in-house Certificate Authority (CA).

-
- Have an exportable CA cert.
 - Issue any number of client certs (signed by the CA) that each contain a common name (CN).
 - Go to the CT-VL CLI to:
 - Upload the CA cert to CT-VL.
 - Enable using client certs with ID.
 - Use the CT-VL graphical user interface (GUI) for:
 - Creating users and entering common names as user names (or import from Active Directory for LDAP).
 - Managing user permissions (If importing from LDAP, you can use the CT-VL CLI setting to designate a user group as the “active” user group thereby activating all members of the group.)
 - Associating key(s) with the cert/user.
 - Assigning permissions to the cert/user.

Note

CT-VL requires that all client certificates be in PEM format.

CT-VL Authorization Overview

CT-VL employs a multi-layered security model with separation of duties and granular permissions settings built in to it. It is also flexible, adaptive to individual corporate structures and policies. This structure implies some inherent complexity, but the concepts below form the foundational assumptions.

CT-VL Roles

When a user is created in CT-VL, it can be assigned one or more of the following core CT-VL roles:

- **Superuser:** Can perform both read and write actions in the CT-VL GUI.

Superusers are GUI administrators who manage users, user groups, keys, tokenization groups, tokenization templates, data masks, and character sets. Superusers associate users with key(s) and assign the crypto, key, and tokenization permissions to other users. Superusers are automatically also “Staff.”

- **Staff:** Users with read-only permission to the CT-VL GUI.
- **Active:** Users must be “active” in order to have encryption, key, or tokenization permissions. Active users are those authorized to use the CT-VL REST APIs.

The role-based permissions (Active/Staff/Superuser) are incremental. With no role assigned, the user cannot

access any CT-VL functionality. (This is equivalent to being deleted, but the information is retained in the database for possible reactivation.)

Tip

- When users are created directly in the CT-VL GUI, they are automatically assigned the status “active,” which can be disabled as needed.
- If users are imported from AD/LDAP, the “active” designator is handled differently.
- If Active is unchecked, the user cannot log in to the GUI even if assigned Superuser/Staff status.

Key-Aware Permissions Matrix

The GUI Administrator can assign permissions based on user (or user group), action, and key.

In the following example, User **CTUser1** may encrypt, decrypt, tokenize and detokenize with symmetric key **Sym_key1**.

Update User Permission

User: vtsuser1 Key: Sym_key1

Encryption	Key Management	Tokenization
☒ Encrypt	☐ Destroy	☒ Tokenize
☒ Decrypt	☐ Modify	☒ Detokenize
☐ Sign	☐ Export	Mask
☐ Verify	☐ Find	last 4 ▼

Apply Cancel

Note

- Permissions can be assigned at the user and the user group level, and are additive.
- Different types of keys (symmetric, asymmetric, and opaque objects) have different permissions

available.

- Tokenize and detokenize functionality require a couple of additional steps. See [Tokenization Groups, Templates, and Masks](#) for details.

Other (Non-Key Dependent) Permissions

Users or user groups can be authorized to dynamically create and/or import key names and key material on the keystore, under the "Other Permissions" column of the Permissions main page.

Create key: Creates the key name and key material in the keystore; the key name will be listed in the CT-VL GUI.

Import key: Used to import any key from one keystore to another, or from an external system into the keystore. The imported key name will be listed in the CT-VL GUI.

Note

The *Other Permissions* assignment window also includes check boxes for Tokenize and Detokenize, along with a warning about overriding symmetric key assignments.

This Tokenize/Detokenize feature is used during upgrade from earlier versions of CT-VL, as its grandfathers in the global tokenize/detokenize feature that was used previously.

Understanding CT-VL User Groups and Permissions

▪ CT-VL role-based user groups:

The three core CT-VL roles (Superuser/Staff/Active) are also three required core CT-VL user groups. An organization may choose to assign different group names, but the meanings are retained. CT-VL GUI administrators must create at least three user groups that correspond to the roles Superuser/Staff/Active.

▪ Assigning users to user groups in two different ways:

- If importing from AD/LDAP, users will be sorted into their role groups from the CLI.
- If entering users directly in the GUI, the administrator will assign the role(s) on the user's page and then check the appropriate group(s).

