

Ao

CENTRO DE TECNOLOGIA DE INFORMAÇÃO E COMUNICAÇÃO DO ESTADO DO RIO DE JANEIRO

PREGÃO ELETRÔNICO PARA REGISTRO DE PREÇOS N° 019/2025

ÓRGÃO/ENTIDADE GERENCIADORA (UNIDADE GESTORA – UG: 403200)

PROPOSTA COMERCIAL

Sr. Pregoeiro

A empresa **ARS TECNOLOGIA SERVIÇOS E CONSULTORIA LTDA.** (Neotel Segurança Digital), inscrita no CNPJ sob o n.º 04.189.909/0001-90, estabelecida na Rua do Rocio n.º 220, Vila Olímpia, São Paulo-SP, Fone/Fax n.º 11-3044-1819, e-mail: licitacao@neotel.com.br, site: www.neotel.com.br, por intermédio de seu representante legal Ângelo Antônio Polizel, portador(a) do RG n.º 23.058.270 e do CPF n.º 073.330.198-37, **APRESENTA SUA PROPOSTA COMERCIAL:**

SERVIÇO PÚBLICO ESTADUAL PRODERJ	PREGÃO ELETRÔNICO - PE - RP N° 019/2025 A realizar-se em: 05/02/2026 às 11:00 horas Processo N°: SEI-430002/001510/2024		
A firma ao lado mencionada propõe fornecer ao PRODERJ - Centro de Tecnologia de Informação e Comunicação do Estado do Rio de Janeiro, pelos preços abaixo assinalados, obedecendo rigorosamente as condições estipuladas no EDITAL PE-RP N° 019/2025	CARIMBO DA FIRMA ARS TECNOLOGIA SERVIÇOS E CONSULTORIA LTDA. (Neotel Segurança Digital), inscrita no CNPJ sob o n.º 04.189.909/0001-90, estabelecida na Rua do Rocio n.º 220, Vila Olímpia, São Paulo-SP, Fone/Fax n.º 11-3044-1819, email: licitacao@neotel.com.br , site: www.neotel.com.br , por intermédio de seu representante legal Ângelo Antônio Polizel, portador(a) do RG n.º 23.058.270 e do CPF n.º 073.330.198-37.  Ângelo A. Polizel CPF 073.330.198-37		
Registro de Preços para a aquisição de console de gerenciamento, agentes de proteção de dados para banco de dados, agentes para compartilhamento seguro de base de dados, agentes de proteção de dados para servidores de arquivo (file servers), agentes de proteção de dados para contêineres, agentes de proteção de dados para aplicação, agentes para gestão de chaves local, e	U.F.	QTDE	VALOR (R\$)

prestação dos serviços de plataforma de anonimização e gerenciamento de chaves criptográficas, controle de acesso, visibilidade e rastreabilidade de utilização de dados em servidores de arquivos, banco de dados, infraestrutura em nuvem (IaaS), plataforma como serviço (PaaS) e software como serviço (SaaS), incluindo serviço de instalação, configuração da solução, integrações necessárias com soluções de terceiros e serviços especializados, conforme Anexo I (Especificações Técnicas do Objeto) deste documento, bem como treinamentos operacionais das soluções contratadas, voltados para os técnicos da CONTRATANTE, conforme especificações e quantidades estabelecidas no Termo de Referência e nos requisitos da contratação descritos nos Estudos Técnicos Preliminares.						Com ICMS		Sem ICMS	
LOTE ÚNICO	Item	ID-SIGA	Descrição			Unitário	Total	Unitário	Total
	1	169955	CONSOLE DE GERENCIAMENTO centralizado em alta disponibilidade, incluindo licenciamento permanente, instalação e configuração	UN	10	-	-	R\$ 830.000,00	R\$ 8.300.000,00
	2	169956	Serviço de manutenção, atualização de versão e garantia para CONSOLE DE GERENCIAMENTO	SERVIÇO	12	-	-	R\$ 15.235,20	R\$ 2.193.868,80
	3	169957	AGENTES DE PROTEÇÃO DADOS PARA BANCO DE DADOS, incluindo licenciamento permanente, instalação e configuração	UN	29	-	-	R\$ 213.500,00	R\$ 6.191.500,00
	4	169958	Serviço de manutenção, atualização de versão e garantia para AGENTES DE PROTEÇÃO DADOS PARA BANCO DE DADOS	SERVIÇO	29	-	-	R\$ 3.550,08	R\$ 1.235.427,84
	5	169959	AGENTES PARA COMPARTILHAMENTO SEGURO DE BASE DE DADOS, incluindo licenciamento permanente, instalação e configuração	UN	6	-	-	R\$ 711.696,00	R\$ 4.270.176,00
	6	169960	Serviço de manutenção, atualização de versão e garantia para AGENTES PARA COMPARTILHAMENTO SEGURO DE BASE DE DADOS	SERVIÇO	6	-	-	R\$ 11.851,68	R\$ 853.320,96
	7	169961	AGENTES DE PROTEÇÃO DE DADOS PARA SERVIDORES DE ARQUIVO (FILE SERVER), incluindo licenciamento permanente, instalação e configuração	UN	13	-	-	R\$ 201.979,20	R\$ 2.625.729,60

8	169962	Serviço de manutenção, atualização de versão e garantia para AGENTES DE PROTEÇÃO DE DADOS PARA SERVIDORES DE ARQUIVO (FILE SERVER)	SERVIÇO	13	-	-	R\$ 3.878,40	R\$ 605.030,40
9	169963	AGENTES DE PROTEÇÃO DE DADOS PARA CONTÊINERES, incluindo licenciamento permanente, instalação e configuração	U N	13	-	-	R\$ 199.463,52	R\$ 2.593.025,76
10	169964	Serviço de manutenção, atualização de versão e garantia para AGENTES DE PROTEÇÃO DE DADOS PARA CONTÊINERES	SERVIÇO	13	-	-	R\$ 4.075,20	R\$ 635.731,20
11	169965	AGENTES DE PROTEÇÃO DE DADOS PARA APLICAÇÃO, incluindo licenciamento permanente, instalação e configuração	U N	11	-	-	R\$ 235.728,48	R\$ 2.593.013,28
12	169966	Serviço de manutenção, atualização de versão e garantia para AGENTES DE PROTEÇÃO DE DADOS PARA APLICAÇÃO	SERVIÇO	11	-	-	R\$ 3.787,20	R\$ 499.910,40
13	170934	Subscrição de agentes para gestão de chaves na nuvem, incluindo instalação e configuração - Termo de licenciamento por 12 (doze) meses	U N	4	-	-	R\$ 768.331,68	R\$ 3.073.326,72
14	169969	AGENTES PARA GESTAO DE CHAVES LOCAL, incluindo licenciamento permanente, instalação e configuração	U N	9	-	-	R\$ 17.520,00	R\$ 157.680,00
15	169970	Serviço de manutenção, atualização de versão e garantia para AGENTES PARA GESTAO DE CHAVES LOCAL	SERVIÇO	19	-	-	R\$ 364,80	R\$ 83.174,40
16	170384	Subscrição de agentes para descoberta e classificação de dados, incluindo instalação e configuração. Termo de Licenciamento por 12 (doze) meses (Franquia 50 TB)	U N	4	-	-	R\$ 1.762.267,20	R\$ 7.049.068,80
17	169975	Serviço de treinamento para solução de gerenciamento de chaves criptográficas	VAGA (aluno)	37	-	-	R\$ 6.624,00	R\$ 245.088,00

18	170683	Serviço de operação assistida para solução de gerenciamento de chaves criptográficas	UST	23.404	-	-	R\$ 548,64	R\$ 12.840.370,56
VALOR TOTAL GLOBAL DO LOTE					-	-	-	R\$ 56.045.442,72

Valor global por extenso: cinquenta e seis milhões, quarenta e cinco mil, quatrocentos e quarenta e dois reais e setenta e dois centavos.

Dados do Banco Bradesco do Licitante: Banco Bradesco

Agência (Nome/n°): 1133

Conta Corrente: 11387-5

OBSERVAÇÕES

1ª - A PROPOSTA DE PREÇOS apresentada foi:

- preenchida integralmente por processo mecânico ou eletrônico, sem emendas e rasuras;
- contém os preços em algarismos e por extenso, por unidade, já incluídas as despesas de fretes, impostos federais ou estaduais e descontos especiais; e
- é datada e assinada pelo gerente ou seu procurador.

2ª - O Proponente se obrigará, mediante devolução da PROPOSTA DE PREÇOS, a cumprir os termos nela contidos.

3ª - A licitação poderá ser anulada no todo, ou em parte, de conformidade com a legislação vigente.

• **Prazo de execução:** Conforme o TR.

• **Validade da PROPOSTA:** Preços válidos por 60 (sessenta) dias.

• **Local de entrega desta Proposta:** Rua da Conceição nº 69/24º andar - Centro - Rio de Janeiro - RJ.

Declaramos inteira submissão ao presente termo e legislação vigente.

Em 05 de fevereiro de 2026.

ANGELO ANTONIO

POLIZEL:0733301983

7

Assinado de forma digital por
ANGELO ANTONIO
POLIZEL:07333019837
Dados: 2026.02.05 15:53:28 -03'00'

Firma proponente - Assinatura do Responsável

MARCA E FABRICANTE DAS SOLUÇÕES: THALES

Os preços contemplam todos os custos de acordo com as condições estabelecidas no Termo de Referência.

São Paulo, 05 de fevereiro de 2026.

ANGELO ANTONIO

POLIZEL:07333019837

Assinado de forma digital por

ANGELO ANTONIO

POLIZEL:07333019837

Dados: 2026.02.05 15:53:50 -03'00'

ARS TECNOLOGIA SERVIÇOS E CONSULTORIA LTDA.

CNPJ 04.189.909/0001-90

(Neotel Segurança Digital)

Angelo Antônio Polizel - RG: 23.058.270 - Sócio-Diretor



CipherTrust Manager

Version: v2.22 (latest)

30 January 2026

Contents

CipherTrust Manager Administration	5
CipherTrust Manager Release Model	7
Supported Upgrade Paths	10
Authentication	15
Authentication Settings for NAE, KMIP, Web, and Preboot Interfaces	19
Users	58
Certificate Based Authentication for Users	85
Managing Your Own User Account Settings	90
Authentication Tokens	93
Client Management	102
Authenticating to the REST API	147
Attribute-based Access Control Permissions for Resources	151
Backup	152
Certificate Authority	153
Client Management	154
Cluster	155
Keys	156
Connection Managers	157
Proxies & Allowed Proxies	164
DNS Hosts	164
Domains	164
Groups & Groupmaps	165
Interfaces	166
Connections	167
Logs	167
Log Forwarders	168
Quorum	168
Records	169
Users	170
Scheduler	170
Servers	171
SNMP	171
Policies	172
Migrations	172
Licensing	173
HSM	174
Miscellaneous	174
Network	177
ProtectApp	177
Secrets	178
Password Policy	178
Key Policies	192

Changing Passwords	197
Domain Management	198
Groups	212
Group Mapping	230
Services	233
Root of Trust Hardware Security Module	236
Clusters and Nodes	268
CipherTrust Manager System Monitoring	296
Records	298
Alarms	312
Syslogs	315
Debug Logs	319
Log Forwarding	324
Prometheus Metrics Endpoint	332
Policies	340
Banners	357
Basic Interface Configuration	360
Proxy Configuration	376
Quorums	384
SNMP	414
SMTP	421
Backups	429
Scheduling Backups	466
Disk Encryption After Initial Deployment	474
Scheduler	478
Connection Manager	491
Connections	492
Labels in Connection Manager	496
AWS	498
Google	511
Secure Copy Protocols (SCP/SFTP)	518
Azure	527
Hadoop Knox	539
Loki	554
Salesforce	561
Luna Network HSM	578
Elasticsearch	600
SAP Data Custodian	607
CIFS/SMB	618
Syslog	625
Oracle Cloud Infrastructure	634
DSM Connection	641
OIDC	657
LDAP	662
Attestation Authority Connection	672

CM Connection/External CM	673
Managing CM Connections using GUI	674
Configuring CM Connections using ksctl	678
Managing External CM Server Connections using ksctl	684
Managing CM Connections using ksctl	692
Akeyless Gateway	701
Browsing LDAP Users and Groups	706
System Upgrade/Downgrade	713
Configuring DNS Hosts	721
CLI Toolkit	725
CLI Toolkit Installation	726
Batching Commands with Tokens	734
Example CLI User Set Up	737
Support CLI	743
Keys	745
Key Rotation	769
Crypto Operations	773
MKEK Rotation	778
Certificate Authority	781
System Info	822
System Properties	824
Network Diagnostics	833
Integrating Logging with Splunk App	837
Return Material Authorization (RMA) Guidance	848
Secrets Management	851
Reports	852
Key Templates	862
Managing public SSH keys for ksadmin	867

CipherTrust Manager Administration

The CipherTrust Manager is built on prevailing cloud-based technologies, providing a cloud friendly, key management solution. It employs a REST interface and a microservice based architecture, allowing for easy deployment and scalability within your environment. With this new architecture, the CipherTrust Manager simplifies administration, helps ensure compliance, and maximizes security by providing centralized management of keys, policies, and essential functions.

Product references made in this document

CipherTrust Manager - refers to all platforms running the CipherTrust Manager software.

Virtual CipherTrust Manager k170v and Virtual CipherTrust Manager k470v (or simply k170v and k470v) - refer specifically to virtual and private cloud instances running the CipherTrust Manager Software.

- k170v is intended for simplified and centralized key management. For example, k170v is suited to lab environments, low transaction encryption use cases, and storage encryption key management with KMIP (Key Management Interoperability Protocol). k170v allows for the usage of four CPUs or fewer. k170v is applied as a [license](#).
- k470v is intended to support high transaction-per-second encryption operations as is typically required from the CipherTrust Data Security Portfolio suite. k470v allows for the usage of more than four CPUs. k470v is applied as a [license](#).

CipherTrust Manager k470, k570, and k160 Appliances (or simply k470, k570, and k160) - refers specifically to the k470, k570, and k160 physical appliances running the CipherTrust Manager Software.

SafeNet KeySecure Classic (or simply KeySecure Classic) - refers to the previous generation of SafeNet KeySecure Appliances (models k450 and k460). You can [migrate data](#) from these appliances to CipherTrust Manager k470 and k570 physical appliances.

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.

It is assumed that users of this document are proficient with security concepts.

Key Management

CipherTrust Manager offers robust capabilities for managing cryptographic keys across their lifecycle, including key generation, key import and export, and key rotation. With the CipherTrust Manager, all cryptographic keys are stored in a centralized, hardened environment to simplify administration while ensuring tight security for the broadest array of data types.

High Performance

Customers can deploy multiple CipherTrust Manager appliances in a clustered configuration with real-time replication of keys, policies, and configuration information across multiple appliances - enabling complete disaster recovery and business continuity. For large distributed enterprises that use multiple encryption solutions, keys can be centrally managed without making any perceptible impact on system performance.

Broad Flexibility

CipherTrust Manager offers key management capabilities that can be integrated with virtually any commercial encryption product. Supported technologies include:

- SafeNet Luna Network HSM, TCT Luna T-Series Network HSM, and AWS HSM partitions
- Application encryption
- Database encryption, including native database encryption
- File and storage level encryption solutions

The CipherTrust Manager supports a wide range of open standard cryptographic interfaces, including PKCS #11, JCE, and .NET. The CipherTrust Manager also supports the Key Management Interoperability Protocol (KMIP). Further, customers and partners can take advantage of the REST interface to develop their own custom software utilizing the enterprise key management functionality of the CipherTrust Manager.

Robust Security

CipherTrust Manager offers a range of robust security features:

- Granular Attribute Based Access Control (ABAC) authorization capabilities
- Secure key distribution through support of TLS
- Secure storage of key encryption keys on a SafeNet Luna Network HSM

Key Interfaces

CipherTrust Manager's NAE-XML interface is an adapter layer that translates XML requests into CipherTrust Manager's REST API. However, the REST API is structured differently than NAE-XML. The functions and configuration of CipherTrust Manager is modeled as a set of resources that can be managed via the REST API. The REST API online documentation goes into more detail about the specific attributes of the resources, and what operations you can perform on them.

Note

The CipherTrust Manager also includes a CLI toolkit, named **ksctl**, that can be downloaded and run locally to perform management functions. **ksctl** exclusively uses the REST API to communicate with the CipherTrust Manager. Many examples make use of **ksctl**. For more information on **ksctl**, go to [CLI toolkit](#).

CipherTrust Manager Release Model

CipherTrust Manager distributes new firmware releases regularly on a predictable schedule to give customers the benefit of new functionality and bug fixes while allowing time to plan upgrades.

Version Numbering

CipherTrust Manager firmware releases follow [semantic versioning conventions](#). Version numbering format is <MAJOR>.<MINOR>.<PATCH>.

Most releases are minor versions and patches on minor versions, to introduce new functionality without breaking backward compatibility.

Release Schedule and Support Time Frame

Starting with the 2.11.x release, a minor version is designated either a **Long Term Support (LTS) Release** or a **Feature Release (FR)**.

The following table summarizes the release schedule and support time frame differences between the two release types:

Release Characteristic	Future LTS versions	FR version
Technical Support Timeframe	Four years	Two years
Patches	Four scheduled patches for two years	As needed for the current latest FR version only
Best strategy to fix bugs	Upgrade to scheduled patches	Upgrade to the next minor release
CipherTrust Manager model support	Every CipherTrust Manager model is supported	CipherTrust Manager k160 supported only on selected FR versions

- **Long Term Support Releases** are enterprise-grade releases focused on stability. The CipherTrust Manager interfaces display the version with the suffix **LTS**. Future LTS releases will have the following characteristics:
 - Each LTS release will have four scheduled patches, which include the latest security fixes and other critical or low-risk functional fixes. These patches will be released twice a year for the two years following the initial release.
 - Technical support continues for four years past the original release. This support consists of updates to the [support portal knowledge center](#), and availability of Thales technical staff to help troubleshoot problems. After the first two years, Thales will no longer issue further patches or bug fixes, and will provide technical support only.

Note

We are currently determining the schedule and designation for the next LTS release. We will update this page with that information as soon as it is finalized.

- **2.11.x-LTS Release** is the first LTS release. We have made some adjustments the standard LTS release model for 2.11.x.
 - Scheduled patches are released twice a year, continuing until Q4 2026.

Note

2.11.6 was an unscheduled, hot fix release.

- Technical support, consisting of updates to the [support portal knowledge center](#), and availability of Thales technical staff to help troubleshoot problems, extends four and a half years into Q3 2027.
- **Feature Releases** are production-quality releases providing early access to new features for evaluation, released three times a year.