▪ Adding more user groups:

Additional groups can be added to CT-VL for organizational purposes.

▪ Assigning group permissions:

Key-based and non-key-dependent permissions can be assigned at the user group level or the user level, but assigning at the group level is recommended.

- **Permissions are additive:**

Individual and group permissions are additive. For example, suppose Joe is a member of the user group “Finance,” which has permission to tokenize credit card numbers, and Joe is also a team lead granted user-level permission to detokenize. In total, Joe can tokenize and detokenize.

Users, User Groups, and Tokenization Groups

To allow for abbreviated Tokenization REST API calls and to enable division of an organization into sub-organization either by location or department, CT-VL adds a so-called “tokenization group” concept.

So, there could be a user group “Finance” in San Francisco, and another “Finance” user group in Paris, each associated with its own tokenization group.

CT-VL CLI Navigation Summary and Guidelines

Navigation	Command	Example
Starting prompt	main>	
List CLI categories or the commands within categories	?	<pre>main> ? authAuthentication setup clusterCluster Setup icapiSafeNet ProtectApp Setup networkNetwork Setup systemSystem Setup vaeVAE Configuration vtsTokenization Setup quit q up <ctrl-d>Quit or return to previous menu exitExit application</pre>
Open a category and display its commands	?	<pre>main> cluster cluster> ? addRegister a new node to a cluster apiadminAPI admin user setup createCreate New Cluster joinJoin node to a cluster remotejoinRemotely join nodes to a cluster removeRemove node from cluster showShow cluster status quit q up <ctrl-d>Quit or return to previous menu exitExit application</pre>
Get help: See command usage and example output		<pre>network> checkpoint usage: checkpoint [-h] [-- host HOST] [--port</pre>

Navigation	Command	Example
		<pre>PORT[PORT ...]]</pre> Check port connection status of a host Optional arguments: <pre>-h, --helpshow this help message and exit</pre> <pre>--host HOSThostname or IP address of host</pre> <pre>--port PORT [PORT ...] port numbers to scan</pre>
Have CLI complete a category name, command, or argument when typing	Press	Enter enough characters to uniquely identify a category, command, or argument, and then press the key. The CT-VL CLI will complete it for you.
Return to the main level	up	<pre>main> cluster</pre> <pre>cluster> up</pre> <pre>main></pre>
End the CLI session	exit	

Note

You must enter a category to execute any of its commands. For example, the reboot command is in the system category, so you must first enter system, then enter reboot.

Tokenization Setup Workflow in CT-VL GUI

Although the CT-VL GUI does not constrain you to one particular implementation path, one logical sequence is:

- Specify the CM Keys for Tokenization (must be symmetric keys).
- Create Tokenization Groups and associate it with the appropriate key(s).
- Create custom character sets as needed (optional).
- Create a data mask for dekotenization.
- Create tokenization templates and associate the applicable tokenization group and character set.
- Create users and user groups, assigning a data mask from the drop-down menu.
- Grant tokenization privileges to users and user groups by going to their permissions page, selecting the symmetric key associated with the appropriate tokenization group, and checking the Tokenize box.
- Grant detokenization privileges to users and user groups in the same way.

Note

See [Other \(non-key dependent\) permissions](#) for an explanation of the Tokenize/Detokenize check boxes listed under "Other Permissions" in the Permissions matrix. This feature is included primarily for legacy and upgrade purposes.

CT-VL Tokenization REST APIs

The CipherTrust Vaultless Tokenization API consists of:

- **tokenize** - A POST call that tokenizes data. A user submits sensitive data, the CipherTrust Vaultless Tokenization Server returns a token. The token is then stored in the production database.
- **detokenize** - A POST call that detokenizes data. A user submits a token to the CipherTrust Vaultless Tokenization Server and the actual data is returned detokenized. Permission settings for that user dynamically specify the appropriate data mask.
- **log** - Changes the log level. Four different log levels (error, info, debug, and trace) can be selected. Chose 'error' or 'info' for normal operation, or 'debug' when instructed to do so by Thales eSecurity support.

Using the APIs

To enable tokenization, you do not have to change your database schema, because CipherTrust Vaultless Tokenization preserves the format of your data. However, you must change your application code.

When a user makes a request to tokenize or detokenize, the username and password must be provided. This is a part of a REST request. The username/password is validated by either the AD/LDAP server or CT-VL, depending on where the user is defined.

Note

- The data passed to and from the CipherTrust Vaultless Tokenization Server; user name, tokenization group name, tokens, and sensitive data, is case-sensitive, and the application code that passes data to CT-VL must be aware of this.
- Tokenization input can be a normal text format.

The maximum input data size for CT-VL Tokenization REST APIs is given below:

API	Maximum Input Data Size
Tokenization and Detokenization	128 KB
Batch Tokenization and Detokenization	1.25 MB (maximum records: 5000, maximum individual record size: 128 KB)

Create a Token

Description

Use the `tokenize` endpoint to encrypt a data string to a token using format-preserving encryption (FPE or FF1).

Tokenize Endpoint Details

Data	Description
HTTP Request	<code>https://IP_Address/vts/rest/v2.0/tokenize</code>
Method	POST
CT-VL IP Address	The IP address of the CT-VL instance to access with the API call.
Response Format	JSON
Required Authentication	A valid user name and password must be passed as an argument. The values are case sensitive.

Tokenize Parameters

The parameters for tokenize endpoint are described in the table below:

Parameter	Required	Type	Description
tokengroup	Y	string	Defines a group name space in the configuration database. For example, let's say you are a credit card company and you want the CT-VL to support three departments supporting Store1, Store2, and Store3. You want employees in these departments to have access to data for those respective stores. Two actions are required: • The application code must specify a different tokengroup parameter in the POST calls for each group supported (example: Store1, Store2, and Store3). The tokengroup parameter is case sensitive. • The CT-VL GUI administrator must create token groups that match the tokengroup parameter names used in the

Parameter	Required	Type	Description
			application code. If you do not support multiple token groups, then tokengroup will always be the same value. Example value: store1
data	Y	string	The data string to tokenize. The argument considers case and is limited to 128KiBs Example value: 1432445695309134
tokentemplate	Y	string	CT-VL GUI Administrator-defined name for a group of properties that define tokenization operation. Properties include the token group to which the template applies, the tokenization format (date, FPE or FPE with Luhn check, or FF1 or FF1 with Luhn check, or Random (lookup) or Random with Luhn check), number of leftmost or rightmost characters to not tokenize, whether you wish to never detokenize a tokenized entry, the character set used for tokenization, and an optional prefix for tokens. Example value: Credit Card
returnData	N	string	Tokenization requests can return the plaintext string together with the encrypted data in the response. To enable, add the "returnData" parameter and set it to true.

Tokenize Response Parameters

The tokenize endpoint response parameters, their types, and corresponding descriptions are listed in the table below:

Parameter	Type	Description
token	string	Token for the sensitive data.

Parameter	Type	Description
		Example value: 2134234509874567
status	string	•Succeeded (if request succeeded) •Error (if request failed)
reason	string	If status is Error a reason is displayed. Otherwise the reason is omitted.
data	string	If returnData in the request is set to true, contains the plaintext string for the encrypted data.

Create Token CURL Example

Note

If you use the `--tlsv1.2` CURL Command syntax option, you must use CURL version 7.34 or higher.

Example Request

```
curl -k -X POST -u Donald_Duckmiller:Panda45# -d '{"tokengroup" :
"ctsUsers-t" , "data" : "9453677629008564", "tokenTemplate" :
"ctsUsersTemplate" }' https://192.168.118.140/vts/rest/v2.0/tokenize
```

The user name (Donald_Duckmiller) and tokengroup (ctsUsers-t) are case sensitive.