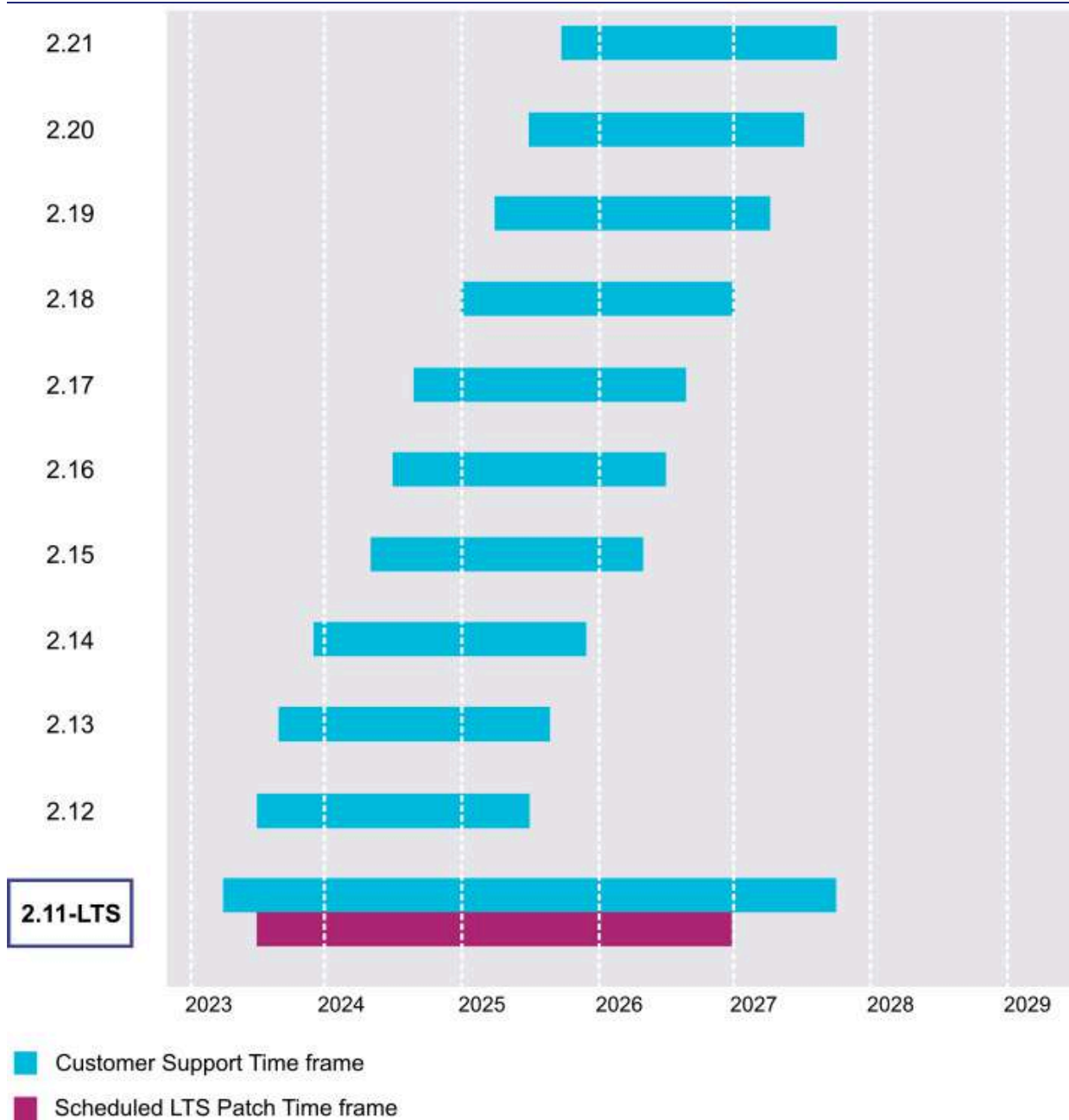
Patches for FR versions are created as needed for security fixes ranked as critical or major and bug fixes ranked as critical. If fixes are qualified for a patch, one is issued for the current latest FR.

Note

If you wish to apply non-critical bug fixes to an FR version, upgrade to the next minor version.

Technical support continues for two years past the original release, until the end of the same quarter for that year. This support consists of updates to the [support portal knowledge center](#), and availability of Thales technical staff to help troubleshoot problems.

The following diagram demonstrates support and scheduled patch time frames for the current planned LTS and FR versions:



Versions 2.10.x and older will reach end of support at the end of Q3 in 2025.

CipherTrust Manager k160 Release Model

Most CipherTrust Manager models are supported with every release. The CipherTrust Manager k160 will be supported on all future LTS releases. Support in FR versions is determined per release in order to offer early access to features specific to the k160 platform. Consult the release notes for an FR version to confirm whether k160 is supported.

Release Distribution and Upgrade

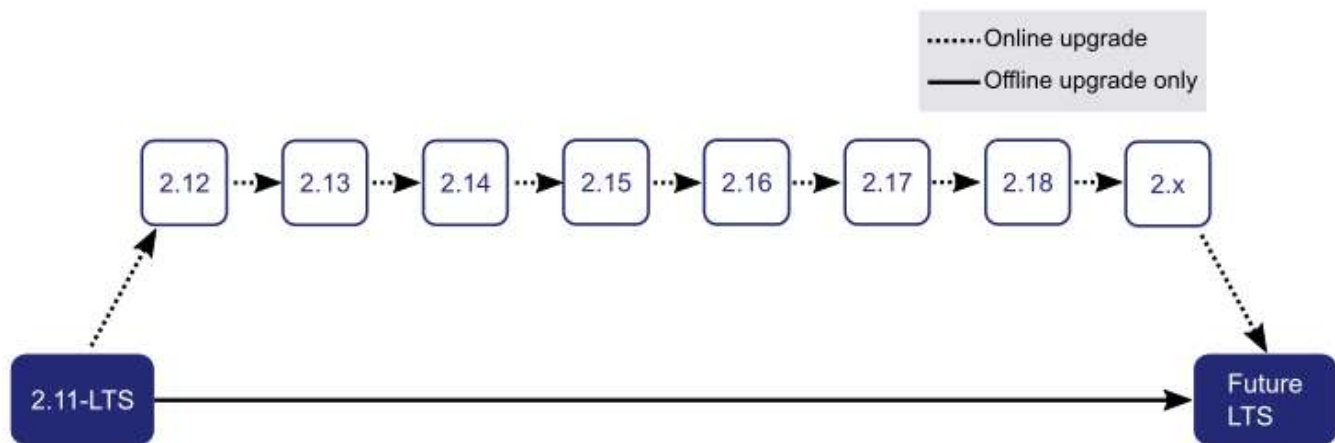
All minor releases are available as an upgrade on physical and virtual appliances, and as new deployments of Virtual CipherTrust Manager on supported public and private clouds. Consult release notes for a specific version

for more details. Minor releases contain new features and bug fixes for issues of any severity.

For new minor versions, x.y.0, upgrade paths and available upgrade methods for a cluster vary depending on **LTS** or **FR** designation. You can only upgrade to FR versions from the immediate previous release. For example, you are only able to upgrade a cluster directly from 2.12.x to 2.13.x. You will be able to upgrade directly to LTS versions from previous LTS versions. For example, you will be able to upgrade from 2.11.x to the next LTS version.

You can exercise an online in-place cluster upgrade, where the CipherTrust Manager remains available for requests during upgrade, from the immediate previous release. You can exercise the offline in-place cluster upgrade to skip intermediate versions if you are moving from an LTS release to an LTS release. For offline in-place cluster upgrade, the CipherTrust Manager requires true downtime and does not allow requests from connector or client applications. Refer to the [System Upgrade/Downgrade](#) page of the target release version for steps to exercise the upgrade.

The following diagram demonstrates upgrade paths for clusters between minor releases:



Scheduled patches for LTS releases are available as cumulative upgrades. You can upgrade from any LTS version to the latest patch for that LTS minor version. For example, if there are x.y.1, x.y.2, and x.y.3 patches released for the minor LTS version, you can upgrade from x.y.0 or x.y.1 to x.y.3 directly without having to pass through any intermediate versions.

Thales intends to make the latest LTS release available as an option for physical k570, k570, and k160 appliance orders.

Note

As of February 2023, you can order physical appliances with either 2.9.x or 2.0 pre-installed. These versions existed before Long Term Support and Feature Release designation.

Patches for FR versions are usually upgrade only. In some cases, new deployments for Virtual CipherTrust Manager are also made available on supported public and private clouds. Consult release notes for a specific version for supported upgrade paths.

Supported Upgrade Paths

Supported upgrade paths are presented for the latest Feature Release (FR) and Long Term Support (LTS) CipherTrust Manager software versions. Consult the [release model page](#) for details about these release types.

Release Notes for individual release versions also contain supported upgrade paths to that version, alongside any special instructions or considerations.

Upgrade path support follows some general patterns, with exceptions identified for particular cases:

- You can upgrade a standalone, unclustered appliance up to three minor versions at one time.
- You can upgrade cluster nodes in the cluster remove/rebuild method up to three minor versions at one time.
- You can upgrade cluster nodes in the in-place online cluster upgrade up to one minor version at one time.

Note

An important exception is that you cannot use this method to upgrade from 2.21.x to 2.22.x.

- Upgrade to the latest patch of the minor version for any starting, intermediate, or destination minor version.

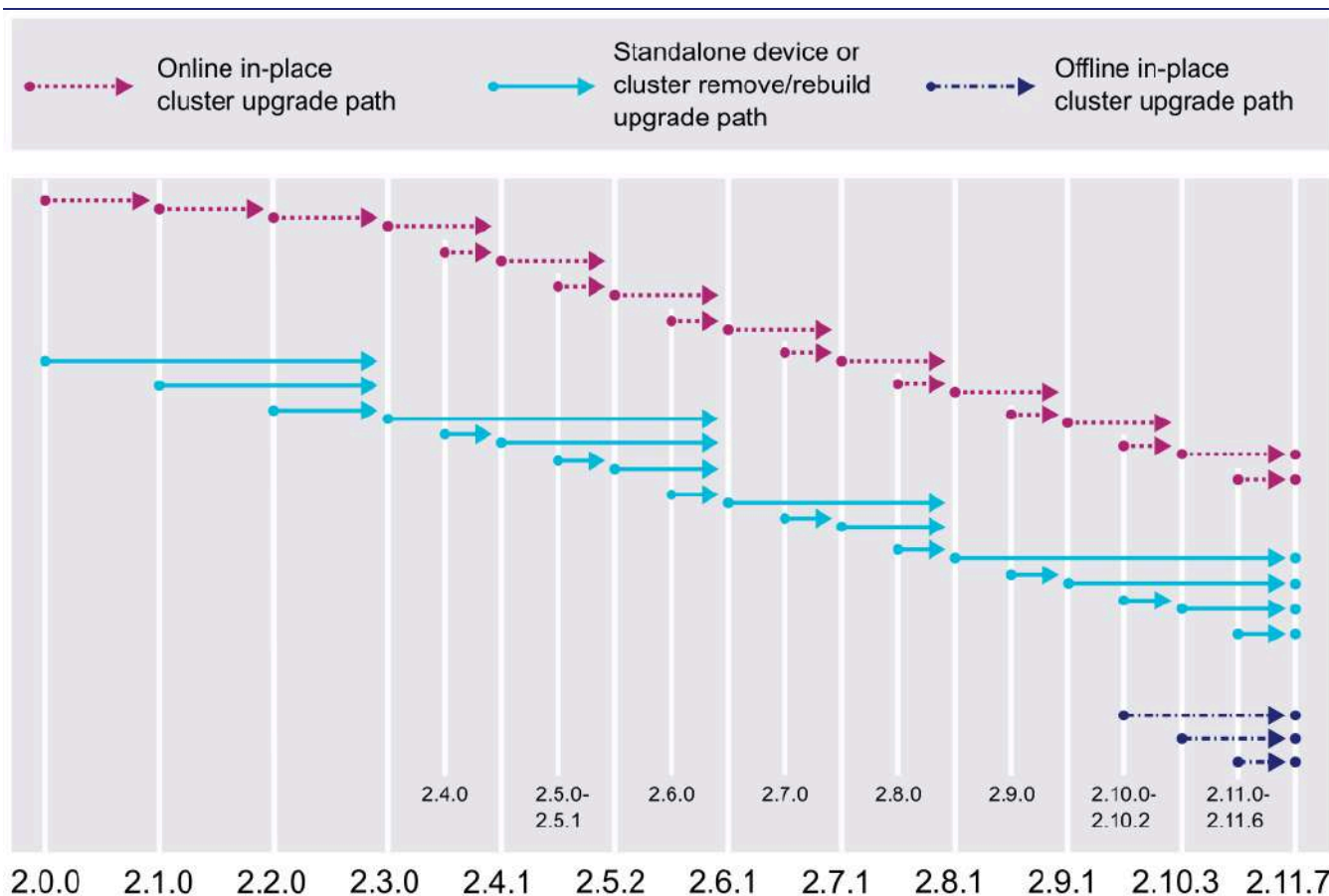
Upgrade to Latest Long Term Support Version

[Upgrade procedures for latest LTS Version](#) are included in software documentation for that version.

The supported cluster upgrade methods to LTS releases are online in-place, offline in-place, and cluster remove/rebuild. These paths are valid for the CipherTrust Manager k470 and k570 physical appliances, and Virtual CipherTrust Manager k170v and k470v instances.

Note

- Upgrade paths for the Thales TCT k160 small form factor appliance are not presented here. Consult [k160 Upgrade Path for Latest Supported Release](#) for 2.11.2-TCT to the latest supported release.
- If you are starting at a version below 2.8.x, you must upgrade to 2.8.1 before upgrading to higher versions.
- Upgrade to 2.9.2 is not recommended. If you are already on the 2.9.2 patch, you can upgrade to 2.10.3 with a standalone device or with the cluster rebuild/remove method.



Upgrade to This Release

Upgrade procedures are included in software documentation for that version.

The supported cluster upgrade methods to FR versions are online in-place and cluster remove/rebuild.

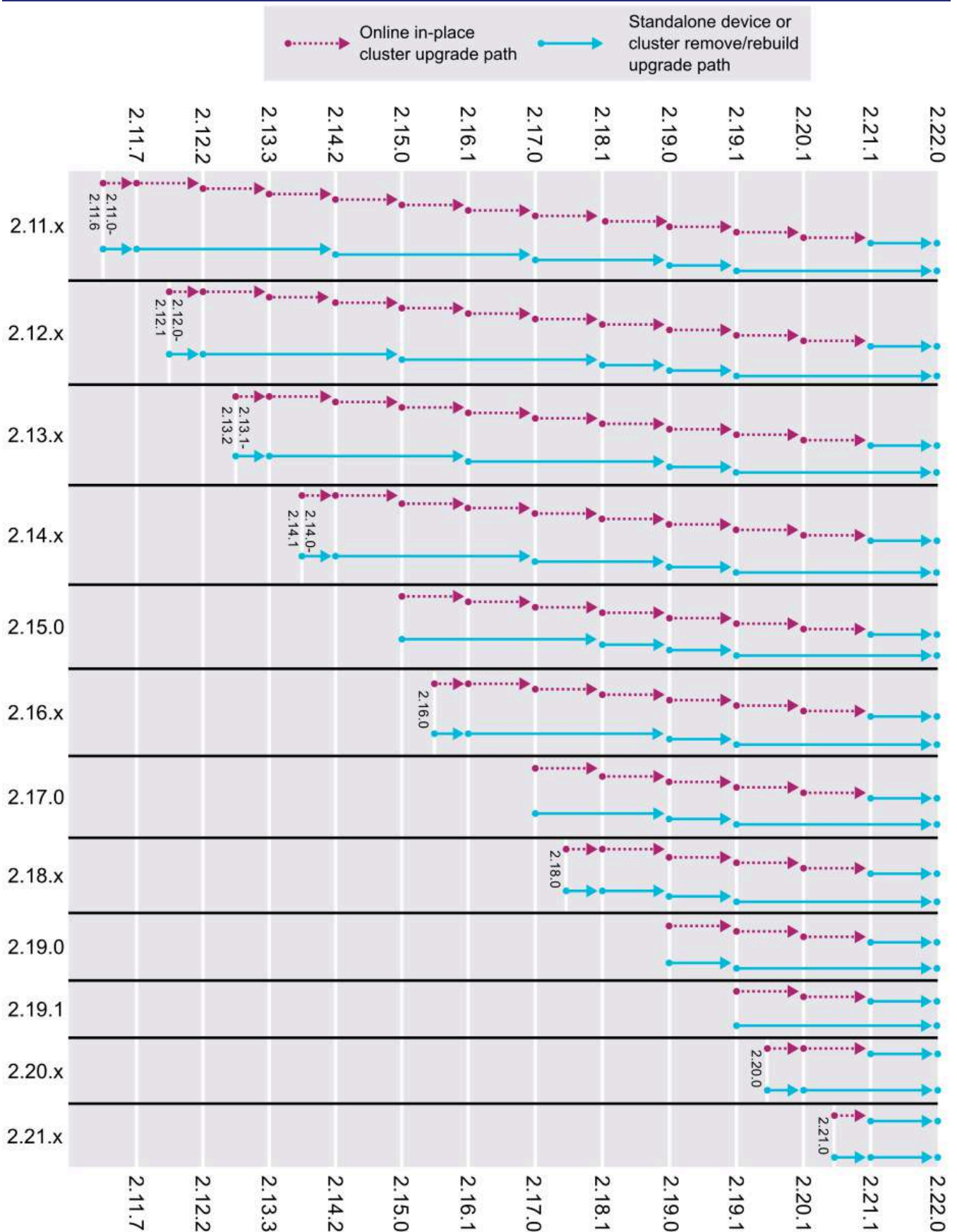
Note

- Upgrade paths for the Thales TCT k160 small form factor appliance are not presented here. Consult [k160 Upgrade Path for Latest Supported Release](#) for 2.11.2-TCT to the latest supported release.
- This section presents upgrade paths from 2.11.x to 2.22. It is not intended to present paths to any earlier destination version. Consult [Upgrade to Latest Long Term Support Version](#) to upgrade from 2.0.x-2.10.x to 2.11.x. Consult release notes for upgrade paths to any other destination version.
- Some upgrades require the force flag, `-o`, using the command `sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path> -o` to upgrade. This is required for the following upgrades:
 - 2.11.7 to 2.12.2 - cluster online in-place

- 2.11.7 to 2.14.2 - standalone or cluster remove/rebuild
- 2.12.2 to 2.15.0 - standalone or cluster remove/rebuild
- 2.18.1 to 2.19.0 - cluster online in-place
- To upgrade 2.12.2 to 2.15.0 or 2.11.7 to 2.14.2 with a standalone device, or with the cluster rebuild/remove method, you must use the force flag, `-o`. The upgrade command is `sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path> -o`.
- Cluster online in-place upgrade is not available for upgrade to 2.22.0. We recommend using cluster online in-place to upgrade up to 2.21.1, and then cluster remove/rebuild for the final upgrade to 2.22.0.

These paths are valid for the CipherTrust Manager k470 and k570 physical appliances, and Virtual CipherTrust Manager k170v and k470v instances.

The CipherTrust Manager k160 small form factor appliance has different supported upgrade paths than all other models, presented [below](#).



k160 Upgrade Path for Latest Supported Release

The CipherTrust Manager k160 support for FR versions is determined per release in order to offer early access to features specific to the k160 platform. Consult the release notes for an FR version to confirm whether k160 is supported.

For releases leading up to the next LTS release, only the validated upgrade paths below are supported:

- 2.11.2-tct to 2.15.0 to 2.16.x to 2.20.x to 2.21.x to 2.22.x
- 2.11.2-tct to 2.15.0 to 2.16.x to 2.20.x to 2.22.x
- 2.11.2-tct to 2.15.0 to 2.20.x to 2.22.x



Due to the specifics of the CipherTrust Manager k160 platform, there are a number of advisories to be mindful of when performing system upgrades:

- Always use the [cluster remove/rebuild method](#) for performing system upgrades.
- Due to a known issue [CM-2](#), if performing a system reset at versions 2.11.2-tct or 2.15.x, do not re-register token HSMs until upgrades to v2.16.x or higher are complete.
- Certain error messages may appear on the console output while the system upgrade process proceeds. These messages are normal and can be ignored. Contact Thales TCT Support if the process completes and reports something other than successful.
- Only the Standard model k160 can be upgraded to 2.22.x. The High Assurance (HA) model is not supported with this release.

Authentication

Two entities authenticate to CipherTrust Manager web console UI, ksctl CLI and REST API: clients and users. Clients are applications that access CipherTrust Manager keys as needed to perform cryptographic or key management operations. Users are people who access CipherTrust Manager to perform configuration and key management tasks manually.

Note

The `ksadmin` user is a specialized operating system administrator that can perform infrequent configuration tasks at the operating system level. `ksadmin` does not authenticate to the web console UI, `ksctl` CLI, or REST API. It only authenticates to the console using SSH key or using password authentication on a physical server console port.

Any cryptographic or key management operation occurs over a client, and CipherTrust Manager always attempts to identify the client for these operations, and adds the client identity to the [audit records](#) if identified. Client authentication occurs when a client identity is found.

CipherTrust Manager authenticates users when a user identity is presented with the request. This authentication provides a mechanism to enforce permissions on a user, and an audit trail of a user's activities.

It is possible to only provide a client identity and not a user identity, in which case the client is the authenticated entity and its identity appears in the audit record. This authentication is appropriate for an automated client or a service account that requires no human interaction.

When a user identity is presented, with or without a client identity, only the user is authenticated. The client's actions are assumed to be performed on behalf of a user.

CipherTrust Manager authenticates a user or client, CipherTrust Manager checks the user or client's [group membership](#), and applies the permissions associated with those groups.

There are three [interfaces](#) which CipherTrust Manager users and clients authenticate to:

- NAE
- KMIP
- Web. This comprises UI, CLI, and REST API access.