Example Response

```
{"token":"CC-0332335756020172","status":"Succeed"}
```

If FPE-luhn, FF1-luhn, or Random-luhn is selected in the Change Token Template page of the CT-VL GUI, the data must pass a Luhn check or tokenization will fail with the error message:

```
"status":"error","reason":"Luhn check failed for input data."
```

Tokenizing Multiple Data Items

Up to 5,000 data items can be tokenized in a single `curl` command by submitting a JSON array in the REST request. For example, to tokenize four data items you might enter the following in a `postcnn.txt` file:

Example Request

```
cat postcnn.txt
```

```
[{"tokengroup" : "ctsUsers-t" , "data" : "9453677629008564",
"tokentemplate" : "ctsUsersTemplate" }, {"tokengroup" : "ctsUserst"
, "data" : "9453677629008564", "tokentemplate" :
"ctsUsersTemplate" }, {"tokengroup" : "ctsUsers-t" , "data" :
"9453677629008564", "tokentemplate" : "ctsUsersTemplate"
}, {"tokengroup" : "ctsUsers-t" , "data" : "9453677629008564",
"tokentemplate" : "ctsUsersTemplate" }]
```

```
curl -k -X POST -u Donald_Duckmiller:Panda45# --data-binary
@postccn.txt https://192.168.118.140/vts/rest/v2.0/tokenize
```

Warning

Newlines are NOT allowed in the input.

Example Response

```
[{"token": " 9465976792116170", "status": "Succeed"},
{"token": " 9465976792116170", "status": "Succeed"},
{"token": " 9465976792116170", "status": "Succeed"},
{"token": " 9465976792116170", "status": "Succeed"}]
```

The tokenized data is returned in the CURL Command syntax response, in order of submission.

Get Detokenize Resource Data

Get detokenized data from a resource.

Detokenize Endpoint Details

Data	Description
HTTP Request	https://IP_Address/vts/rest/v2.0/detokenize
Method	POST
CT-VL IP Address	The IP address of the CT-VL instance to access with the API call.
Response Format	JSON
Required Authentication	A valid user name and password must be passed as an argument. The values are case sensitive.

Detokenize Request Parameters

Parameter	Required	Type	Description
token	Y	string	The token to detokenize.

Parameter	Required	Type	Description
			token is case sensitive. Example value: 6029541314537206
tokengroup	Y	string	Defines a group name space in the configuration database. For example, let's say you are a credit card company and you want the CT-VL to support three departments supporting Store1, Store2, and Store3. You want employees in these departments to have access to data for those respective stores. Two actions are required: • The application code must specify a different tokengroup parameter in the POST calls for each group supported (example: Store1, Store2, and Store3). The tokengroup parameter is casesensitive. • The CT-VL GUI administrator must create token groups that match the tokengroup parameter names used in the application code. If you do not support multiple token groups, then tokengroup will always be the same value. Example value: store1
tokentemplate	Y	string	CT-VL GUI Administrator-defined name for a group of properties that define tokenization operation. Properties include the token group to which the template applies, the tokenization format (date, FPE or FPE with Luhn check, or FF1 or FF1 with Luhn check, or Random (lookup) or Random with Luhn check), number of leftmost or rightmost characters to not tokenize, whether you wish to never detokenize a tokenized

Parameter	Required	Type	Description
			entry, the character set used for tokenization, and an optional prefix for tokens. Example value: Credit Card

Detokenize Response Parameters

Parameter	Type	Description
data	string	The detokenized sensitive data. May be masked depending on the configuration. data is case sensitive and limited to 128KiB characters.
status	string	Succeeded (if request successful) or Error (if request failed).
reason	string	If status is Error, then a reason will be displayed. Otherwise the reason will be omitted.

Example

Note

If you use the `--tlsv1.2` CURL option, you must use CURL version 7.34 or higher.

Request

```
curl -k -X POST -u tok-user1:Panda45# -d '{"tokengroup" : "ctsUsers-t" , "token" : "CC-0332335756020172", "tokenTemplate" : "ctsUsersTemplate" }' https://192.168.118.140/vts/rest/v2.0/detokenize
```

If the client authentication feature is enabled, the example syntax is as follows:

```
curl -k --key client.key --cert client.crt -X POST -u tok-user1:Panda45# -d '{"tokengroup" : "ctsUsers-t" , "token" : "CC-0332335756020172", "tokenTemplate" : "ctsUsersTemplate" }' https://192.168.118.140/vts/rest/v2.0/detokenize
```

Response

```
{"data":"9453677629008564","status":"Succeed"}
```

Detokenizing Multiple Data Items

Up to 5,000 data items can be detokenized in a single curl command by adding multiple data items. For example, to detokenize four data items, you might enter the following in the `posttoken.txt` file:

Request

```
cat posttoken.txt

[{"tokengroup" : "ctsUsers-t" , "token" : "CC-0332335756020172",
"tokentemplate" : "ctsUsersTemplate" }, {"tokengroup" :
"ctsUsers-t" , "token" : "CC-0332335756020172", "tokentemplate"
: "ctsUsersTemplate" }, {"tokengroup" : "ctsUsers-t" , "token" :
"CC-0332335756020172", "tokentemplate" : "ctsUsersTemplate"
}, {"tokengroup" : "ctsUsers-t" , "token" : "CC-
0332335756020172", "tokentemplate" : "ctsUsersTemplate" }]
curl -k -X POST -u tok-user1:Panda45# --data-binary
@posttoken.txt https://192.168.118.140/vts/rest/v2.0/detokenize
```

Response

```
[{"data": "9453677629008564", "status": "Succeed"}, {"data": "9453677
629008564", "status": "Succeed"}, {"data": "9453677629008564", "statu
s": "Succeed"}, {"data": "9453677629008564", "status": "Succeed"}]
```

The detokenized data is returned in the CURL response in corresponding order of submission.

Note

Newlines are NOT allowed in the input.

If FPE-luhn, FF1-luhn, or Random-luhn is selected in the Change Token Template page of the CT-VL GUI, the data must pass a Luhn check or detokenization will fail with the error message.

Set Log Levels

Set the log level for the REST API.

Log Endpoint Details

Data	Description
HTTP Request	https://IP_Address/vts/rest/v2.0/log
Method	POST
CT-VL IP Address	The IP address of the CT-VL instance to access with the API call.

Data	Description
Response Format	JSON
Required Authentication	None

Log Request Parameter

Parameter	Required	Type	Description
loglevel	Y	Enum	loglevel can take one of the following values: • error - Default value used in normal production environment. • debug - Use only when directed by support. • info - Do not use unless maximum logging is required.

Note

The performance of tokenization and detokenization operations are severely affected by a log level of **debug**, and substantially affected by a log level of **info**. For maximum throughput, use a loglevel of **error**.

Log Response Parameter

Parameter	Type	Description
status	string	Success (if request succeeded) or Error (if request failed).

Example

Note

If you use the `--tlsv1.2` cURL option, you must use CURL version 7.34 or higher.

Request

```
curl -k --tlsv1.2 -X POST https://127.0.0.1/vts/rest/v2.0/log --databinary '{"log level":"debug"}'
```

Response

```
{"status": "success"}
```

CT-VL Admin REST APIs

The CT-VL Administration REST API allows you to manage the administration of the CT-VL. After completing the administrative tasks, you will be able to run the tokenize and detokenize data commands.

To use the Admin REST API you must be using:

- The CT-VL Server version 2.3.0 or newer.
- Any keystore server version compatible with CT-VL 2.3.0.

Base URL

The following is the base URL for Admin REST API operations:

```
https://<server_address>/api/
```

Authentication

JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JavaScript Object Notation (JSON) object that is used as the payload of a JSON Web Signature (JWS) structure or as the plaintext of a JSON Web Encryption (JWE) structure, enabling the claims to be digitally signed or MACed and/or encrypted.

By default, JWT authentication expires in 10 minutes. This setting can be changed using the CLI, under the `auth` menu.

Retrieve a Token Request

Request

```
curl -X POST \
-H "Content-Type: application/json" \
-d '{"username": "admin", "password": "Password123!"}' \
https://{server_address}/api/api-token-auth/
```

Response

```
{ "token": "eyJ0eXA [...] KnA74" }
```

Refresh a Token Request

Request

```
curl -X POST \
-H "Content-Type: application/json" \
-d '{"token":"<EXISTING_TOKEN>"}' \
https://{server_address}/api/api-token-refresh/
```

Response

```
{ "token": "eyJ0eXA [...] KnA74" }
```

Responses

This section describes request response codes and response pagination details.

Response Codes

HTTP status codes are grouped into five categories:

- 1xx for informational
- 2xx for success
- 3xx for redirection
- 4xx for client errors
- 5xx for server errors

Pagination

Requests that return multiple items will be paginated to 25 items.