Client and User Identities

There are three types of clients:

- **unregistered clients** - are not registered with the CipherTrust Manager.
- **public clients** - are `ksctl`, NAE, Web-UI, and API playground. These clients are pre-registered with the CipherTrust Manager.
- **confidential clients** - are registered and able to securely authenticate with the CipherTrust Manager. For example, NAE, KMIP, and CTE clients. We recommend this client setting wherever possible.

Public clients and confidential clients have a client identity. The client identity is in the JSON Web Token (JWT) included in every request, so that CipherTrust Manager recognizes the client as a registered identity. Unregistered clients are allowed to make requests without an associated client identity.

Client identities are recorded in the [audit records](#).

Only confidential clients can be authenticated. This is because public clients cannot store secrets, and CipherTrust Manager cannot validate secrets for an unregistered client.

User identities are tied to a user name, and can be authenticated through password or certificate. CipherTrust Manager authenticates local users directly, while an external identity provider authenticates Open ID Connect (OIDC) and Lightweight Directory Access Protocol (LDAP) users. LDAP and OIDC users have an associated user name on CipherTrust Manager, and their user identities are recorded in the [audit records](#), as local user identities are.

You can control what client types a user can communicate through. By default, all three client types are allowed.

User Delegation and User Impersonation for KMIP and NAE

The KMIP and NAE interfaces require user authentication for backward compatibility with existing permissions configuration. KMIP and NAE clients, however, often operate in an automated manner without human interaction.

There are currently two ways to fulfill the user authentication requirement for a client:

- User delegation - This is providing a user's credentials or authorization grant when the client starts a new session with CipherTrust Manager.
- User impersonation - This is connecting with a client certificate containing a user's name in a defined field.

To enforce user delegation on an interface, set the following [interface modes](#):

- `tls-pw-req` or "TLS, verify client cert, user must supply password"
- `tls-cert-and-pw` or "TLS, verify client cert, password is needed, user name in cert must match user name in authentication request"

To set user impersonation on an interface, set the following [interface mode](#):

- `tls-cert-pw-opt` or "TLS, verify client cert, user name taken from client cert, auth request is optional"

LDAP and OIDC Connection Types

CipherTrust Manager supports LDAP and OIDC for authentication using dedicated connections for particular scenarios.

- An [LDAP connection in the Connection Manager](#) service for secure storage of credentials to authenticate CipherTrust Manager users.
- A [LDAP connection in the Access Management](#) service to authenticate CipherTrust Manager users. This connection is separate and additional to the Connection Manager LDAP connection.
- An [OIDC Connection in Connection Manager](#) to authenticate CTE Agents.
- A [dedicated OIDC connection in Access Management](#) to authenticate CipherTrust Manager users.

Note

If you create an OIDC connection through Access Management, a duplicate Connection Manager OIDC connection is visible, with no product. This duplicate is created for secure storage of credentials. We strongly recommend managing OIDC connections to authenticate CipherTrust Manager users through the Access Management UI menus and `ksctl connections` commands, and not Connection Manager.

Client Registration

When a client is successfully [registered](#), the client is automatically added to the CipherTrust Manager, a client identity is assigned, and activity performed using that client identity can be monitored through records. The client becomes a confidential client.

ProtectFile, CTE, CipherTrust Data Protection Gateway (DPG), and KMIP clients need to be registered with the CipherTrust Manager to successfully authenticate and perform key and cryptographic operations. CipherTrust Manager currently allows unregistered clients of other types.

Client registration is supported with:

- CTE Agents
- CTE Userspace Agents
- CTE for Kubernetes Agents
- DPG
- Custom KMIP clients
- Custom NAE clients
- Custom REST API clients

Suggested Authentication for Users and Clients

- Require login with a user identity to any action performed manually by a person. This is already required and enforced for accessing `ksctl` CLI tool and CipherTrust Manager GUI.
- Set the following [interface modes](#) for NAE and KMIP. They are more secure, as they allow two factor authentication.
 - `tls-pw-req` or "TLS, verify client cert, user must supply password"
 - `tls-cert-and-pw` or "TLS, verify client cert, password is needed, user name in cert must match user name in authentication request"

- Register all clients that are possible to register, to assign a client identity.
- Register custom REST clients which are automated or represent a service account that does not require human interaction. Use the client identity to issue API tokens for such clients.

Authentication Settings for NAE, KMIP, Web, and Preboot Interfaces

The Web, KMIP, and NAE interfaces have several options to control authentication to those interfaces.

Note

Refer to [Basic Interface Configuration](#) for the following operations:

- Viewing interfaces
- Enabling/disabling interfaces
- Adding/removing interfaces
- Changing an interface web port

Effect of updating interface settings

Warning

If you have active NAE, KMIP, or web connections, we recommend that you plan for connection downtime before updating interface settings, especially for updates to the initial default interfaces.

Note

- When you update the port of the **default interface** of NAE, KMIP, and WEB, all nodes in the cluster restart. However, if you update any other setting apart from the port, no node in the cluster restarts.
- If you update any setting on the **non-default interfaces** of NAE, KMIP, and WEB, no node in the cluster restarts.

There are some interface changes that are applied immediately and trigger an automatic services restart. This restart can disrupt running NAE, KMIP, and web connections and cause an immediate downtime of a few minutes. If your CipherTrust Manager is part of a [cluster](#), the interface settings change can also replicate to other nodes, and disrupt running NAE, KMIP, and web connections to those nodes. Therefore, plan for some downtime before

updating interfaces.

Interface settings changes that are known to cause an immediate loss of connection are:

- Changing the [interface mode](#)
- Changing any TLS-related setting, including updating minimum/maximum TLS version, trusted CAs, enabled cipher suites, uploading a new server certificate, changing any option for auto-generating the server certificate.
- Changing the [interface default connection](#) between LDAP or local accounts.

Note

For CCKM DKE TLS connection to work, a dedicated prefix `dke.` for hostname is introduced. This hostname communicates with the server without verifying the SSL client certificate. All hostnames mentioned in the SANs of the server certificate start with `dke.` will not verify the client certificates, that is, the `ssl_verify_client` will be disabled. With this setting disabled, the certificate-based authentication will not work.

Other interface changes require a manual [Services Restart](#).

SSL/TLS supported protocols and cipher suites

CipherTrust Manager allows the customers to customize the cipher suites for TLS handshake for the following interfaces:

- NAE
- KMIP
- Web

A cipher suite is a set of algorithms that help secure a network connection that uses Transport Layer Security (TLS). The set of algorithms that cipher suites usually contain include: a key exchange algorithm, an authentication algorithm, a bulk encryption algorithm, and a message authentication code (MAC) algorithm.

When you add an Interface, it is added with the default recommended list of cipher suites. You can customize them by updating an interface.

Note

The protocol has a higher priority than the cipher suites. In other words, if certain protocol is disabled and its respective cipher suites are enabled, then the cipher suites will not be used for communication.

You can not add/remove any cipher suite, however, they can be enabled/disabled from the given list of cipher suites.

Following table details the priority list of cipher suites for the respective Interfaces in the descending order.

Interface	Cipher Suites Priority List
NAE/KMIP	• TLS_AES_256_GCM_SHA384 (TLSv1.3) • TLS_CHACHA20_POLY1305_SHA256 (TLSv1.3) • TLS_AES_128_GCM_SHA256 (TLSv1.3) • - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) • - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (disabled) • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (disabled) • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (disabled) • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (disabled) • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (disabled) • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (disabled) • TLS_RSA_WITH_AES_128_CBC_SHA256 (disabled) • TLS_RSA_WITH_AES_256_CBC_SHA (disabled) • TLS_RSA_WITH_AES_128_CBC_SHA (disabled)
Web	• TLS_AES_256_GCM_SHA384 (TLSv1.3) • TLS_CHACHA20_POLY1305_SHA256 (TLSv1.3) • TLS_AES_128_GCM_SHA256 (TLSv1.3) • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-SHA384 (disabled) • ECDHE-ECDSA-AES128-SHA256 (disabled) • ECDHE-RSA-AES256-SHA384 (disabled) • ECDHE-RSA-AES128-SHA256 (disabled) • ECDHE-ECDSA-AES256-SHA (disabled) • ECDHE-ECDSA-AES128-SHA (disabled) • ECDHE-RSA-AES256-SHA (disabled) • ECDHE-RSA-AES128-SHA (disabled)

Note

- TLS 1.3 ciphers suites are not configurable that is, they cannot be disabled. However, you can [disable the TLS protocol as a whole](#).

- The CipherTrust Manager uses an **in-built** order for TLS ciphers, therefore the order of TLS ciphers cannot be changed to set the preference order.
- The cipher suites marked as disabled remain disabled by default as they are considered weak. If required, user can explicitly enable them for use.

If you update the minimum and maximum TLS versions, the cipher suite also gets updated based on the TLS versions. Consider a scenario for a cipher suite where the minimum TLS version is 1.1 and maximum version is 1.3 and the cipher is enabled for TLS 1.1, 1.2, and 1.3. Now, if you update the minimum TLS version to 1.2, ciphers corresponding to TLS 1.1 will be disabled and rest will remain the same.

Following table details the recommended and supported versions of the TLS protocol for the respective Interfaces. The recommended versions are enabled by default.

Interface	Recommended Version	Supported Version
NAE/KMIP	• TLSv1.2 • TLSv1.3	• TLSv1.0 • TLSv1.1 • TLSv1.2 • TLSv1.3
Web	• TLSv1.2 • TLSv1.3	• TLSv1.2 • TLSv1.3

Updating cipher suites using ksctl

To update cipher suite list in an Interface, run:

For Web interface-Syntax

```
ksctl interfaces modify --name web --tls-ciphers-file <json-file>
```

Example Request

```
ksctl interfaces modify --name web --tls-ciphers-file tls-ciphers.json
```

For NAE interface-Syntax

```
ksctl interfaces modify --name nae --tls-ciphers-file <json-file>
```

Example Request

```
ksctl interfaces modify --name nae --tls-ciphers-file tls-ciphers.json
```

For KMIP interface-Syntax

```
ksctl interfaces modify --name kmip --tls-ciphers-file <json-file>
```

Example Request

```
ksctl interfaces modify --name kmip --tls-ciphers-file tls-ciphers.json
```

Format of the JSON File

```
{
  "tls_ciphers": [
    {
      "cipher_suite": "TLS_AES_256_GCM_SHA384",
      "enabled": true
    },
    {
      "cipher_suite": "TLS_CHACHA20_POLY1305_SHA256",
      "enabled": true
    },
    {
      "cipher_suite": "TLS_AES_128_GCM_SHA256",
      "enabled": true
    },
    {
      "cipher_suite": "ECDHE-ECDSA-AES256-GCM-SHA384",
      "enabled": true
    },
    {
      "cipher_suite": "ECDHE-RSA-AES256-GCM-SHA384",
      "enabled": true
    },
    {
      "cipher_suite": "ECDHE-ECDSA-AES128-GCM-SHA256",
      "enabled": true
    },
    {
      "cipher_suite": "ECDHE-RSA-AES128-GCM-SHA256",
      "enabled": true
    },
    {
      "cipher_suite": "ECDHE-ECDSA-AES256-SHA384",
      "enabled": false
    },
    {
      "cipher_suite": "ECDHE-ECDSA-AES128-SHA256",
      "enabled": false
    },
    {
      "cipher_suite": "ECDHE-RSA-AES256-SHA384",
      "enabled": false
    },
    {

```



```

        "cipher_suite": "ECDHE-RSA-AES128-SHA256",
        "enabled": false
    },
    {
        "cipher_suite": "ECDHE-ECDSA-AES256-SHA",
        "enabled": false
    },
    {
        "cipher_suite": "ECDHE-ECDSA-AES128-SHA",
        "enabled": false
    },
    {
        "cipher_suite": "ECDHE-RSA-AES256-SHA",
        "enabled": false
    },
    {
        "cipher_suite": "ECDHE-RSA-AES128-SHA",
        "enabled": false
    }
]
}

```

Example modify TLS version of an interface

Note

- The CipherTrust Manager supports TLS 1.0, TLS 1.1, TLS 1.2, and TLS 1.3.
- TLS 1.0 and TLS 1.1 are deprecated and will be discontinued in a future release.

Note

The TLS 1.3 version is supported from CipherTrust Manager 2.3 onward.

```
ksctl interfaces modify -n <interface name> -e <minimum tls version> -i <maximum
tls version>
```

Here:

- `<interface name>` can be NAE, KMIP, or WEB interface name.
- `<minimum tls version>` can be `tls_1.0`, `tls_1_1`, `tls_1_2`, or `tls_1_3`. Support for `tls_1_0` and `tls_1_1` will be discontinued in a future release.
- `<maximum tls version>` can be `tls_1.0`, `tls_1_1`, `tls_1_2`, or `tls_1_3`. Support for `tls_1_0`

and `tls_1_1` will be discontinued in a future release.

Example

```
ksctl interfaces modify -n kmip_all_5697 -e tls_1_2 -i tls_1_3
```

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",
  "maximum_tls_version": "tls_1_3"
}
```

Restoring to default TLS cipher suite and TLS groups using ksctl

To restore to the default TLS cipher suite of NAE and KMIP Interface, run:

Example Request

```
ksctl interfaces restore-default-tls-ciphers --name nae_all_9009
```

To restore to the default TLS cipher suite and TLS groups of a WEB Interface, run:

```
ksctl interfaces restore-default-tls-ciphers --name web
```

Interface default connection (NAE and KMIP)

NAE and KMIP interfaces may be configured to use a default connection other than `local_account` for username/password authentication and client certificate authentication. Typically this setting is modified to simplify authentication because the username does not need to provide the connection name. For example, when a user

presents their username for authentication without a connection, e.g. `jdoe`, then the CipherTrust Manager uses the default connection of the interface to authenticate the user. If the default connection is configured as `local_account` then the user is authenticated as if `local_account|jdoe` was presented. However, if configured as `myldapdomain` then the user is authenticated as if `myldapdomain|jdoe` was presented.

Example - Interface configuration update for KMIP to use a new default connection

```
ksctl interfaces modify -n kmip -d myldapdomain
```

Response

```
{
  "id": "476618b4-7310-4783-819f-517c387d2281",
  "name": "kmip",
  "mode": "tls-cert-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:0d29d15d-0ddd-444d-8cc5-ffcc85d0928a",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:0d29d15d-0ddd-444d-8cc5-ffcc85d0928a"
    ],
    "external": []
  },
  "createdAt": "2018-07-09T14:49:43.316319Z",
  "updatedAt": "2018-07-09T14:49:50.136975Z",
  "default_connection": "myldapdomain"
}
```

SSL/TLS server certificate (NAE, KMIP, and WEB)

During initial bootstrapping of a new CipherTrust Manager, a new local `KeySecure Root CA` is automatically generated. This CA is later used to issue a server certificate for the interfaces available in the system. All interfaces and services trust this CA by default, meaning that a client certificate issued from this initial `KeySecure Root CA` is automatically trusted by the system.

Note

The certificate duration shouldn't be more than the CA duration. However, if the certificate duration exceeds the CA duration, the certificate duration is automatically set to CA's duration.

Trusting the default server certificate

Clients connecting to an interface examine the presented server certificate and match the signing CA against their trust stores. If the server certificate doesn't match, the client might present an error or close the connection. To avoid this mismatch, you can either add the `KeySecure Root CA` to the client truststore, so that the default server certificate is trusted, or you can upload an external, already trusted server certificate to the CipherTrust

Manager interface.

Using an externally generated server certificate for an interface

The CLI tool (ksctl) can be used to upload an external server certificate for a given interface. You can upload a certificate (full chain including server certificate, intermediate CA, and external root CA) and private key or regenerate a self-signed certificate. The certificates (and the private key) can be uploaded in PEM or Base64 encoded PKCS12 format. The PKCS12 files are encrypted, therefore the `--private-key-password` field is required to use them.

The common name (CN) in the certificate must have a valid value.

Note

Currently, we do not support encrypted PKCS8 keys embedded within a PEM file.

To use an externally generated server certificate for an interface

1. Add the external server certificate in PEM or PKCS12 format to CipherTrust Manager's list of external CAs.

```
ksctl ca externals upload --cert-infile mycert.pem
```

2. View the identifier for the uploaded external CA.

```
ksctl ca externals list --cert-infile <ca-file-name>
```

3. Look for the `id` field and copy this value. It is the `<external-CA-identifier>` in the next step.
4. Change the Trusted CA for the desired CipherTrust Manager interface (NAE in this example).

```
ksctl interfaces modify --name nae --trusted-external-cas <external-CA-identifier>
```

5. Disable auto-generation of the server certificate for the desired CipherTrust Manager interface (nae in this example). This command sets the auto-generation local CA to an empty string, indicating no auto-generation.

```
ksctl interfaces modify --name nae --auto-gen-ca-id ""
```

6. Create the external certificate chain, and combine them into a single PEM or PKCS12 file. The file should include the following certificates and key in the following order:
 - a. The server certificate.
 - b. Any intermediate certificate authorities that sign the server certificate. Start with the intermediate CA that

issued the server certificate. Next, add the issuer of the intermediate CA, if any. Continue adding any intermediate CAs higher up the hierarchy, each time adding the issuer of the last certificate you uploaded. Alternatively, you can add the intermediate CA as external CA in the CipherTrust Manager. See [Adding an external CA](#).

- c. The external root CA that signs any intermediate CAs. If there are no intermediate CAs, add the root CA that issued the server certificate.
Alternatively, you can add the root CA as external CA in the CipherTrust Manager. See [Adding an external CA](#).
- d. The server certificate's private key.

Note

- In the PEM format, the server certificate's private key should be included first, and in the PKCS12 format, it should be included last.
- It is optional to include the private key in the PEM file if you have already generated a CSR. Creating a CSR allows you to generate the key pair on the CipherTrust Manager and prevents exposing the private key outside of the CipherTrust Manager.

7. Upload the certificate chain file to the desired interface (NAE interface is shown in the example)

```
ksctl interfaces certificate modify --name=nae --format=pem --file=certchainfile.pem --private-key-password password
```

The CipherTrust Manager validates the entire certificate chain from server certificate till its root CA. If you want to skip the validation process (*not recommended*), use the `skip_validation` flag.

```
ksctl interfaces certificate modify --name=nae --format=pem --file=certchainfile.pem --private-key-password password --skip_validation true
```

Managing the preboot interface server certificate

The preboot interface can have a dynamically generated self-signed certificate or an external certificate. REST API and ksctl communicate with the preboot interface while the **disk is encrypted** and locked in the preboot state. This interface is only present if disk encryption has been enabled.

If an external certificate is not uploaded, CipherTrust Manager dynamically generates a self-signed certificate for the preboot interface with the default server name `preboot.ciphertrustmanager.local`.

The preboot interface uses an external certificate if a user **uploads** one.

View the preboot interface server certificate

- Use `ksctl interfaces certificate get --name preboot` to return the preboot interface certificate in the CLI.
- In the web UI:
 - a. Navigate to **Admin Settings > Interfaces**.
 - b. Next to the preboot interface, click the corresponding overflow icon () and select **Certificate Options**.
 - c. Select **View Certificate** click **Ok**.

Upload an external server certificate

CipherTrust Manager supports PEM or base64-encoded PKCS#12 format certificates. The private key algorithm must be RSA-2048 or RSA-4096. The certificate file must include the private key. You can include a chain of certificates, in ascending order from server certificate to root certificate authority (CA).

We recommend that you convert a copy of this certificate to DER format, and then provide the DER encoded certificate file to anyone unlocking the disk, to compare the fingerprint with the presented preboot interface certificate.

- Use `ksctl interfaces certificate modify --name preboot --file <certificate_file> --format <pem_or_pkcs12>` in the CLI.
- In the web UI:
 - a. Navigate to **Admin Settings > Interfaces**.
 - b. Next to the preboot interface, click the corresponding overflow icon () and select **Certificate Options**.
 - c. Select **Upload New Certificate** and click **Ok**.
 - d. Select a method to provide the external certificate, **File Upload**, **Text**, or **Certificate Chain**.

File Upload

Click **Choose Certificate File** and select the certificate file in your local file system.

This file must include the full certificate chain. The file format must be PEM or PKCS12. The file contents in order must be:

- the server certificate
- any intermediate CAs starting with the intermediate CA that issued the server certificate,
- the root CA

For PEM format, the server certificate's private key is included first, and for PKCS12 the server certificate's private key is included last.

Text

Paste the full text of the certificate into the text box.

This text must include the full certificate chain. The format must be PEM or PKCS12. The text contents in order must be:

- the server certificate
- any intermediate CAs starting with the intermediate CA that issued the server certificate,
- the root CA

For PEM format, the server certificate's private key is included first, and for PKCS12 the server certificate's private key is included last.

Certificate Chain

- a. Click **Build Certificate Chain**.

The **Add Certificate Chain** dialog displays.

- b. Provide the **Server Certificate** as a **File Upload** or **Text**.
- c. Click **Next**.
- d. Add any **Intermediate CAs** which sign the server certificate. Skip this step if there are no intermediate CAs.

You can provide an intermediate CA as a **File Upload** or **Text**. Click **Add Another** to add each additional intermediate CA.

- e. Click **Next**.
- f. Provide the **Root CA** as a **File Upload** or **Text**.
- g. Click **Next**.
- h. Provide the server certificate's **Private Key** as a **File Upload** or **Text**.
- i. Click **Add Certificate Chain**.
- e. Select the **Format** as **PEM** or **PKCS12**.
- f. (Optional) Provide a **Password** for the private key.
- g. Click **Upload Certificate**.

Download the preboot interface server certificate in the Web UI

1. Navigate to **Admin Settings > Interfaces**.
2. Next to the preboot interface, click the corresponding overflow icon () and select **Certificate Options**.
3. Select **Download Certificate** click **Ok**.
4. Click **Download Certificate** to download the certificate to a file or **Copy to Clipboard** to copy the certificate contents.
5. Click **Close**.

Interface modes (NAE, KMIP, and WEB)

The "mode interface" configuration parameter specifies the Interface mode, and must be one of the following:

Interface mode	Description
no-tls-pw-opt	This is a TCP mode to communicate over the interface. It is not mandatory to provide password in the authentication request to log on to the interface. If no password is provided, the login will be treated as anonymous.
no-tls-pw-req	This is a TCP mode to communicate over the interface and requires a password in the authentication request to log on to the interface.
unauth-tls-pw-opt	This is a TLS mode to communicate over the interface. A valid certificate signed by a trusted CA can establish a TLS connection. It is not mandatory to provide a password in the authentication request to log on to the interface. If no password is provided, the login will be treated as anonymous.
unauth-tls-pw-req	This is a TLS mode to communicate over the interface. A valid certificate signed by a trusted CA can establish a TLS connection. It is mandatory to provide password in the authentication request to log on to interface.
tls-pw-opt	This is a TLS mode to communicate over the interface, and authentication with a client certificate is required. It is not mandatory to provide password in the authentication request to log on to the interface. If no password is provided, the login will be treated as anonymous.
tls-pw-req	This is a TLS mode to communicate over the interface, and authentication with a client certificate is required. It is mandatory to provide password in the authentication request to log on to interface.
tls-cert-pw-opt	This is a TLS mode to communicate over the interface, and authentication with a client certificate is required. The username is extracted from the Username Location in the Certificate field, for example, the

Interface mode	Description
	certificate's Common Name (CN). It is not mandatory to provide password in the authentication request to log on to the interface.
tls-cert-and-pw	This is a TLS mode to communicate over the interface, and authentication with a client certificate is required. It is mandatory to provide a password in the authentication request to log on to the interface. Additionally, the username in the certificate must match the username provided in the authentication request.

The interface mode can be changed using the API or the CLI "ksctl interfaces modify" command.

Interface modes for each interface type

- **nae:** All modes are allowed. Default is `unauth-tls-pw-opt`.

`tls-pw-req` and `tls-cert-and-pw` are recommended for [user delegation flows](#).

`tls-cert-pw-opt` is recommended for [user impersonation flows](#).

Warning

The client validation will be mandatory for all NAE clients in the future release leading to deprecation of the following interface modes:

- TLS, ignore client cert, allow anonymous logins
- TLS, ignore client cert, user must supply password

Therefore, it is recommended *not* to use the above mentioned modes.

- **web:** The only allowed mode is `tls-cert-opt-pw-opt`.
- **kmip:** Only following modes are allowed:
 - `tls-cert-pw-opt` - This setting is recommended for [user impersonation flows](#)
 - `tls-cert-and-pw` - recommended for [user delegation flows](#)
 - `tls-pw-opt`

Username location in certificate (NAE, KMIP, and WEB)

The `cert_user_field` parameter specifies how the user name is extracted from the client certificate. The

default value is `CN`. This parameter must be one of the following:

- `CN`: Common Name
- `OU`: Organizational Unit
- `SN`: Surname
- `E`: Email address
- `E_ND`: Email without domain; the data to the left of the `@` sign in the email address is taken as the user name
- `UID`: User ID

For KMIP, the `cert_user_field` parameter options are only available when **Auto Registration** is enabled on the interface.

Note

For WEB, only CN is supported.

Certificate signing requests (CSR for NAE, KMIP, and WEB)

CSR provides you the ability to not expose private key for certificates created on the CipherTrust Manager for the Interfaces.

Adding a CSR

To add a CSR, run:

Syntax

```
ksctl interfaces csr add --name <interface-name> --common-name <common-name> --dn
s-name <dns-names-comma-separated> --email-address <email-addresses-comma-separat
ed> --ip-address <ip-addresses-comma-separated> --csr-name <names-comma-separate
d> --csr-out-file <csr-file>
```

Example Request

```
ksctl interfaces csr add --name kmip --common-name admin
```

Example Response

```
{
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIIEVTCCAj0CAQAwEDEOMAwGA1UEAxMF
```

```
YWRtaW4wggiMA0GCSqGSIb3DQEBAQUA\ nA4ICDwAwggIKAoICAQDMo2aufRdlvB3F1tiGnQdRokmsXGc
4nTMYVvkYrNn1zqTqK\ nJxLy1KY0/I07V+yWsZnbkXoieggBEo48FjqD2HDSr3K4BCSJOLkfw6velFG
U+7+8\ n6ei8eXZNJcBND6/RGRpe2Dz3xr17I4dbW4uoJhoPZei3It2yPhWlvIDc7sah591\ nXTzKzuaA
H24d6sJSD1OU0PN5SvGNw/v1D7VJoqsRrcytnhgCVav160ZWH24Pu4ty\ nhBK2wi+ElhRlQSScA7h8s2B
TFD/EG+TR2IinjAqNgCo4UspVJnADOGYkMJnSQLz1\ nJFf1VIMEAJQ2lnRYzpDodbIt/6jSZgqlXUmt1P
ZE5LOxWCQ0yXNEUJtP/+K6CS1\ nspVHGVwhWpD81tTwSYHq3Sx/aVvA/LSkfuOKgDFyclmxSM4o6qmKM
YfanYyqaD9k\ nTI/Xuf2aWrDWB3KqeKuVYj99tcTdcYbeYl4BeseAA8RJj5D0jLhnEAkgQWZN50\ nul
T+uvBhCgVJaXPOJfIeGDnJkCeeAxloa9RXztZItncUAvEWvS9Lvfl0YXnH9axf\ n7ofqHhdDocMX6w0yE4
2dGaqQQFWZCKsb+vUb2DaJrZCLQqoMitmHdMY+eOQacnXhN\ niGzN43oEspGBaIxKFRFm/ZTvZc6hHCfh
e6Uk6xqGUPgsIyzofoVsDarOq563MwID\ nAQABoAAwDQYJKoZIhvcNAQENBQADggIBAGWhiUGy1TywCnS
cen8P1AkS+qUVKksR\ niWP313J+3YW26aNFcTdsHdEzRCD/p/um8K6q5gEstdQRwMb0KaMDYJTRR+1Wf
i9\ nT3xio6rVQbkBSgl6amCF36yuCX2QiHI2PfpBhTnV5Q16YotjqLg+M2rxYMIKUpuB\ nCsAmBbV2lRk
14M2vdP6GIjUuY/mAIRCh23cPeKaMi6vQthoMDRvBnybuZoh8YbEn\ nbQ7tJbj0R/33zHRBUAUvNuN4qF
SdOTB0icOnomFvOHGTelebDoR+jABCS+ndePn\ nIqf9hDi59NJQeR5huvbQa5gJMulKMz1Hx4gQ2zTvH
xS4K8VYFE4SMfpd8gNNKbJF\ nVh2j+Mv3m5HPvwiGawEZAxZyRipyQ4GHc9pTCDDpfdyAfqOd3ogwEEln
f2DfBb5x\ n7mfiruOrcqQ9Kz6xYx2Xb0B3pSxTo5UamtL+ct5cabd+xQrhM7y2m5n8vvOkSne3\ n+JH9w
h7oMx3qRGzxS/aO9Fm6lGkE24f4QyO5KUHniD6Cl0E+mloMSIwb0hpX+zzf\ nXs5v9JUAyaoiEeNsFs0
1MDeIXuYyEY7x52qBNj8XsKbDeEnS9Fzf+qIX5Z2LldB\ ng0M6p/vXTZ+7YCFqphVnnx10E1BXnWBKDCO
CDFXDxZ9clV+uxkVVJHgzF1lrNvyI\ n1ATjjOZSmouN\ n-----END CERTIFICATE REQUEST-----\ n"
}
```

Note

If CSR parameters are not provided, it will use the default values.

Generating CSR from interface

UI

To generate a CSR from the interface:

1. Log on to the CipherTrust Manager GUI.
2. Navigate to **Admin Settings > Interfaces**.
3. Navigate to the **web** interface and click the ellipsis () icon.
4. Click **Generate CSR** and enter the required details.
5. Click **Generate CSR**.

Make sure to save the generated CSR.

Similarly, you can generate CSR from the other interfaces.

CLI

To generate a CSR from the Interface:

Syntax

```
ksctl interfaces csr get --name <interface-name> --csr-infile <csr-file>
```

Example Request

```
ksctl interfaces csr get --name kmip
```

Example Response

```
"csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIIEVTCCAj0CAQAwEDEOMAwGA1UEAxMFYWRTaW4wggiMA0GCSqGSIb3DQEBAQUA\nInA4ICDwAwggIKAoICAQDMo2aufRdlvB3F1tiGnQdRokmsXGc4nTMYVkyrNn1zqTqK\nInJxLy1KY0/I07V+yWsZnbkXo...AyaoiEeNsFsf01MDeIXuYYeY7x52qBNj8XsKbDeE\nnS9Fzf+qIX5Z2Lldb\ng0M6p/vXTZ+7YCFqphVnnx10E1BXnWBKDcOCDFXDxZ9clV+uxkVVJHgZF11rNv\nyI\nIn1ATjjOZSmouN\n-----END CERTIFICATE REQUEST-----\n"
```

Note

If you use the method described above to generate the CSR, the private key is already stored on the CipherTrust Manager.

To upload the generated CSR, refer to [Upload renewal server certificate](#) section.

Get Certificate for NAE, KMIP, and WEB Interfaces

To get the certificate for an Interface, run:

Syntax

```
ksctl interfaces certificate get --name <interface/port> --identifiertype <identifier type> --icertfile <cert-file>
```

- name - specifies name or port number of the interface. The interface name can be "nae", "web", "kmip", or "ssh".
- identifiertype - type of the interface identifier. It can be "name" or "port". If not specified, the default interface identifier type will be "name".
- icertfile - pipes the certificate to the designated file.

****Example request 1 ****

```
ksctl interfaces certificate get --name nae
```

Example response 1

```
{
```

```

    "certificates": "-----BEGIN CERTIFICATE-----\nMIICajCCAg+gAwIBAgIQX8CTju31Jry
7vZyOCoiq+TAKBggqhkJOPQODA jCBijEL\nMAkGA1UEBhMCVVMxCzAJBgNVBAGTA1RYMQ8wDQYDVQQHEw
ZBdXN0aW4xNjA0BgNV\nBAkTLTk0NDIgaQ2FwaXRhbCBvZiBUZXhhcyBlaWdod2F5IE5vcnRoIFN1aXRlI
DQw\nnMDEPMA0GA1UEChMGVGhhbGVzMRQwEgYDVQQDEwtaXBoZXJUCnVzdAeFw0yMzA0\nnMTMwNzEwNT
RaFw0yNDA0MTMwNzEwNTRaMIGKMQswCQYDVQQGEwJVUzELMAkGA1UE\nnCBMCMVGFgxDzANBgNVBACTBkF1c
3RpbjE2MDQGA1UECRMtOTQ0MiBDYXBpdGFsIG9m\nnIFRleGFzIEhpZ2h3YXkgTm9ydGggU3VpdGUgNDaw
MQ8wDQYDVQQKEwZUaGFsZXN0\nnFDASBgNVBAMTC0NpcGhlclRydXN0MFkwEwYHkoZIZj0CAQYIKoZIZj0
DAQcDQgAE\nnxP1SETV3zzuYodXPWGaIT46oJStDQj1dnp8QHTVSZjY6+dNZ/U6ZiOpR+yPlNXFs\nnQmDf
n/EtC0ViaAqbszQAf6NVMFMwDgYDVR0PAQH/BAQDAgKkMBMGA1UdJQOMMAoG\nnCCsGAQUFBwMBMCwGA1U
dEQQlMCOBIXRlY2huaWNhbC5zdXBwb3J0QHROYWxlcl2dy\nnb3VwLmNvbTAKBggqhkJOPQODA gNJADBGAi
EAmCR6dDL1PoAykXqmSCiBypFFq276\nnFKcKRh1QLTrX8ncCIQCblJYOuV+PqLqg+7gwZ5+R6Y1LFsA4P
qyPdnTLC6xz6g==\n-----END CERTIFICATE-----\n"
}

```

It returns the public portion of the certificate in the PEM format.

Example request 2 (identifiertype is set as port)

```
ksctl interfaces renewal-cert get --name 9000 --identifiertype port
```

Example response 2

```

{
  "certificates": "-----BEGIN CERTIFICATE-----\nMIICajCCAg+gAwIBAgIQX8CTju31Jry
7vZyOCoiq+TAKBggqhkJOPQODA jCBijEL\nMAkGA1UEBhMCVVMxCzAJBgNVBAGTA1RYMQ8wDQYDVQQHEw
ZBdXN0aW4xNjA0BgNV\nBAkTLTk0NDIgaQ2FwaXRhbCBvZiBUZXhhcyBlaWdod2F5IE5vcnRoIFN1aXRlI
DQw\nnMDEPMA0GA1UEChMGVGhhbGVzMRQwEgYDVQQDEwtaXBoZXJUCnVzdAeFw0yMzA0\nnMTMwNzEwNT
RaFw0yNDA0MTMwNzEwNTRaMIGKMQswCQYDVQQGEwJVUzELMAkGA1UE\nnCBMCMVGFgxDzANBgNVBACTBkF1c
3RpbjE2MDQGA1UECRMtOTQ0MiBDYXBpdGFsIG9m\nnIFRleGFzIEhpZ2h3YXkgTm9ydGggU3VpdGUgNDaw
MQ8wDQYDVQQKEwZUaGFsZXN0\nnFDASBgNVBAMTC0NpcGhlclRydXN0MFkwEwYHkoZIZj0CAQYIKoZIZj0
DAQcDQgAE\nnxP1SETV3zzuYodXPWGaIT46oJStDQj1dnp8QHTVSZjY6+dNZ/U6ZiOpR+yPlNXFs\nnQmDfn/EtC0ViaAqbszQAf6NVMFMwDgYDVR0PAQH/BAQDAgKkMBMGA1UdJQOMMAoG\nnCCsGAQUFBwMBMCwGA1U
dEQQlMCOBIXRlY2huaWNhbC5zdXBwb3J0QHROYWxlcl2dy\nnb3VwLmNvbTAKBggqhkJOPQODA gNJAD
BGAiEAmCR6dDL1PoAykXqmSCiBypFFq276\nnFKcKRh1QLTrX8ncCIQCblJYOuV+PqLqg+7gwZ5+R6Y1LFsA4P
qyPdnTLC6xz6g==\n-----END CERTIFICATE-----\n"
}

```

It returns the public portion of the certificate in the PEM format.

Example request 3 (identifiertype is set as name)

```
ksctl interfaces renewal-cert get --name nae --identifiertype name
```

Example response 3

```

{
  "certificates": "-----BEGIN CERTIFICATE-----\nMIICajCCAg+gAwIBAgIQX8CTju31Jry
7vZyOCoiq+TAKBggqhkJOPQODA jCBijEL\nMAkGA1UEBhMCVVMxCzAJBgNVBAGTA1RYMQ8wDQYDVQQHEw
ZBdXN0aW4xNjA0BgNV\nBAkTLTk0NDIgaQ2FwaXRhbCBvZiBUZXhhcyBlaWdod2F5IE5vcnRoIFN1aXRlI
DQw\nnMDEPMA0GA1UEChMGVGhhbGVzMRQwEgYDVQQDEwtaXBoZXJUCnVzdAeFw0yMzA0\nnMTMwNzEwNT
RaFw0yNDA0MTMwNzEwNTRaMIGKMQswCQYDVQQGEwJVUzELMAkGA1UE\nnCBMCMVGFgxDzANBgNVBACTBkF1c
3RpbjE2MDQGA1UECRMtOTQ0MiBDYXBpdGFsIG9m\nnIFRleGFzIEhpZ2h3YXkgTm9ydGggU3VpdGUgNDaw
MQ8wDQYDVQQKEwZUaGFsZXN0\nnFDASBgNVBAMTC0NpcGhlclRydXN0MFkwEwYHkoZIZj0CAQYIKoZIZj0
DAQcDQgAE\nnxP1SETV3zzuYodXPWGaIT46oJStDQj1dnp8QHTVSZjY6+dNZ/U6ZiOpR+yPlNXFs\nnQmDfn/EtC0ViaAqbszQAf6NVMFMwDgYDVR0PAQH/BAQDAgKkMBMGA1UdJQOMMAoG\nnCCsGAQUFBwMBMCwGA1U
dEQQlMCOBIXRlY2huaWNhbC5zdXBwb3J0QHROYWxlcl2dy\nnb3VwLmNvbTAKBggqhkJOPQODA gNJAD
BGAiEAmCR6dDL1PoAykXqmSCiBypFFq276\nnFKcKRh1QLTrX8ncCIQCblJYOuV+PqLqg+7gwZ5+R6Y1LFsA4P
qyPdnTLC6xz6g==\n-----END CERTIFICATE-----\n"
}

```

```
Izj0DAQcDQgAE\nxP1SETV3zzuYodXPWGaIT46oJStDQj1dnp8QHTVSZjY6+dNZ/U6ZiOpR+yPlNXFs\n
QmDfn/EtC0ViaAqbszQAF6NVMFMwDgYDVR0PAQH/BAQDAgKkMBMGA1UdJQQMMAoG\nCCsGAQUFBwMBMCw
GA1UdEQQ1MCOBIXRlY2huaWNhbC5zdXBwb3J0QHRoYWxlY2dy\nnb3VwLmNvbTAKBgggqhkJOPQQDAgNJAD
BGAiEAmCR6dDLlPoAykXqmSClbyFFq276\nFKcKRh1QLTrX8ncCIQCblJY0uV+PqLqg+7gwZ5+R6Y1LF
sA4PqyPdnTLC6xz6g==\n-----END CERTIFICATE-----\n"
}
```

It returns the public portion of the certificate in the PEM format.

Auto-generation of server certificates (NAE, KMIP, and WEB)

By default, the CipherTrust Manager is configured to auto-generate a server certificate for each interface using the initial Local CA. The current certificate becomes the certificate issued by the Local CA. If the Local CA that is used to auto-generate is changed/updated, [manually renew and apply the new certificate](#).

Note

Local CAs for interfaces are often updated when a backup is restored, necessitating manual renewal.

In case of cluster join, the existing Local CA of the new node is overwritten by the Local CAs from the cluster. Therefore, the current certificate of the new node issued by its Local CA becomes unknown to the cluster. As a result, auto-generation happens on the first restart of the new node after it joins the cluster.

Manually regenerating server certificates

To regenerate the server certificate manually for an interface, perform the following steps:

1. Set the CA ID on the interface for auto generation.

Syntax

```
ksctl interfaces modify --name <interface-name> --auto-gen-ca-id <ca-id>
```

Example