Responses include a **count** field, which specifies the total number of objects available via pagination.

Responses also include a **next** and **previous** field. If **count** is superior to 25, these fields contain a URL path to access the objects located respectively after or before.

Request

```
GET https://{server_address}/api/v1/users/?limit=100&offset=400
```

Response

```
HTTP 200 OK
{
  "count": 1023
  "next": "https://{server_address}/api/users/?limit=100&offset=500",
  "previous": "https://{server_address}/api/users/?limit=100&offset=300",
  "results": [ ... ]
}
```

Operations

The CT-VL Admin REST API supports the following operations:

Searching

To search objects, add the attribute `?search=` at the end of the URL followed by the pattern you want to look for. All the objects containing that pattern in one of their attributes will be included in the Response.

Request

```
GET https://{server_address}/api/users/?search=robert
GET https://{server_address}/api/groups/?search=finance
```

Ordering

To query object in a specific order, add the attribute `?ordering=` at the end of the URL followed by the attribute you want to order by.

Ordering Examples

Order users by username:

```
GET https://{server_address}/api/v1/users/?ordering=username
```

You may also specify reverse orderings by prefixing the field name with '-', like so:

```
GET https://{server_address}/api/v1/users/?ordering=-username
```

Multiple orderings may also be specified:

```
GET https://{server_address}/api/v1/users/?ordering=username,email
```

Object Format

The response format for all requests is a JSON object.

Creating Objects

To create a new object, send a POST request to the class URL containing the contents of the object. For example:

```
curl -X POST \
  -H '{"Authorization': 'Bearer '+<token>}' \
  -H "Content-Type: application/json" \
  -d '{"name": "User1", ...}' \
  https://{server_address}/api/{some_resource}/
```

Retrieving Objects

To retrieve a list of objects, send a GET request to the class URL. For example:

```
curl -X GET \
  -H '{"Authorization': 'Bearer '+<token>}' \
  -H "Content-Type: application/json" \
  https://{server_address}/api/{some_resource}/
```

Updating Objects

To edit an object, send a PUT request to the URL containing the new content with the object's ID. For example:

```
curl -X PUT \
  -H '{"Authorization': 'Bearer '+<token>}' \
  -H "Content-Type: application/json" \
  -d '{"name": "User1", ...}' \
  https://{server_address}/api/{some_resource}/{id}
```

Deleting Objects

To delete an object, send a DELETE request to the class URL with the object's ID. For example:

```
curl -X DELETE \
  -H '{"Authorization': 'Bearer '+<token>}' \
  -H "Content-Type: application/json" \
  https://{server_address}/api/{some_resource}/{id}
```

Backing up CT-VL

Schedule automatic backups of the CT-VL.

```
GET https://<IP_Address>/api/backup/create
```

This request must be preceded by a successful authentication request. Use the secure token provided with the response in the portion of this example..

Set Backup Endpoint Details

Data	Description
HTTP Request	https://IP_Address/api/backup/create
Method	GET
CT-VL IP Address	The IP address of the CT-VL to access with the API call.
Response Format	JSON
Required Authentication	Yes

Example cURL Requests

Authentication

```
curl -X POST\
  -H "Content-Type: application/json" \
  -d '{username:"ctsuser","password:"Password123!"}' \
  https://cts_server_address/api/api-token-auth
```

Backup Request

```
curl -X GET\
  -H '{Authorization}:"Bearer <token>"}'\
  -d '{username:"ctsuser","password:"Password123!"}' \
  https://cts_server_address/api/backup/create -o backup.zip
```

Unauthenticated Healthcheck

Description

Use the `healthcheck` endpoint to get the status of the CT-VL. This endpoint integrates with load balancers to ensure all the daemons are running, and the tokenization server is healthy. An HTTP response code not equal to 200 indicates failure in CT-VL. The HTTP response also provides a display of status of all services, in JSON format.