```
ksctl interfaces modify --name nae_all_9090 --auto-gen-ca-id kylo:kylo:naboo:l
ocalca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a
```

2. Generate the server certificate.

Syntax

```
ksctl interfaces auto-gen-server-cert --name <interface-name> --pending-renewa
l <true/false>
```

Example

```
ksctl interfaces auto-gen-server-cert --name nae_all_9090 --pending-renewal false
```

The above command updates the interface with the newly generated server certificate using the CA present in the `auto-gen-ca-id` field.

If `pending-renewal` is set to `true`, the certificate will be saved as an [upcoming/renewed certificate](#) to be applied later.

If `pending-renewal` is set to `false` (default behavior), the auto-generated server certificate will immediately apply on the server certificate resource and replace the currently active server certificate.

Changing the local CA used for auto-generation

If needed, the Local CA used for auto-generation of the server certificate can be changed. Run the following command:

Example

```
ksctl interfaces modify --name web --auto-gen-ca-id kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a
```

Response

```
{
  "id": "3b85c5cf-6075-4555-a17c-2a12479081d4",
  "name": "web",
  "mode": "unauth-tls-opt-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a"
    ],
    "external": []
  },
  "createdAt": "2018-07-06T16:10:23.705291Z",
  "updatedAt": "2018-07-06T17:15:38.285859Z"
}
```

Disabling auto-generation of server certificate

To disable the auto-generation feature, run the following command:

Syntax

```
ksctl interfaces modify --name web --auto-gen-ca-id ""
```

Response

```
{
  "id": "3b85c5cf-6075-4555-a17c-2a12479081d4",
  "name": "web",
  "mode": "tls-cert-opt-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a"
    ],
    "external": []
  },
  "createdAt": "2018-07-06T16:10:23.705291Z",
  "updatedAt": "2018-07-06T17:16:41.119686Z"
}
```

Note

The auto-generation feature is automatically disabled if an external CA chain and server certificate are uploaded to the interface.

Customizing auto-generation of interface certificate

To supply custom fields for auto-generation of an interface certificate, specify `--local-auto-gen-attributes-file` in the `ksctl interfaces modify` command. For example, run:

```
ksctl interfaces modify --local-auto-gen-attributes-file <local-auto-gen-attributes-file-name>.json
```

The `--local-auto-gen-attributes-file` option specifies a file that contains attributes for auto-generation of a given interface. This provides an option to supply custom fields for automatic interface certification generation.

These parameters are for the local node only and have no effect on other nodes in a cluster. Without customization, the system default values are used, or taken from cloud-init, if applicable.

Content of a sample local auto-generation attributes file is:

```
{
  "cn": "KeySecure Server on ks.eng.thesgroup.com",
  "dns_names": [
    "ks.eng.thesgroup.com",
  ]
}
```



```

    "ks-ha.eng.thalesgroup.com"
  ],
  "email_addresses": [
    "admin1@eng.thalesgroup.com",
    "admin2@eng.thalesgroup.com"
  ],
  "ip_addresses": [
    "10.3.111.11",
    "10.3.111.22"
  ],
  "names": [
    {
      "C": "US",
      "ST": "CA",
      "L": "San Jose",
      "O": "Thales CPL",
      "OU": "Engineering"
    }
  ]
}

```

If fields for the auto-generation of an interface certificate are changed, on restart a new server certificate gets generated based on the new attributes and signed by the CA specified in the `auto-gen-ca-id` field.

Managing renewal server certificate

When a server certificate, or one of its signing CAs, is about to expire, we need to advertise to the service consumers (the clients) that a new replacement certificate chain will be coming soon. This allows the clients to prepare upfront to accept both old and new server CA chains in order to avoid any downtime when the server certificate actually switches to the new one.

Therefore, a mechanism is available to have an upcoming renewal server certificate for an interface. Each interface can have one upcoming server certificate chain. Clients can check for a renewal certificate, to have the chance to timely download and configure their local trust stores. As well, when the active server certificate expires for an interface, CipherTrust Manager automatically applies the upcoming renewal certificate.

The following operations can be performed:

- Put or upload renewal certificate
- Get renewal certificate
- Delete renewal certificate
- Immediately apply renewal certificate

An upcoming renewal certificate must be uploaded to an interface for the get, delete, and apply operations to be available.

Managing Renewal Server Certificate through Web Console UI

Upload renewal server certificate

1. Login to the root domain as an Application Administrator in the Admins group, such as `admin`.

2. Navigate to **Admin Settings > Interfaces**.
3. Find relevant interface in the table. NAE clients use the `nae` interface type, KMIP clients use the `kmip` interface type, and all other clients use the `web` interface type.
4. Click the ellipsis icon () corresponding to the interface you want to retrieve the upcoming certificate.
5. Click **Renewal Certificate Options....**
6. Click **Upload/Generate**.
7. Click **Ok**.
8. Select a method to provide the renewal certificate, **File Upload**, **Text**, or **Certificate Chain**. **File Upload** and **Text** are suitable for a server certificate signed by CAs already known to both the CipherTrust Manager and the clients. **Certificate Chain** is needed if one or more signing CA is not yet known by the CipherTrust Manager or the clients.

File Upload

Click **Choose Certificate File** and select the certificate file in your local file system.

This file must full include the full certificate chain. The file format must be PEM or PKCS12. The file contents in order must be:

- the server certificate
- any intermediate CAs starting with the intermediate CA that issued the server certificate,
- the root CA
- the server certificate's private key

the server certificate's private key *is not required* if the CSR is generated directly through the [Interface](#), as it is already stored on the CipherTrust Manager.

Note

- In the PEM format, the server certificate's private key should be included first, and in the PKCS12 format, it should be included last.
- It is optional to include the private key in the PEM file if you have already generated a CSR. Creating a CSR allows you to generate the key pair on the CipherTrust Manager and prevents exposing the private key outside of the CipherTrust Manager.

Text

Paste the full text of the certificate into the text box.

This text must full include the full certificate chain. The format must be PEM or PKCS12. The text contents in order must be:

- the server certificate
- any intermediate CAs starting with the intermediate CA that issued the server certificate,
- the root CA
- the server certificate's private key

the server certificate's private key *is not required* if the CSR is generated directly through the [Interface](#), as it is already stored on the CipherTrust Manager.

Note

- In the PEM format, the server certificate's private key should be included first, and in the PKCS12 format, it should be included last.
- It is optional to include the private key in the PEM file if you have already generated a CSR. Creating a CSR allows you to generate the key pair on the CipherTrust Manager and prevents exposing the private key outside of the CipherTrust Manager.

Certificate Chain

- a. Click **Build Certificate Chain**.

The **Add Certificate Chain** dialog displays.

- b. Provide the **Server Certificate** as a **File Upload** or **Text**.
- c. Click **Next**.
- d. Add any **Intermediate CAs** which sign the server certificate. Skip this step if there are no intermediate CAs.

You can provide an intermediate CA as a **File Upload** or **Text**. Click **Add Another** to add each additional intermediate CA.

- e. Click **Next**.
- f. Provide the **Root CA** as a **File Upload** or **Text**.

-
- g. Click **Next**.
 - h. Provide the server certificate's **Private Key** as a **File Upload** or **Text**.

the server certificate's private key *is not required* if the **CSR** is generated directly through the **Interface**, as it is already stored on the CipherTrust Manager.

- i. Click **Add Certificate Chain**.
9. Confirm that the **Format** dropdown correctly specifies the server certificate as **PEM** or **PKCS12**.
10. Click **Upload Certificate**.

Get renewal server certificate

1. Login to the root domain as an Application Administrator in the Admins group, such as `admin`.
2. Navigate to **Admin Settings > Interfaces**.
3. Find relevant interface in the table. NAE clients use the `nae` interface type, KMIP clients use the `kmip` interface type, and all other clients use the `web` interface type.
4. Click the ellipsis icon (`...`) corresponding to the interface you want to retrieve the upcoming certificate.
5. Click **Renewal Certificate Options....**
6. Click **Download**.
7. Click **Ok**.

A dialog pops up displaying the certificate text.

8. Click **Download Certificate** to download the certificate as a file or **Copy to Clipboard** to copy the certificate text.
9. After you have saved the certificate, click **Close**.

Delete renewal server certificate

1. Login to the root domain as an Application Administrator in the Admins group, such as `admin`.
 2. Navigate to **Admin Settings > Interfaces**.
 3. Find relevant interface in the table. NAE clients use the `nae` interface type, KMIP clients use the `kmip` interface type, and all other clients use the `web` interface type.
 4. Click the ellipsis icon (`...`) corresponding to the interface you want to retrieve the upcoming certificate.
 5. Click **Renewal Certificate Options....**
-

6. Click **Delete**.

7. Click **Ok**.

You are prompted with a pop-up to confirm deletion.

8. Click **Delete** to confirm.

Apply renewal server certificate

1. Login to the root domain as an Application Administrator in the Admins group, such as `admin`.

2. Navigate to **Admin Settings > Interfaces**.

3. Find relevant interface in the table. NAE clients use the `nae` interface type, KMIP clients use the `kmip` interface type, and all other clients use the `web` interface type.

4. Click the ellipsis icon (`...`) corresponding to the interface you want to retrieve the upcoming certificate.

5. Click **Renewal Certificate Options...**

6. Click **Apply**.

7. Click **Ok**.

You are prompted with a dialog to confirm that you want to apply the upcoming renewal certificate.

8. Click **Apply**.

View renewal server certificate properties

1. Login to the root domain as an Application Administrator in the Admins group, such as `admin`.

2. Navigate to **Admin Settings > Interfaces**.

3. Find relevant interface in the table. NAE clients use the `nae` interface type, KMIP clients use the `kmip` interface type, and all other clients use the `web` interface type.

4. Click the ellipsis icon (`...`) corresponding to the interface you want to retrieve the upcoming certificate.

5. Click **Renewal Certificate Options...**

6. Click **View**.

7. Click **Ok**.

A pop-up dialog displays the following properties:

- issuer

- subject
- serialNumber
- notAfter
- notBefore
- subjectAlternativeNames

8. After you have read the properties, click **Close**.

Managing Renewal Server Certificate through ksctl CLI

All the responses in the sub-sequent sections will contain the certificate chain including the CA.

Upload renewal certificate

To upload a new renewal certificate server and its CA chain:

Example request

```
ksctl interfaces renewal-cert upload --name nae --file ./serverCertKey.pem --form
at pem
```

Make sure not to activate it yet for the interface.

The certificate and its associated key data must be in PEM format or base64 encoded PKCS12 format. The certificate can also include a chain of certificates. For the certificate chain, maintain this order - <server CA> <root CA>.

Example response

```
{
  "certificates": "-----BEGIN CERTIFICATE-----\nMIIEEgqjCCApKgAwIBAgIQZkQE+oYbUzq
L+bmYF2f3VjANBgkqhkiG9w0BAQsFADBa\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCMVFGxDbzANBgNVBA
cTBkF1c3RpbjEPMA0G\nA1UEChMGVGVhbnGVzMRwwGgYDVQQDEwNDaXB0ZXJUCnVzdCBSb290IENBMB4XD
TIz\nnMDMyMjA2MTEwOV0XDTMzMzMDA5MzUzMDUwDjEMMAoGA1UEAxMDbmF1bG1lIjAN\nnBgkqhkiG9w
0BAQEFAAOCAQ8AMIIBCgKCAQEAuB2UJMu2lMYSx181SaLKXxuOrz39\nnDFd6yae+fzFYXzyZxudRDeHMS
t0knPoV2L3WsfP8PibtAKHyUK6b8Ego9CcAh16X\nnYeOmeehBcYk/IwbUcp0LAW4/4lB/a++Gkd9HPmUm
OezbVMOyjk4LoerSLiqfbXIU\nnHpbrFcXuTNKjvelymMMT2LubguXWu6xPB6faZtspn3ZB22FgqT4KPUL
UrDuiTTJ3\nnwXJyKCIEb5kpEI56u6aE2N+IGQ2gFWN9tHWr6SwjQNm7QhdV63y5+JLMXyHuINf\nnLr24
fHEBUoSptbu4yhTu5t7h1O5TTKIV/PBjjvYbYtt0uYIcQhw3U5qhQWQIDAQAB\nno4G3MIG0MA4GA1UdDwE
B/wQEAWIdiDATBgNVHSUEDDAKBggrBgEFBQcC\nDATAMBgNV\nnHRMBAf8EAjAAMB8GA1UdIwQYMBaAFDEexv
fPs6aqKYwDCC6r7dQUEhKMF4GA1Ud\nnHwRXMFUwU6BR0E+GTWh0dHA6Ly9jaXB0ZXJ0cnVzdG1hbmFnZ
XIubG9jYjYwWwY3J3\nncy9iMTk2M2U4NS0wMmUuXLTQ2MmQtYTQzNi1hNWUuXmWNlY2NmMDEuY3J3sMA0GCSq
G\nnS1b3DQEBChwAA4ICAQA+4SrI4C4214C8Aw5CA1bPdLFPoMoABwaCg2p0DqR0Su7E\nnXZSZexDxod8
w/5aNAJA2rrHCJqwaDoKehNfaa/fQ/wDhVPkw6Nn2gjn6IEVNpxgU\nnBQbE+i0vmU0NnsHF6VEMh3299s
f94K9IxHqZbDgogBeBLxtqjvLSSECeOmmfL2V\nn/GfzMXKsiHp97gNwK8Mt6vWA4uJ/ojhK5mU1fRoEp
4MyH6IuauLoWMNDGQ9xEmj9\nnRYKzZgZsEraS3sOLj3ZWDkpF/tGMAiadqRav9Zyb7/mMJ014RZdpRKMT
LFQ/xqnM\nnZPsKYj0g+Q9BfK2vRKmTX2phTnjqeoVTy3qB6XxIHItmViURCT0t40tva7Ici89U\nnxYans
hAtFyCzq+1aMKbt+SeDD6jo2Q4LBNnMTK8liZqMhNUs3A0rz1mTlpUbiLBi\nnd4Nz8d55iJxwOMSkbdQv
WpoXHJnDEQW9d/9KFgnJu4nR8192Yta5BuxvuZHY57HL\nnriA3FJ+VFzZeZpEJOdUIpaAfGcsMBWd+ir
```

```
H+uPA70bcrkIlF+9YCr81LOcv7gY7l\n8cIMrAikMjY3YMOgXmqxzHq1WeooHKFrtrjYJDL9gVeKn2EmM
WCSYY1lKXzZ1pgG\nmdEOyrSYjCYgMyj5vp/Irfl57KwVVUo5DHZVx9fBLWAosKdeXNgYrXwhEUzx9
A==\n-----END CERTIFICATE-----\n"
}
```

Get renewal certificate

To fetch the renewal server certificate:

Syntax

```
ksctl interfaces renewal-cert get --name <Interface-name> --identifiertype <type
of interface identifier>
```

- **name** - specifies name or port number of the interface. The interface name can be "nae", "web", "kmip", or "ssh".
- **identifiertype** - type of the interface identifier. It can be "name" or "port". If not specified, the default interface identifier type will be "name".

Example request 1

```
ksctl interfaces renewal-cert get --name nae
```

Example response 1

```
{
  "certificates": "-----BEGIN CERTIFICATE-----\nMIICajCCAg+gAwIBAgIQX8CTju31Jry
7vZyOCoiq+TAKBggqhkJOPQODAajCBijEL\nMAkGA1UEBhMCVVMxCzAJBgNVBAGTA1RYMQ8wDQYDVQQHEw
ZBdXN0aW4xNjA0BgNV\nBAkTLTk0NDIgaQ2FwaXRhbCBvZiBUZXhhcyBlaWdod2F5IE5vcnRoIFN1aXRlI
DQw\nMDkPMA0GA1UEChMGVGVhbnBvMRQwEgYDVQQDEwtaXBoZXJUCnVzdDAeFw0yMzA0\nMTMwNzEwNT
RaFw0yNDA0MTMwNzEwNTRaMIGKMQswCQYDVQQGEwJVUzELMAkGA1UE\nnCBMCMVFgxdzANBgNVBACTBkF1c
3RpbjE2MDQGA1UECRMtOTQ0MiBDYXBpdGFsIG9m\nIFRleGFzIEhpZ2h3YXkgTm9ydGggU3VpdGUgNDaw
MQ8wDQYDVQQKEwZUaGFsZXN1c\nFDASBgNVBAMTC0NpcGhlclRydXN0MFkwEwYHKoZIzj0CAQYIKoZIzj0
DAQcDQgAE\nnxP1SETV3zzuYodXPWGaIT46oJStDQj1dnp8QHTVSZjY6+dNZ/U6ZiOpR+yPlNXFs\nnQmDf
n/EtC0ViaAqbszQAF6NVMFMwDgYDVR0PAQH/BAQDAgKkMBMGA1UdJQQMMAoG\nnCCsGAQUFBwMBMCwGA1U
dEQQlMCOBIXRlY2huaWNhbc5zdXBwb3J0QHRoYWxlc2dy\nnb3VwLmNvbTAKBggqhkJOPQODAgnJADBGai
EAmCR6dDLlPoAykXqmSCIbypFFq276\nnFKcKRhlQLTrX8ncCIQCblJYOuV+PqLqg+7gwZ5+R6Y1LFsA4P
qyPdnTLC6xz6g==\n-----END CERTIFICATE-----\n"
}
```

It returns the public portion (only public key and CA chain) of the upcoming renewal certificate in the PEM format.

Example request 2 (identifiertype is set as port)

```
ksctl interfaces renewal-cert get --name 9000 --identifiertype port
```

Example response 2

```
{
```

```

    "certificates": "-----BEGIN CERTIFICATE-----\nMIICajCCAg+gAwIBAgIQX8CTju31Jry
7vZyOCoiq+TAKBggqhkJOPQQDAjCBijEL\nMAkGA1UEBhMCVVMxCzAJBgNVBAGTA1RYMQ8wDQYDVQQHEw
ZBdXN0aW4xNjA0BgNV\nnBAkTLTk0NDIgQ2FwaXRhbCBvZiBUZXhhcyBlaWdod2F5IE5vcnRoIFN1aXRlI
DQw\nnMDEPMA0GA1UEChMGVGhhbGVzMtQ0MiBDYXBpdGFsIG9m\nnIFRleGFzIEhpZ2h3YXkgTm9ydGgffgfgU3VpdGUg
NDAwMQ8wDQYDVQQKEwZUaGFsZXMx\nnFDASBgNVBAMTC0NpcGhlclRydXN0MFkwEwYHKoZIzj0CAQYIKoZ
Izj0DAQcDQgAE\nnxP1SETV3zzuYodXPWGaIT46oJStDQj1dnp8QHTVSZjY6+dNZ/U6ZiOpR+yPlNXFs\nn
QmDfn/EtC0ViaAqbszQAF6NVMFMwDgYDVR0PAQH/BAQDAgKkMBMGAlUdJQQMMAoG\nnCCsGAQUFBwMBMCw
GA1UdEQQlMCOBIXRlY2huaWNhbC5zdXBwb3J0QHRoYWxlclcy\nnb3VwLmNvbTAKBggqhkJOPQQDAgNJAD
BGAiEAmCR6dDLlPoAykXqmSCibypFFq276\nnFKcKRhlQLTrX8ncCIQCblJYOuV+PqLqg+7gwZ5+R6Y1LF
sA4PqyPdnTLC6xz6g==\n-----END CERTIFICATE-----\n"
}

```

It returns the public portion (only public key and CA chain) of the upcoming renewal certificate in the PEM format.

Example request 3 (identifiertype is set as name)

```
ksctl interfaces renewal-cert get --name nae --identifiertype name
```

Example response 3