Healthcheck Endpoint Details

Data	Description
HTTP Request	https://IP_Address/healthcheck
Method	GET
CT-VL IP Address	The IP address of the CT-VL to access with the API call.
Response Format	JSON

The unauthenticated healthcheck response parameters and their respective types are described in the table below:

Unauthenticated Healthcheck Response Parameters

Data	Type	Description
status	string	• 200 (if request succeeded) • 503 (if one of the daemons is not running)
	string	Configured services are listed along with each one's status (for example: {"name": "vaed", "status": "started"})

Healthcheck Example

Example Request

```
curl -k -X GET https://IP_Address/healthcheck
```

Example Response

```
HTTP/1.1 200 OK
Date: Thu, 29 Oct 2020 22:51:16 GMT
Content-Type: application/json
Content-Length: 310
Connection: keep-alive
Server: CT-VL server
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Pragma: no-cache
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Strict-Transport-Security: max-age=31536000; includeSubdomains; preload
Content-Security-Policy: default-src 'self'
{
  "services": [
    {
      "name": "vtsauthz",
      "status": "Started"
    },
    {
      "name": "vaed",
      "status": "Started"
    },
    {
      "name": "uwsgi",
      "status": "Started"
    },
    {
      "name": "nginx",
```

```

    "status": "Started"
  },
  {
    "name": "kmd",
    "status": "Started"
  },
  {
    "name": "iam",
    "status": "Started"
  },
  {
    "name": "cryptod",
    "status": "Started"
  }
],
"status": "Started"
}

```

CT-VL with Multiple Key Managers

This section describes the CT-VL operation mode (round-robin or failover) with multiple key managers.

Data Security Manager (DSM)

The VAE agent receives a list of the DSMs to contact during registration. This list is arranged in the order that primary DSM is at the top and failovers at the bottom. When the VAE agent detects that the primary DSM is down, it switches the order of both DSMs (primary and the first failover) in memory. After switching, the first failover DSM is at the top and primary at the second position.

This is done so that the VAE agent does not try to contact the primary DSM and timeout every time before going to the failover. The VAE agent will now skip the primary DSM and go to the failover.

Note

The VAE agent will not automatically failback to the first DSM even if it comes up. The services need to be restarted on CT-VL.

CipherTrust Manager (CM)

CT-VL with the CipherTrust Manager uses the round-robin operation mode. You just specify multiple IP addresses of the CipherTrust Manager servers and internal round-robin load balancing is performed.

Troubleshooting

Tokenization API Call Failure Error Messages and Workarounds

Following error messages may occur when an Tokenization API call is made:

Message	Description
--tlsv1.2: is unknown.	Only CURL version 7.34 or later supports the --tlsv1.2 option. If you get this error message, upgrade to CURL version 7.34 or higher.
API call aborted and no JSON vector elements are processed.	These are error messages which pertain to the whole JSON vector. In other words, in these cases no vector elements are processed at all. The whole operation is aborted.
User name and/or password mismatch.	Due to a wrong user name and/or password, access to the CipherTrust Vaultless Tokenization Server is denied.
Unknown JSON keyword encountered. The valid keyword for the log request is loglevel.	For the REST API called log , the only valid keyword on the left of the JSON valuename-value tuple is loglevel .
Could not parse JSON input, perhaps it is too long.	An input size of 1 MByte cannot be exceeded. It could also mean that the JSON input is malformed.
User name and/or password missing.	Either the user name, or the password, or both are missing from the request.
This user does not have permission to tokenize.	While the user exists and the password is correct, this particular user does not have the permission to tokenize. Consider assigning this user the permission to tokenize via the GUI.
User name found, but the user is not marked active.	While the user exists and the password is correct, this particular user is not marked as an active user. Consider marking this user an active user in the GUI.
Invalid operation. Use /vts/rest/v2.0/tokenize , or /vts/rest/v2.0/detokenize .	An invalid URL was specified. Allowed operations are: /vts/rest/v2.0/tokenize , /vts/rest/v2.0/detokenize , or /vts/rest/v2.0/log .
No password specified.	While a user name was specified, the password is missing.
User name not found.	The specified user name exists neither in the local database which is configured via the GUI, nor in the optional LDAP database.
LDAP user found in local database, but LDAP is disabled.	Presumably due to a prior use of LDAP, an LDAP user was cached in the local database, but LDAP has since been disabled, disallowing authentication with this user. Prior to using LDAP users for authentication, at least one of the groups this user belongs to according to the LDAP database must be configured as a user group via the GUI.
API call aborted, one JSON vector element fails, others are processed.	These messages pertain to only one element of the JSON vector. In these cases, one vector element may result in an error, but all other vector elements are still processed.
The length of the token exceeds 128 Kbytes.	The length of a token is limited to 128 KiB. However,