```

{
  "certificates": "-----BEGIN CERTIFICATE-----\nMIICajCCAg+gAwIBAgIQX8CTju31Jry
7vZyOCoiq+TAKBggqhkJOPQQDAjCBijEL\nMAkGA1UEBhMCVVMxCzAJBgNVBAGTA1RYMQ8wDQYDVQQHEw
ZBdXN0aW4xNjA0BgNV\nnBAkTLTk0NDIgQ2FwaXRhbCBvZiBUZXhhcyBlaWdod2F5IE5vcnRoIFN1aXRlI
DQw\nnMDEPMA0GA1UEChMGVGhhbGVzMtQ0MiBDYXBpdGFsIG9m\nnIFRleGFzIEhpZ2h3YXkgTm9ydGgffgfgU3VpdGUg
NDAwMQ8wDQYDVQQKEwZUaGFsZXMx\nnFDASBgNVBAMTC0NpcGhlclRydXN0MFkwEwYHKoZIzj0CAQYIKoZ
Izj0DAQcDQgAE\nnxP1SETV3zzuYodXPWGaIT46oJStDQj1dnp8QHTVSZjY6+dNZ/U6ZiOpR+yPlNXFs\nn
QmDfn/EtC0ViaAqbszQAF6NVMFMwDgYDVR0PAQH/BAQDAgKkMBMGAlUdJQQMMAoG\nnCCsGAQUFBwMBMCw
GA1UdEQQlMCOBIXRlY2huaWNhbC5zdXBwb3J0QHRoYWxlclcy\nnb3VwLmNvbTAKBggqhkJOPQQDAgNJAD
BGAiEAmCR6dDLlPoAykXqmSCibypFFq276\nnFKcKRhlQLTrX8ncCIQCblJYOuV+PqLqg+7gwZ5+R6Y1LF
sA4PqyPdnTLC6xz6g==\n-----END CERTIFICATE-----\n"
}

```

It returns the public portion (only public key and CA chain) of the upcoming renewal certificate in the PEM format.

Delete renewal certificate

To delete the previously uploaded upcoming renewal certificate for the specified interface:

Example request

```
ksctl interfaces renewal-cert delete --name nae
```

There will be no response if the certificate deleted successfully.

Apply renewal certificate

To immediately replace the existing or in-use certificate of the specified interface with the upcoming renewal

certificate:

Example request

```
ksctl interfaces renewal-cert apply -n nae
```

There will be no response if the renewal certificate applies successfully.

Trusted CAs (NAE, KMIP, and WEB)

Each interface can have a set of either local and/or external trusted CAs for client authentication. By default each interface trusts the initial Local CA for client authentication but it can be modified by updating the interface. Please note that the concept of trusted CAs is only relevant when client authentication is used by updating the interface mode.

If the client provides a complete chain of certificates in the TLS call, the CipherTrust Manager interface must be configured to trust the root CA of the issued certificate. If the request does not contain the CA chain, the server needs to trust the intermediate CAs that issued the client certificate. For example, if the client is registered in the sub-domain "dom1" and tries to connect to the interface using only the client certificate, the connection will not be successful. To establish a connection:

1. Export the issuer CA of the client certificate from the "dom1".
2. Add the issuer CA to an external CA in the root domain.
3. Add the external CA in the interface trusted CAs list explicitly to establish the trust.

Example - configuration update for web interface to trust the set of two local CAs

Example

```
ksctl interfaces modify --name web --trusted-local-cas kylo:kylo:naboo:localca:d21584bd-7d8e-44a0-9eac-693f2fa32504,kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a
```

Response

```
{
  "id": "3b85c5cf-6075-4555-a17c-2a12479081d4",
  "name": "web",
  "mode": "tls-cert-opt-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:d21584bd-7d8e-44a0-9eac-693f2fa32504",
      "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a"
    ],
    "external": []
  },
}
```

```

    "createdAt": "2018-07-06T16:10:23.705291Z",
    "updatedAt": "2018-07-06T17:35:29.053809Z"
  }

```

Example - configuration update for web interface to trust the set of two external CAs

Example

```

ksctl interfaces modify --name web --trusted-external-cas kylo:kylo:naboo:localca:d21584bd-7d8e-44a0-9eac-693f2fa32504,kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a

```

Response

```

{
  "id": "3b85c5cf-6075-4555-a17c-2a12479081d4",
  "name": "web",
  "mode": "tls-cert-opt-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a",
  "trusted_cas": {
    "local": [],
    "external": [
      "kylo:kylo:naboo:localca:d21584bd-7d8e-44a0-9eac-693f2fa32504",
      "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a"
    ]
  },
  "createdAt": "2018-07-06T16:10:23.705291Z",
  "updatedAt": "2018-07-06T17:35:29.053809Z"
}

```

Enforcing Client Validation on NAE Interface

The `allow_unregistered` flag is added to prohibit the unregistered clients to access the CipherTrust Manager. For new NAE interfaces, the default value of this flag is set to `false`, however, you can change the value of this flag to `true` (allowing unregistered clients to access the CipherTrust Manager).

Note

Although the `allow_unregistered` flag is configurable, but it is recommended not to change the value of the flag from `false` (default) to `true` for the new NAE interfaces, or else an alarm will be triggered. This flag is applicable only for NAE interfaces.

For old (legacy), already existing interfaces, the `allow_unregistered` flag is set to `true` for backward compatibility. For such interfaces, whenever a connection is established with the CipherTrust Manager, a record gets generated with the following message **"Unregistered client found, please register a client. System may go out of compliance."** and the connection continues to work.

When `allow_unregistered` flag is set to `false` and an unregistered client tries to connect, a record gets generated with the following message **"Client registration enforcement enabled. Unregistered client found, please register a new client manually or enable auto registration."**

Creating NAE Interface

To create an NAE interface, run:

Syntax

```
ksctl interfaces create --type <Interface-type> --port <Port> --allow-unregistered <true/false>
```

Request

```
ksctl interfaces create --type nae --port 9002 --allow-unregistered true
```

Response

```
{
  "id": "4540efac-8180-4265-bc81-75d3e1ce26d5",
  "name": "nae_all_9002",
  "mode": "unauth-tls-pw-req",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:1804704e-029c-4bd7-9d47-3c947c6d08fd",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:1804704e-029c-4bd7-9d47-3c947c6d08fd"
    ]
  },
  "createdAt": "2022-12-01T05:07:46.972399Z",
  "updatedAt": "2022-12-01T05:07:46.972399Z",
  "default_connection": "local_account",
  "port": 9002,
  "network_interface": "all",
  "interface_type": "nae",
  "minimum_tls_version": "tls_1_2",
  "maximum_tls_version": "tls_1_3",
  "local_auto_gen_attributes": {
    "cn": "nae_all_9002.ciphertrustmanager.local",
    "dns_names": [
      "nae_all_9002.ciphertrustmanager.local"
    ],
    "email_addresses": [
      "technical.support@thalesgroup.com"
    ],
    "names": [
      {
        "C": "US",
        "ST": "TX",
        "L": "Austin",
        "O": "Thales",
        "OU": ""
      }
    ]
  }
}
```

```

    },
    "generated": false
  },
  "enabled": true,
  "meta": {},
  "tls_ciphers": [
    {
      "cipher_suite": "TLS_AES_256_GCM_SHA384",
      "enabled": true,
      "configurable": false
    },
    {
      "cipher_suite": "TLS_CHACHA20_POLY1305_SHA256",
      "enabled": true,
      "configurable": false
    },
    {
      "cipher_suite": "TLS_AES_128_GCM_SHA256",
      "enabled": true,
      "configurable": false
    },
    {
      "cipher_suite": "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
      "enabled": true,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
      "enabled": true,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
      "enabled": true,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
      "enabled": true,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
      "enabled": false,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
      "enabled": false,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_RSA_WITH_AES_128_CBC_SHA256",
      "enabled": false,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA",

```

```

        "enabled": false,
        "configurable": false
    },
    {
        "cipher_suite": "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA",
        "enabled": false,
        "configurable": false
    },
    {
        "cipher_suite": "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA",
        "enabled": false,
        "configurable": false
    },
    {
        "cipher_suite": "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA",
        "enabled": false,
        "configurable": false
    },
    {
        "cipher_suite": "TLS_RSA_WITH_AES_256_CBC_SHA",
        "enabled": false,
        "configurable": false
    },
    {
        "cipher_suite": "TLS_RSA_WITH_AES_128_CBC_SHA",
        "enabled": false,
        "configurable": false
    }
],
"allow_unregistered": true
}

```

Updating NAE Interface

To modify an NAE interface, run:

Syntax

```
ksctl interfaces modify --type <Interface-type> --port <Port> --allow-unregistered <true/false>
```

Request

```
ksctl interfaces modify --type nae --port 9002 --allow-unregistered false
```

Response

```

{
  "id": "4540efac-8180-4265-bc81-75d3e1ce26d5",
  "name": "nae_all_9002",
  "mode": "unauth-tls-pw-req",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:1804704e-029c-4bd7-9d47-3c947c6d08fd",

```

```

"trusted_cas": {
  "local": [
    "kylo:kylo:naboo:localca:1804704e-029c-4bd7-9d47-3c947c6d08fd"
  ]
},
"createdAt": "2022-12-01T05:07:46.972399Z",
"updatedAt": "2022-12-01T05:08:50.697664Z",
"default_connection": "local_account",
"port": 9002,
"network_interface": "all",
"interface_type": "nae",
"minimum_tls_version": "tls_1_2",
"maximum_tls_version": "tls_1_3",
"local_auto_gen_attributes": {
  "cn": "nae_all_9002.ciphertrustmanager.local",
  "dns_names": [
    "nae_all_9002.ciphertrustmanager.local"
  ],
  "email_addresses": [
    "technical.support@thalesgroup.com"
  ],
  "names": [
    {
      "C": "US",
      "ST": "TX",
      "L": "Austin",
      "O": "Thales",
      "OU": ""
    }
  ],
  "generated": false
},
"enabled": true,
"meta": {},
"tls_ciphers": [
  {
    "cipher_suite": "TLS_AES_256_GCM_SHA384",
    "enabled": true,
    "configurable": false
  },
  {
    "cipher_suite": "TLS_CHACHA20_POLY1305_SHA256",
    "enabled": true,
    "configurable": false
  },
  {
    "cipher_suite": "TLS_AES_128_GCM_SHA256",
    "enabled": true,
    "configurable": false
  },
  {
    "cipher_suite": "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
    "enabled": true,
    "configurable": true
  },
  {
    "cipher_suite": "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
    "enabled": true,
    "configurable": true
  }
]

```

```

    },
    {
      "cipher_suite": "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
      "enabled": true,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
      "enabled": true,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
      "enabled": false,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
      "enabled": false,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_RSA_WITH_AES_128_CBC_SHA256",
      "enabled": false,
      "configurable": true
    },
    {
      "cipher_suite": "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA",
      "enabled": false,
      "configurable": false
    },
    {
      "cipher_suite": "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA",
      "enabled": false,
      "configurable": false
    },
    {
      "cipher_suite": "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA",
      "enabled": false,
      "configurable": false
    },
    {
      "cipher_suite": "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA",
      "enabled": false,
      "configurable": false
    },
    {
      "cipher_suite": "TLS_RSA_WITH_AES_256_CBC_SHA",
      "enabled": false,
      "configurable": false
    },
    {
      "cipher_suite": "TLS_RSA_WITH_AES_128_CBC_SHA",
      "enabled": false,
      "configurable": false
    }
  ],
  "allow_unregistered": false
}

```

Interface Certificate Expiration Check

The CipherTrust Manager inspects the expiration date of interface server certificates everyday, at a preset system time to log the record for the following interfaces:

- WEB
- NAE
- KMIP

The CipherTrust Manager then creates list of certificates based on their expiration date:

- Certificates whose expiration dates are within 91 days.

This list is logged in the **Records** section once every week.

- Certificates whose expiration dates are within 7 days.

This list is logged in the **Records** section once every day.

- Certificates that are already expired.

This list is logged in the **Records** section once every day.

You can also create alarm triggers for these records. For more details, go to [Creating Alarm Trigger for Interface Certificate Expiration](#).

Create/modify the NAE interface to mask system groups from NAE requests

Syntax

```
ksctl interfaces create --type nae --port <port number> --mask-system-groups <bool>
ksctl interfaces modify -n <interface-name> --mask-system-groups <bool>
```

Here:

- <interface name> : It is the name of the interface.
- <bool> : It can be either true or false.
- <port number> : It can be any positive number for which port is not allocated.

Note

System-defined groups can be masked if the `mask_system_groups` flag is set to `true` for the NAE interface. The groups will be masked for:

- `UserInfoRequest`
- `UserQueryRequest`
- `UserGroupInfoRequest`
- `UserGroupQueryRequest`

Example

```
ksctl interfaces modify -n nae_all_9001 --mask-system-groups true
```

Response

```
{
  "id": "513abafa-c22c-471f-a6b9-42762efab3dd",
  "name": "nae_all_9001",
  "mode": "no-tls-pw-opt",
  "cert_user_field": "CN",
  "default_connection": "local_account",
  "port": 9001,
  "network_interface": "all",
  "interface_type": "nae",
  "minimum_tls_version": "tls_1_2",
  "maximum_tls_version": "tls_1_3",
  "enabled": true,
  "meta": {
    "nae": {
      "mask_system_groups": true
    }
  }
}
```

Copying server certificate using use-certificate

The `use-certificate` command copies the server certificate from one interface to another. To copy the server certificate from the source interface to destination interface, run:

Example

```
ksctl interfaces use-certificate --name <destination-interface-name> --copy-from
<source-interface-name> --pending-renewal <true/false>
```



```
aTvpCyGHwOFQGuuy\nJzKnH6jG1u7qRF6JhKCPQwVNOmIyDENTGIhKxKnXZGNgssk7mpXnBmvK4wXOmC9
6\nm1Re5auo6hoorsk3JFPbgTT2zSZmwGf7CuzMuH+ZTvhlnJNdpqrkGffAdEe7rMPT\nBqVCS1iB9Isq
1lJBpBg1fcIXBDpKUdLHmZ+CNlHOPMqeYUJO89yd/CV8LQbN++ns\nagicBOm8UF0kFZmihydenrw7hC5
GKEO5r9rXv6YV9nC6RAREC5Ai7DPdsQG/eVaw\n7f/KKWtmTBYSSajo82xL/9C1E7NGqigfaq0Ly3dAjF
zHKRZMEt/rAmL9oD9XNuS1\nRWO6+JZN7BEXo5UiuaJg7kyW\n-----END CERTIFICATE-----\n"
}
```

In the response above, you can see the copied server certificate from the `kmip` interface.

If `pending-renewal` is set to `true`, the certificate will be saved as an [upcoming/renewed certificate](#) to be applied later

If `pending-renewal` is set to `false` (default behavior), the auto-generated server certificate will immediately apply on the server certificate resource and replace the currently active server certificate.

Users

Users are the authenticated entities that make KMIP, REST or NAE-XML calls. To authenticate to the NAE-XML interface, you send an `AuthRequest` with the user credentials. To authenticate to the REST API, you POST the user credentials to an authentication endpoint, and receive an API token, which you can use in subsequent API calls.

Users can be managed locally, or through an LDAP connection, or through an OIDC connection.

The operations which users are allowed to perform are controlled through [group membership](#).

Managing Users

Users can be created and managed through the web console, `ksctl` CLI, or REST API.

The legacy NAE-XML requests for user management are also supported, through the NAE-XML interface.

On launch, a single user exists, named "admin". This user can create, modify and delete other users. The user that is created in the `ksctl` setup is this admin user; see [CLI Installation](#).

The following options are available for managing users:

- Create
- Delete
- Get a single record
- List records
- Modify

Note

- You must set a `username` while creating a new user. It acts as the login credential for that user and cannot be changed. The `username` is *not* case-sensitive and *should not* include `/`, `|`, `\`, `<`, and `>` characters. The `username` field cannot be named as `global`.
- The `name` field is the full name of the user. It is optional and can be modified as required. The `user_id` field is created by the server and is used to update the fields of the user. The format of the `user_id` field is `local|<UUID>`.

Creating a Local User through GUI

Perform the following steps to create a user through the GUI:

1. Navigate to **Access Management > Users**.
2. Click **Add User** available on the top right corner of the screen. The **Add User** screen appears.
3. Select the **Connection Type** to create a user:
 - Local Account - Select this option and specify the following:
 - Full Name
 - Username
 - Email
 - Password
 - Password Match

Note

- You can optionally assign a [custom password policy](#) before or after creation. The global password policy is the default. Available custom password policies are specific to the [domain](#) of the new user.
- You can select the required checkbox to allow the user to log in using the web app, password, and/or certificate.
- If you select the checkbox **Allow user to login using certificate** only, it enables the certificate based authentication for logging in to the CipherTrust Manager. Consequently, the user will be restricted to login using browser-based web certificates.

- The default value for password hash iteration is *10000*. However, it is configurable and can be changed to *5000* using the `PASSWORD_HASH_ITERATIONS` system property.

The **Password** and **Password Match** fields are optional, if user selects only **Allow user to login using certificate** option and enter **Certificate Subject Distinguished Name**

Add User



Connection Type

☒ Local Account ☐ OIDC ☐ LDAP

Full Name

name

Username *

TestUser999

Email

email

Password

password

Password Match

passwordMatch

- ✖ Length is between 8 and 30 characters
- ✖ Has at least 1 uppercase(s)
- ✖ Has at least 1 lowercase(s)
- ✖ Has at least 1 number(s)
- ✖ Has at least 1 special character(s)

- ☐ Require user to reset password on next login
- ☒ Allow user to login using CipherTrust web app
- ☐ Allow user to login using password
- ☒ Allow user to login using certificate

Certificate Subject Distinguished Name

C=IN,ST=UP,L=Noida,O=CompanyName\,INC,OU=ENC,CN=TestUser999

Cancel

Add User

- OIDC - Select this option and specify the following:
 - OIDC Connection
 - Name
 - Email
- LDAP - Select this option and specify the following:
 - LDAP Connection
 - Name
 - Email

4. Click **Create**. A user is created and will appear in the list of users available on the **Users** page.

Creating a Local User through ksctl CLI

Use the below ksctl CLI to create a user named "bob":

```
ksctl users create --name bob --pword bob
```

The response looks like:

```
{
  "created_at": "2022-01-24T06:53:27.245528Z",
  "email": "bob@local",
  "last_login": null,
  "logins_count": 0,
  "name": "bob",
  "nickname": "bob",
  "updated_at": "2022-01-24T06:53:27.245528Z",
  "user_id": "local|c6537abe-a266-4833-9d60-b9713812136e",
  "username": "bob",
  "failed_logins_count": 0,
  "account_lockout_at": null,
  "failed_logins_initial_attempt_at": null,
  "last_failed_login_at": null,
  "password_changed_at": "2022-01-24T06:53:27.240767Z",
  "password_change_required": false,
  "certificate_subject_dn": "",
  "enable_cert_auth": false,
  "auth_domain": "00000000-0000-0000-0000-000000000000",
  "login_flags": {
    "prevent_ui_login": false
  },
  "allowed_auth_methods": [
    "password"
  ]
}
```

Setting Expiration Date for Users

The `expires_at` flag is added to set the expiration date for the local users to limit the user account to a certain period. This field is optional. After the expiration date has been reached, the access permissions for that user will be revoked. The user will no longer have access to the CipherTrust Manager. Only members of the *admin* and *User Admins* groups can add or modify the expiration date for an existing local user account. If the expiration date is extended or removed for an already expired account, the account will be active again.

Here are the examples of date-time format:

- 2002-10-02T10:00:00-05:00
- 2002-10-02T15:00:00Z
- 2002-10-02T15:00:00.05Z

The CipherTrust Manager throws an error if you set the expiration date before the current time. You can set the

expiration date for both new users as well as existing users.

For new users

For new users, run:

Request example 1 (create user with valid date)

```
ksctl users create --name "john" --expires-at "2023-01-30T10:30:35.421799Z"
```

Response

```
{
  "created_at": "2023-01-24T08:14:57.853232Z",
  "email": "john@local",
  "last_login": null,
  "logins_count": 0,
  "name": "john",
  "nickname": "john",
  "updated_at": "2023-01-24T08:14:57.853232Z",
  "user_id": "local|c3209b29-0343-4a10-a9b6-bd5674580518",
  "username": "john",
  "failed_logins_count": 0,
  "account_lockout_at": null,
  "failed_logins_initial_attempt_at": null,
  "last_failed_login_at": null,
  "password_changed_at": "2023-01-24T08:14:57.833508Z",
  "password_change_required": false,
  "certificate_subject_dn": "",
  "enable_cert_auth": false,
  "auth_domain": "00000000-0000-0000-0000-000000000000",
  "login_flags": {
    "prevent_ui_login": false
  },
  "allowed_auth_methods": [
    "password"
  ],
  "expires_at": "2023-01-30T10:30:35Z",
  "allowed_client_types": [
    "unregistered",
    "public",
    "confidential"
  ]
}
```

Request example 2 (create user with expiration date before the current time)