Message	Description
	the token supplied was longer than 128 KiB.
Unknown JSON keyword encountered. Valid keywords for the detokenize request are tokengroup, token, and tokentemplate.	For the detokenize operation, only three specific keywords are allowed on the left side of a JSON tuple: tokengroup , token , and tokentemplate . Please review and correct your JSON string.
The length of the data exceeds 128 Kbytes.	The length of the input data is limited to 128 KiB. However, the input data supplied was longer than 128 KiB.
Unknown JSON keyword encountered. Valid keywords for the tokenize request are tokengroup, data and tokentemplate.	For the tokenize operation, only three specific keywords are allowed on the left side of a JSON tuple: • tokengroup • data • tokentemplate
Please review and correct your JSON string.	A token group was specified in the JSON input, but it doesn't exist in the GUI-administered configuration database. Consider adding this token group via the GUI.
The requested key name was not found on the CipherTrust Key Manager.	A key name was configured via the GUI and now used in an operation, however this key name does not exist on the CipherTrust Key Manager.
The token group specified was not found in the database.	The token group was not found on CT-VL.
tokengroup not specified in JSON request.	The JSON tuple named tokengroup is missing from a JSON vector element. A tokengroup must be specified.
The requested combination of tokentemplate and tokengroup was not found in the tokentemplate table.	Either the tokengroup specified has not been entered via the GUI, or the tokentemplate has not been entered via the GUI, or the token template of the same name has been entered, but for a different tokengroup. Please ensure that the tokentemplate you are trying to use has been entered for the tokengroup you are trying to use.
It is not allowed to specify the FPE-luhn token format with a non-zero keepright value.	After accounting for keepleft and keepright, not enough input characters are left to successfully tokenize the input. When you use the FPE token format, the number of input characters must exceed the sum of the Keep Left value and the Keep Right value by at least two. When you use the FPE-luhn token format, the number of input characters must exceed the sum of the Keep Left value and the Keep Right value by at least three.
There are not enough input characters for the detokenize operation.	At least two input characters are needed in the token in order to try to detokenize it.
Token format FPE-luhn is only compatible with an alphabet consisting of the 10 decimal digits (in ASCII	The FPE-luhn token format cannot be used with an alphabet containing letters and/or special characters. It

Message	Description
encoding).	must be used with an alphabet only containing the 10 digits.
After accounting for the keepleft and keepright requirements, not enough input characters left to tokenize.	When you use the FPE token format, the number of input characters must exceed the sum of the keepleft value and the keepright value by at least two.
Luhn check failed for input data.	The token format FPE-luhn was specified, but the input data sports an incorrect Luhn checksum. Please check your input data, or change the token format from FPE-luhn to FPE.
After accounting for keepleft and keepright, there are not enough input characters present to successfully tokenize.	• When you use the FPE token format, the number of input characters must exceed the sum of the keepleft value and the keepright value by at least two. • When you use the FPE-luhn token format, the number of input characters must exceed the sum of the keepleft value and the keepright value by at least three.
Token does not begin with the prefix specified in the database.	While a certain token prefix is specified in the token template which is being used in the JSON input, the token seen in the JSON input does not sport this particular token prefix.
The specified token only contains the prefix and no payload.	While the token seen in the JSON input sports the correct token prefix, there are no other characters present in the token.
Luhn check failed for input token.	The token format FPE-luhn was specified, but the token specified sports an incorrect Luhn checksum. Please check your input data.
There are not enough input characters present to successfully detokenize.	• When you use the FPE token format, the number of input characters, not counting the token prefix, must exceed the sum of the keepleft value and the keepright value by at least two. • When you use the FPE-luhn token format, the the number of input characters, not counting the token prefix, must exceed the sum of the keepleft value and the keepright value by at least three.