```
ksctl users create --name "james" --expires-at "2021-01-30T10:30:35.421799Z"
```

Response

```
{
  "code": 9,
  "codeDesc": "NCERRInvalidParamValue",
  "message": "expires_at cannot be before current time"
}
```

```
}
```

For existing users

For existing users, run:

Request example 1 (modify user with valid date)

```
ksctl users modify --id "local|c3209b29-0343-4a10-a9b6-bd5674580518" --expires-at "2023-01-28T10:30:35.421799Z"
```

Response

```
{
  "created_at": "2023-01-24T08:18:57.929628Z",
  "email": "john@local",
  "last_login": null,
  "logins_count": 0,
  "name": "john",
  "nickname": "john",
  "updated_at": "2023-01-24T08:18:57.929628Z",
  "user_id": "local|c3209b29-0343-4a10-a9b6-bd5674580518",
  "username": "john",
  "failed_logins_count": 0,
  "account_lockout_at": null,
  "failed_logins_initial_attempt_at": null,
  "last_failed_login_at": null,
  "password_changed_at": "2023-01-24T08:18:57.915523Z",
  "password_change_required": false,
  "certificate_subject_dn": "",
  "enable_cert_auth": false,
  "login_flags": {
    "prevent_ui_login": false
  },
  "allowed_auth_methods": [
    "password"
  ],
  "expires_at": "2023-01-28T10:30:35Z",
  "allowed_client_types": [
    "unregistered",
    "public",
    "confidential"
  ]
}
```

Request example 2 (modify user with invalid date)

```
ksctl users modify --id "local|cdb01da2-56d1-4d30-8c8a-4cc73a684b2a" --expires-at "1-01-2023"
```

Response

```
Invalid date format for Expiry date: parsing time "1-01-2023" as "2006-01-02T15:0
```



```
4:05Z07:00": cannot parse "-2023" as "2006"
```

Managing Allowed Client Types for Users

The `allowed_client_types` flag is added to manage the list of [client types](#) that are allowed to authenticate the users through their credentials. This field is optional.

By default, the CipherTrust Manager allows all clients types:

- **unregistered clients** - are not registered with the CipherTrust Manager.
- **public clients** - are ksctl, NAE, Web-UI, and API playground. These clients are pre-registered with the CipherTrust Manager.
- **confidential clients** - are able to securely authenticate with the CipherTrust Manager. For example, NAE, KMIP, and CTE clients.

If you set the `allowed_client_types` flag to empty, no client can authenticate the user to log on to the CipherTrust Manager.

You can set any combination of client types based on your requirements.

Note

The `allowed_client_types` property is **not** enforced on admin users.

Example Request

```
ksctl users create --username chanakya --pword KeySecure_1 --allowed-client-type
s confidential,public
```

Example Response

```
{
  "created_at": "2022-12-05T13:29:29.325474Z",
  "email": "chanakya@local",
  "last_login": null,
  "logins_count": 0,
  "name": "chanakya",
  "nickname": "chanakya",
  "updated_at": "2022-12-05T13:29:29.325474Z",
  "user_id": "local|726720ee-c597-4582-92c0-31f2c2f53947",
  "username": "chanakya",
  "failed_logins_count": 0,
  "account_lockout_at": null,
  "failed_logins_initial_attempt_at": null,
  "last_failed_login_at": null,
  "password_changed_at": "2022-12-05T13:29:29.321738Z",
  "password_change_required": false,
```

```

"certificate_subject_dn": "",
"enable_cert_auth": false,
"auth_domain": "00000000-0000-0000-0000-000000000000",
"login_flags": {
  "prevent_ui_login": false
},
"allowed_auth_methods": [
  "password"
],
"allowed_client_types": [
  "confidential",
  "public"
]
}

```

Managing Login Authentication Methods for Local Users

The CipherTrust Manager allows you to manage login authentication methods for local users using the `allowed_auth_methods` parameter. By default, this property is set to `password` that is, password-based authentication.

Note

Certificate-based authentication for local users requires [additional configuration](#)

The allowed login authentication methods are:

- `"password"` : Only password-based authentication. This is the default authentication method.
- `""` : Setting it to empty means no authentication method is allowed to the user.
- `"user_certificate"` : Only user-certificate based authentication.
- `"password", "user_certificate"` : Either of the authentication method is allowed.
- `"password_with_user_certificate"` : Both password and user certificate are required for the *two-factor* authentication and the value of the `certificate_subject_dn` flag should be set while creating the user.

Note

- The `password_with_user_certificate` method overrides any existing value in `allowed_auth_methods`.
- Password is optional when certificate-based authentication is enabled and certificate subject dn is provided while creating the user

- The allowed authentication methods are **not** enforced on admin users.
- If both `enable_cert_auth` and `allowed_auth_methods` parameters are provided in the request, the `enable_cert_auth` parameter is ignored.

Example Request 1 (Using `password,user_certificate`)

```
ksctl users create --username chandragupta --pword KeySecure_1 --allowed-auth-methods password,user_certificate
```

Example Response

```
{
  "created_at": "2022-12-05T13:27:47.317746Z",
  "email": "chandragupta@local",
  "last_login": null,
  "logins_count": 0,
  "name": "chandragupta",
  "nickname": "chandragupta",
  "updated_at": "2022-12-05T13:27:47.317746Z",
  "user_id": "local|d099eb70-e6fe-4a87-8e6e-49ff4ac93fb4",
  "username": "chandragupta",
  "failed_logins_count": 0,
  "account_lockout_at": null,
  "failed_logins_initial_attempt_at": null,
  "last_failed_login_at": null,
  "password_changed_at": "2022-12-05T13:27:47.314971Z",
  "password_change_required": false,
  "certificate_subject_dn": "",
  "enable_cert_auth": true,
  "auth_domain": "00000000-0000-0000-0000-000000000000",
  "login_flags": {
    "prevent_ui_login": false
  },
  "allowed_auth_methods": [
    "password",
    "user_certificate"
  ],
  "allowed_client_types": [
    "unregistered",
    "public",
    "confidential"
  ]
}
```

Example Request 2 (setting empty)

```
ksctl users create --username yum --pword KeySecure_1 --allowed-auth-methods ""
```

Example Response

```
{
  "created_at": "2023-02-03T07:08:37.308092Z",
  "email": "yum@local",
  "last_login": null,
  "logins_count": 0,
  "name": "yum",
  "nickname": "yum",
  "updated_at": "2023-02-03T07:08:37.308092Z",
  "user_id": "local|10244efe-afdc-4759-b9e0-ea6144f7e843",
  "username": "yum",
  "failed_logins_count": 0,
  "account_lockout_at": null,
  "failed_logins_initial_attempt_at": null,
  "last_failed_login_at": null,
  "password_changed_at": "2023-02-03T07:08:37.292784Z",
  "password_change_required": false,
  "certificate_subject_dn": "",
  "enable_cert_auth": false,
  "auth_domain": "00000000-0000-0000-0000-000000000000",
  "login_flags": {
    "prevent_ui_login": false
  },
  "allowed_auth_methods": [],
  "allowed_client_types": [
    "unregistered",
    "public",
    "confidential"
  ]
}
```

Manually Creating LDAP or OIDC Users

CipherTrust Manager can integrate with external authentication and identity providers using [LDAP](#) or [OIDC](#) protocols.

After you configure an LDAP or OIDC connection, by default logging in with the associated users for the first time automatically creates matching CipherTrust Manager user accounts. However, you can also manually create these CipherTrust Manager user accounts prior to first login. As well, there a connection setting called **Disable Automatic Creation of a User** which disables this default behavior and enforces that OIDC or LDAP users are manually created on CipherTrust Manager before first login.

Provide the `--userconnection` parameter to the `ksctl users create` to specify an LDAP or OIDC connection.

For example:

```
ksctl users create --name bob --pword bob --userconnection LDAP-connection
```

For OIDC authentication, the username of a manually created user must match the `sub` claim returned by the identity provider.

Managing Users in Domain

The CipherTrust Manager allows you to create users in a non-root domain if you set the `allow-user-mgmt` property to `true` while creating the domain. To create a user within a non-root domain, set the

`is_domain_user` property to `true`.

Before proceeding to the next steps, we recommend you to get familiar with the following terms:

- **auth_domain** - The `auth_domain` is a domain where the user is created and authenticated during login. The `auth_domain` is represented by the **Home Domain** field on the CipherTrust Manager GUI.

If a user is created in a domain named '*Dom1*', the `auth_domain` for that user will be '*Dom1*'. However, if a user is created in a root domain and assigned to '*Dom1*', the default `auth_domain` for that user will always remain as `root`.

- **Target domain** - The target domain is represented by the `domain` field on CLI and API.
- **Assigned users** - Users assigned from the root domain to any non-root domain. For assigned users, the `auth_domain` is always `root`. By default, if the `auth_domain` is not provided, it is assumed to be `root`.
- **Non-root domain** - Domains other than the root domain.

Login/Token Creation

For login/token creation, the domain users need to pass `auth_domain` with username and password.

Login Directly to a Child Domain

1. Select the **I am a domain user** check box.
2. Specify your auth domain in the **Home Domain** field.
3. Provide your **Username** and **Password**, if applicable.
4. Click **Log In** or **Log In with Certificate Only**.

The domain users are always redirected to their auth domain. However, when assigned users log on to the CipherTrust Web UI for the first time, they are redirected to the root domain. Next time onward, after login, these users are redirected to their last logged-in domain on the GUI.

Note

- While creating a domain, users created in the non-root domain cannot be added to the domains admin's list.
- Users created in the non-root domain cannot be assigned to any other domain.
- Certificate-based login does not work for the users created in the non-root domain.
- OIDC and LDAP authentication are only supported for users created in the root domain. If a user is created in a root domain and later assigned to an additional domain, LDAP and OIDC authentication is

available for login to the root domain, and after login the user can switch to a non-root domain. For OIDC authentication, the user manually logs into the root domain and then manually switches to a non-root domain in two steps. For LDAP authentication, the user can login to the root domain and switch into a non-root domain in a [single step](#).

Switch Domains After Login

In the CipherTrust Web UI, in the upper right hand corner, click the name of the current domain, click **Switch Domains**, and click the name of the desired domain. You are now taken to the specified domain. From this point forward, all key management and admin activity is constrained to the selected domain.



Preventing Users from UI Login

The CipherTrust Manager allows you to prevent the UI login for *users* if you set the `prevent_ui_login` property to `true` while creating the user. By default, this property is set to `false`.

You can change the value of `prevent_ui_login` property while updating the user.

Configuring LDAP Authentication

Note

This feature is not included in Community Edition and requires a valid Virtual CipherTrust Manager license. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

The CipherTrust Manager authenticates users against its internal database and also provides an option to authenticate with an external authentication provider (LDAP). It is possible to have a mix of both local and LDAP users.

Users in the Systems Admins and Admins group, such as the `admin` user, can configure an LDAP connection in the root domain. When LDAP is configured the users and credentials are located in an external LDAP server and CipherTrust Manager acts as a forwarder to the LDAP server. CipherTrust Manager uses simple bind authentication.

By default, the first time a new LDAP user is successfully authenticated, a corresponding user account is stored in the CipherTrust Manager root domain and is marked to belong to the respective LDAP connection. You can also create user accounts for LDAP users prior to authentication in the root domain. There is an LDAP connection setting called **Disable Automatic Creation of a User** which disables this default behavior and enforces that

LDAP users are manually created on CipherTrust Manager before first login. Only users created in the root domain can use LDAP authentication. Users that belong to additional domains can login to the root domain and switch into a non-root domain in a single step as described [below](#).

All CipherTrust Manager user accounts associated with a LDAP connection are deleted when the LDAP connection is deleted from the CipherTrust Manager.

Viewing, editing, and assigning permissions to an LDAP-Authenticated user

CipherTrust Manager administrators can control the allowed operations for the user account with [group](#) and [domain](#) assignment in the same way as with local users. If the LDAP connection has fields for groupmaps configured, administrators can configure [groupmaps](#) to associate a group on the LDAP directory to a local CipherTrust Manager group, which are applied when the user logs in.

Note

- LDAP user access to CipherTrust Manager resources is only controlled through CipherTrust Manager group and domain settings. CipherTrust Manager cannot read or consider any user permission setting from the LDAP server.
- The **Protected Users** group associated with Active Directory is not supported on CipherTrust Manager as the group is not compatible with simple bind authentication. Users in this group cannot login to CipherTrust Manager.

LDAPS Protocol

You can configure the LDAPS protocol, which uses certificate authentication to secure the connection. With LDAPS connections, CipherTrust Manager verifies presented LDAP server certificates, by default checking that the certificates are issued by major well-known certificate authorities (CAs). Alternatively, you can directly provide root certificate authorities to trust for the specific LDAPS connection, which is recommended if the LDAPS connection is secured by a lesser-known root CA or a self-signed certificate. You can also disable server certificate verification entirely, for troubleshooting purposes.

Setting Up a New LDAP Connection Through Web Console UI

1. Login to the root domain as a user in the Systems Admins and Admins group, such as the `admin` user.
2. Navigate to **Access Management > LDAP**.

Note

Do not navigate to the **Connections** page. The LDAP connection available through Access Management described here is separate and additional to the [LDAP connection available through](#)

connection manager.

3. Click **+ Add LDAP** to add a new connection.
4. Provide the required values for any LDAP connection:
 - **Connection Name** - Unique identifier for the connection and commonly referred to as the domain, for example "myco".
 - **Server URL** - Hostname and port of the LDAP server, for example "ldap://192.168.0.1:389". You can optionally enable LDAPS protocol by specifying `ldaps`. Options for LDAPS authentication are described in the step 7.
 - **Root DN for users** - The base DN as the starting point for the search, for example "DC=myco,DC=local".
 - **User's login name attribute** - The field containing the user's login name, for example `uid`.
5. If you wish to enable `groupmaps`, provide the following values:
 - **Bind DN** - If authentication is required to perform search this is the name of the entity to bind to, for example CN=poweruser,CN=Users,DC=myco,DC=local.
 - **Server Bind Password** - Password of the user provided in **Bind DN**.
 - **Root DN for groups** - Starting point to use when searching for groups.
 - **Search filter for listing groups** - For example, "(objectclass=Group)".
 - **Group id attribute for group mapping** - The field that contains the group name, for example "cn".
 - **User's group membership attribute** - The field which contains a user's group membership information, for example, "member".
6. (Optional) Set the following values if desired:
 - **Disable Automatic Creation of a User** - Check the checkbox to disable the default automatic creation of a CipherTrust Manager user when the user logs in via LDAP. Disallows an unknown user from logging in. Enforces that a member of the *admin* or *User Admins* group must manually create a CipherTrust Manager user before the user can log in.
 - **Search filter for listing users** - Optional LDAP search filter which can further restrict the set of users who will be allowed to log in, for example `memberOf="CN=example,OU=Users,DC=myco,DC=local"`.
 - **User's distinguished name** - Attribute inside the user object which contains the user distinguished name. If this value is not provided, CipherTrust Manager attempts to determine default value based on **User's login name attribute**. If **User's login name attribute** is provided as a `sAMAccountName`, Active Directory

configuration is assumed and "distinguishedName" is used as default. Otherwise, **User's distinguished name** defaults to "dn".

- **Unique Identifier Field** - The LDAP field name of a globally unique identifier for the group. If this is not specified, the value the LDAP server uses **User's login name attribute** instead. However, on **User's login name attribute** update, **Unique Identifier Field** does not update automatically.
7. (LDAPS connections only). You can optionally specify some values for LDAPS connections. The options are ignored for LDAP connections, which have the `ldaps` protocol in the **Server URL** and send traffic in cleartext.
- **Server Certificate** - A list of root CAs for CipherTrust Manager to verify the LDAP server certificate. The provided root CAs are used instead of the operating system trust store to verify the LDAP server certificate. Root CAs must be in PEM encoded format. You can upload the root CAs list as a file or provide them as text.
 - **Disable verifying the server certificate** - Check the checkbox to disable CipherTrust Manager's verification of the LDAP server certificate.

Caution

This setting lowers the security of the connection and can disrupt LDAPS connection availability, and so is only recommended for troubleshooting purposes.

8. Provide the following values required to test the connection:
- **Password for test**
 - **Username for test**
9. Click **Test Connection** to ensure CipherTrust Manager can connect to the LDAP server with the provided settings.
10. Click **Add LDAP** to confirm the connection creation.

Setting Up a New LDAP Connection Through ksctl CLI

Using either the CLI or the GUI it is possible to test a connection, create a new connection, get a list of connections, modify a connection, and delete a connection.

Note

The LDAP connection available through Access Management described here is separate and additional to the [LDAP connection available through connection manager](#).

Required fields for LDAP connection

- `name` - Unique identifier for the connection and commonly referred to as the domain, for example `myco`.
- `server-url` - Hostname and port of the LDAP server, for example `ldap://192.168.0.1:389`. You can enable LDAPS protocol with `ldaps`. There are [options to specify how the LDAP server certificate is verified](#).
- `root-dn` - The base DN as the starting point for the search, for example `DC=myco,DC=local`.
- `uid-field` - The field containing the user's login name, for example `uid`.

Fields for Groupmaps for LDAP Connection

You can optionally set up [groupmaps](#) for an LDAP connection. All of the following fields are required for groupmap configuration:

- `bind-dn` - If authentication is required to perform search this is the name of the entity to bind to, for example `CN=poweruser,CN=Users,DC=myco,DC=local`.
- `bind-password` - Password of the user provided in `bind-dn`.
- `group-base-dn` - Starting point to use when searching for groups.
- `group-id-field` - The field that contains the group name, for example `'cn'`.
- `group-filter` - Search filter for listing groups.
- `group-member-field` - The field which contains a user's group membership information, for example `'member'`.

Optional field for LDAP connection

- `guid-field` - The LDAP field name of a globally unique identifier for the group. If this is not specified, the value the LDAP server uses `uid-field` instead. However, on `uid_field` update, `guid_field` does not update automatically.
- `search-filter` - Optional LDAP search filter which can further restrict the set of users who will be allowed to log in, for example `memberOf="CN=example,OU=Users,DC=myco,DC=local"`.
- `disable-auto-create` - When set to `true`, disables the default automatic creation of a CipherTrust Manager user when the user logs in via LDAP. Disallows an unknown user from logging in. Enforces that a member of the *admin* or *User Admins* group must manually create a CipherTrust Manager user before the user can log in.
- `user_dn_field` - Attribute inside the user object which contains the user distinguished name. If this value is not provided, CipherTrust Manager attempts to determine default value based on `uid_field`. If `uid_field` is provided as a `sAMAccountName`, Active Directory configuration is assumed and `"distinguishedName"` is used as default. Otherwise, `user_dn_field` defaults to `"dn"`.

Optional fields for LDAPS connections

The following options are ignored for LDAP connections, which have the `ldap` protocol in the `server-url` and send traffic in cleartext.

- `insecure-skip-verify` - When set to `true`, this option disables CipherTrust Manager's verification of the LDAP server certificate.

Caution

Setting to `true` lowers the security of the connection and can disrupt LDAPS connection availability, and so is only recommended for troubleshooting purposes.

- `root-cas` - This field can contain a list of root CAs for CipherTrust Manager to verify the LDAP server certificate. The provided root CAs are used instead of the operating system trust store to verify the LDAP server certificate. Root CAs must be in PEM encoded format.

Once a connection is created, it is not possible to modify these fields: `name`, `strategy` or `bind-pass`.

CLI command example

Test the connection configuration to verify it can authenticate one of the LDAP users.

```
ksctl connections test --test-username=jdoe --test-password=ji32k7au4a83 --name=myco --strategy=ldap --server-url="ldap://192.168.0.1:389" --user-id-field=uid --root-dn="DC=myco,DC=local" --search-filter="(memberOf=CN=example,OU=Users,DC=myco,DC=local")
```

Create the connection configuration.

```
ksctl connections create ldap --name=myco --server-url="ldap://192.168.0.1:389" --user-id-field=uid --root-dn="DC=myco,DC=local" --search-filter="(memberOf=CN=example,OU=Users,DC=myco,DC=local")
```

Logging in a User with LDAP Authentication

Each LDAP connection is given a name that needs to be known by the end user and be provided during authentication; this is often referred to as the domain. For example, if a user **joe** exists in the LDAP directory `ldap://192.168.0.1:389` and this LDAP server is created in CipherTrust Manager with the name **myco**, the end user would have to authenticate with either **myco\joe**, **myco/joe** or **myco|joe** in the GUI. On the CLI, you would instead specify the name of the connection as a separate parameter.

LDAP authentication also works through NAE-XML. The connection and/or username can be specified in following formats:

- `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**.

Note

The user must be created in the root domain and assigned to additional domains to successfully login with this syntax.

When browsing the GUI, LDAP 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>**

Logging out a User with LDAP Authentication

LDAP users are logged out using standard CipherTrust Manager logout as used for any user.

To logout a user from CipherTrust Manager:

User Interface	Action
WebUI or embedded API Guide	Navigate to the WebUI, click on the username in the top right corner, and click Logout
ksctl CLI	Use <code>ksctl logout</code> command.
External REST API client	Use <code>POST /v1/auth/logout</code> method.

Configuring OpenID Connect (OIDC) Authentication

Note

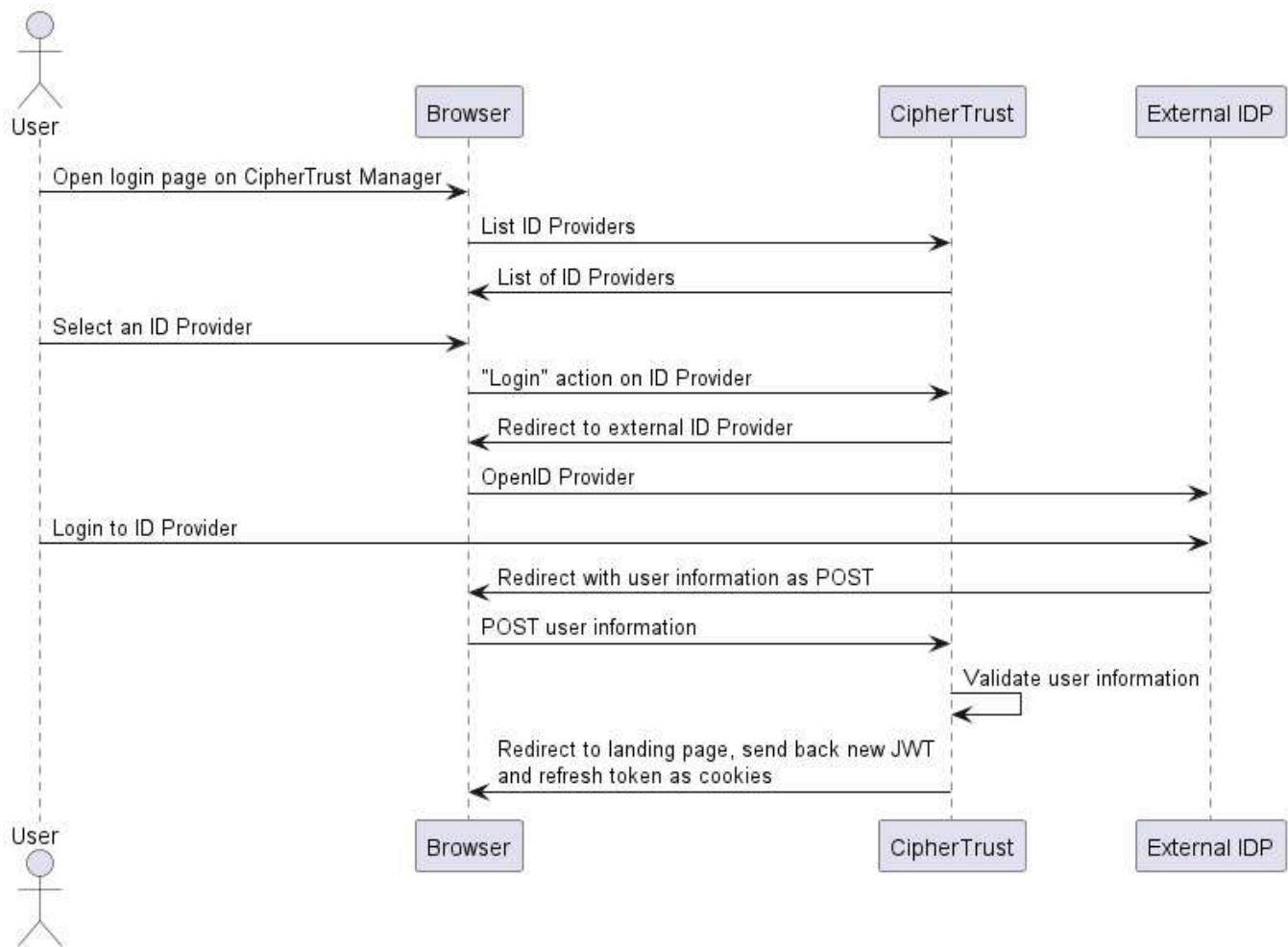
This feature is not included in Community Edition and requires a valid Virtual CipherTrust Manager license. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

CipherTrust Manager supports OpenID Connect as an external identity provider for CipherTrust Manager user login through the web console UI, embedded API Guide, or external REST API client, using either **implicit flow with form post** or **authorization code flow**.

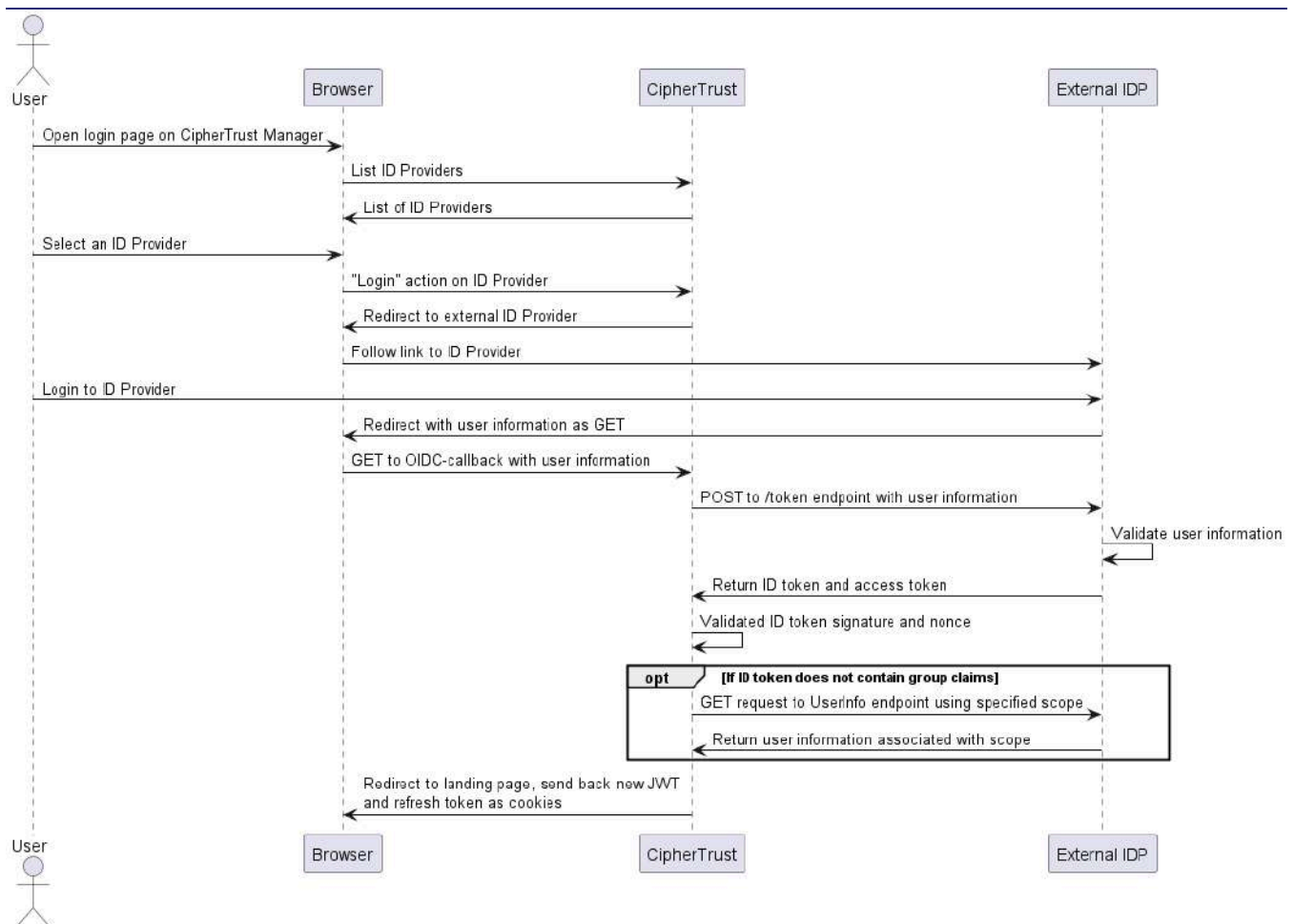
Note

- The OIDC connection to authenticate CipherTrust Manager users is separate from the OIDC connection to authenticate CTE agents, which is configured through [connection manager](#).
- OIDC authentication is not supported with the ksctl CLI.

The following sequence diagram demonstrates how with **implicit flow**, browser login through the CipherTrust Manager GUI returns an ID token from an external identity provider.



The following sequence diagram demonstrates how with **authorization code flow**, browser login through the CipherTrust Manager GUI returns an ID token and other user information from an external identity provider.



Users are created on the external identity provider. To set up authentication for these users and allow them to access CipherTrust Manager and perform operations, there are three stages:

1. Configure the connection between CipherTrust Manager and the identity provider.

You need access rights on the identity provider to create an application and assign users to it, and Admin level access to the CipherTrust Manager to configure the connection details.

2. Create a user account on CipherTrust Manager for a user of the application in the root domain. Alternatively, logging in to CipherTrust Manager as a user of the application automatically creates a user account visible on CipherTrust Manager root domain.

Note

OIDC-authenticated users must be created in the root domain. They can later be assigned to additional non-root domains.

3. As a CipherTrust Manager assign the user account to groups and domains to allow the desired operations, or create a [group map](#) to associate the OIDC group membership to CipherTrust Manager groups.

Supported Flows, Scopes, and Claims

Supported flows:

- Implicit flow with form post
- Authorization code flow

Requested type:

- `response_type=id_token` for implicit flow
- `response_type=code` for authorization code flow

Requested mode:

- `response_mode=form_post` for implicit flow
- `response_mode=query` for authorization code flow

Supported scopes:

- The scope value must be 254 characters or fewer, and can contain spaces, double quotes ("), and backslashes(\), and printable ASCII characters. If the scope does not include `openid`, CipherTrust Manager adds it to the scope values passed to the OpenID provider.
- If no expected scope is configured on CipherTrust Manager, the value defaults to `openid email profile`.

The following standard claims are required to be returned by the external identity provider:

- `sub`: used as the username and part of the user ID.

The following standard claims are optional and are used if returned by the external identity provider:

- `name`: used as the "name" property of a user. If missing it falls back to user ID (which effectively means `sub`).
- `email`: used as the "email" property of a user. If missing it is empty.

To read more about standard scopes:

https://openid.net/specs/openid-connect-core-1_0.html#ScopeClaims

To read more about standard claims:

https://openid.net/specs/openid-connect-core-1_0.html#StandardClaims

Determination of OIDC Group Membership

By default, the OIDC connection is set up to allow, but not require, [group mapping](#). CipherTrust Manager checks for group membership in the ID token or returned from the UserInfo Endpoint. If no group membership can be obtained from either source, no CipherTrust Manager group is assigned.

By default, CipherTrust Manager checks the `groups` claim field in the ID token. When you create the OIDC

connection, you can specify another claim field in the ID token which contains the group membership.

CipherTrust Manager first checks the claim field parameter in the ID token. If no group membership is found in the claim, and the OIDC connection is configured for authorization code flow, CipherTrust Manager then attempts to get group membership from the identity provider's UserInfo Endpoint. This attempt to get group membership from the UserInfo Endpoint is not supported for implicit flow.

CipherTrust Manager obtains the UserInfo endpoint from the configured Discovery URI, so there is no need to provide the UserInfo endpoint value. There is an optional `--userinfo-endpoint` parameter in the ksctl CLI to override this behavior, which is intended for test deployments only, and not for use in production.

To configure an identity provider connection using web console UI

Note

The OIDC connection to authenticate CipherTrust Manager users is separate from the OIDC connection to authenticate CTE agents, which is configured through [connection manager](#).

1. Create a client application on the identity provider to service user logins.
 - a. Ensure the application is set to either implicit or authorization code flow and provide the redirect URI in the format `https://<ciphertrust-hostname>/api/v1/auth/oidc-callback`. If you have a CipherTrust Manager cluster, provide one redirect URI for each node. For example, provide `https://<node1-hostname>/api/v1/auth/oidc-callback` and `https://<node2-hostname>/api/v1/auth/oidc-callback` for two CipherTrust Manager nodes.
 - b. Assign users to your application that correspond to desired CipherTrust Manager users.
 - c. If desired, you can configure the application to provide group membership information, so that CipherTrust Manager can [map group membership](#) in the OIDC application to a local CipherTrust Manager group. Refer to [Determination of OIDC Group Membership](#) for all available configuration options.
 - d. After creating the application, you need to collect the following values from the identity provider:
 - **Client ID:** The client identifier of CipherTrust Manager on the identity provider.
 - **Discovery URI:** This location contains detailed settings for CipherTrust Manager to construct a login request to the identity provider, such as authorization URI and public signing keys. This is also called "well known configuration URL" in some identity providers.
 - **Client Secret:** Needed for Authorization Code Flow only. The identity provider requires this value in the login request to validate. This is sensitive information, so record it in a secure manner.
 - If you set up a user identity claim for group membership in the last step, and the claim field name is different than `groups`, collect the claim field name as well.
2. Login to the root domain as a user in the Systems Admins and Admins group, such as the `admin` user.

3. Navigate to **Access Management > OpenID Connect**.

Note

Do not navigate to the **Connections** page.

4. Click **Add Connection**.

5. Provide the following details:

- **Name** : A desired connection name.
- **Discovery URI**: The location required for CipherTrust Manager to construct login requests, obtained in step 1.
- **Client ID**: The client identifier of CipherTrust Manager on the identity provider, obtained in step 1.

6. Provide a **Groups Claim**, if in step 1 you set up a claim for group membership in the ID token and the claim field name is different than "groups". If you leave this field blank the value defaults to "groups".

7. Provide a **Username Claim**, and set username claim to extract the username from the OIDC ID Token. If unspecified, it defaults to `sub`. In the OIDC ID Token below, username claim can be set as `email` if needed. The claim field can also be a template to support the nested claims (if any) in the OIDC ID Token. For example, the OIDC ID token:

```
{
  "sub": "1117309839505746486071",
  "email": "test@xyz.com",
  "custom": {
    "username": "xyz"
  }
}
```

`{{ .custom.username }}` can be set as username claim to set the username of the CipherTrust Manager user after successful authentication on the external identity provider. Ensure that the provided username claim is unique and not modifiable in the external identity provider.

8. Configure the desired flow type.

- For implicit flow, make sure the **Implicit Flow** radio button is selected for the **Flow Type**.
- For authorization code flow, make sure the **Authorization Code Flow** radio button is selected for the **Flow Type**. Provide the **Client Secret** obtained from the identity provider in step 1.

9. Provide the **Redirect URI** obtained from the identity provider in step 1.

Click **+Add Redirect URI** if you have multiple redirect URIs to add.

10. (Optional) Select the **Disable Automatic Creation of a User** checkbox to disable the automatic creation of a CipherTrust Manager user by default when the user logs in via OIDC. This means an unknown user cannot log in. The CipherTrust Manager enforces that a member of the *admin* or *User Admins* group must manually create a CipherTrust Manager user before the user can log in.
11. (Optional) Provide one or more values for **Scope**.
12. Click **Add Connection** to create the connection.
13. Ensure that a **Network Protocol Server (NTP)** is configured for CipherTrust Manager. NTP configuration is crucial for successful OIDC authentication.

To configure an identity provider connection using ksctl CLI

1. Create a client application on the identity provider to service user logins.
 - a. Ensure the application is set to either implicit or authorization code flow and provide the redirect URI in the format `https://<ciphertrust-hostname>/api/v1/auth/oidc-callback`. If you have a CipherTrust Manager cluster, provide one redirect URI for each node. For example, provide `https://<node1-hostname>/api/v1/auth/oidc-callback` and `https://<node2-hostname>/api/v1/auth/oidc-callback` for two CipherTrust Manager nodes.
 - b. Assign users to your application that correspond to desired CipherTrust Manager users.
 - c. If desired, you can configure the application to provide group membership information, so that CipherTrust Manager can [map group membership](#) in the OIDC application to a local CipherTrust Manager group. Refer to [Determination of OIDC Group Membership](#) for all available configuration options.
 - d. Set `username_claim` to extract the username from the OIDC ID Token. If unspecified, it defaults to `sub`. In the OIDC ID Token below, `username_claim` can be set as `email` if needed. The claim field can also be a template to support the nested claims (if any) in the OIDC ID Token. For example, OIDC ID token:

```
{
  "sub": "1117309839505746486071",
  "email": "test@xyz.com",
  "custom": {
    "username": "xyz"
  }
}
```

`{{ .custom.username }}` can be set as `username_claim` to set the username of the CipherTrust Manager user after successful authentication on the external identity provider. Ensure that the provided

`username_claim` is unique and not modifiable in the external identity provider.

e. After creating the application, you need to collect the following values from the identity provider:

- **Client ID:** The client identifier of CipherTrust Manager on the identity provider.
- **Discovery URI:** This location contains detailed settings for CipherTrust Manager to construct a login request to the identity provider, such as authorization URI and public signing keys. This is also called "well known configuration URL" in some identity providers.
- **Client Secret:** Needed for Authorization Code Flow only. The identity provider requires this value in the login request to validate. This is sensitive information, so record it in a secure manner.
- If you set up a claim for group membership in the ID token, and the claim field name is different than `groups`, collect the claim field name as well.
- If you have set up a **Scope** other than `openid email profile`, collect this value as well.

2. On CipherTrust Manager as a user in the Systems Admins and Admins group, such as the `admin` user, create the OIDC connection. Provide the client ID, redirect URI, and discovery URI. Give the connection a name to display during user login. If the scope is different than `openid email profile`, provide this as well.

For authentication code flow, set the `flow-type` to `authorization_code` and provide the client secret as well.

Note

`flow-type` defaults to `implicit` and so does not need to be set for implicit flow connections.

If the identity provider provides group membership using a claim in the ID token besides `groups`, include the option `--groups-claim`, and provide the claim field name.

Implicit flow, no custom groups claim field:

```
ksctl connections create oidc --name '<desired_name>' --client-id <client_id_from_identity_provider> --redirect-uris <ciphertrust_hostname>/api/v1/auth/oidc-callback --discovery-uri <discovery_uri_from_identity_provider> [--scope <scope_value>]
```

Authorization code flow, no custom groups claim field:

```
ksctl connections create oidc --name '<desired_name>' --client-id <client_id_f
```

```
rom_identity_provider> --redirect-uris <ciphertrust_hostname>/api/v1/auth/oidc-callback --discovery-uri <discovery_uri_from_identity_provider> --client-secret <client_secret_value> --flow-type authorization_code [--scope <scope_value>]
```

Group mapping to a non-default claim field name, with implicit flow:

```
ksctl connections create oidc --name '<desired_name>' --client-id <client_id_from_identity_provider> --redirect-uris <ciphertrust_hostname>/api/v1/auth/oidc-callback --discovery-uri <discovery_uri_from_identity_provider> --groups-claim <claim_field_name> [--scope <scope_value>]
```

Username Claim:

```
ksctl connections create oidc --name '<desired_name>' --client-id <client_id_from_identity_provider> --redirect-uris <ciphertrust_hostname>/api/v1/auth/oidc-callback --discovery-uri <discovery_uri_from_identity_provider> --username-claim <claim_field_name> [--scope <scope_value>]
```

Note

You can provide multiple redirect URIs by repeating the `redirect-uris` parameter. For example `--redirect-uris https://node1.com/api/v1/auth/oidc-callback --redirect-uris https://node2.com/api/v1/auth/oidc-callback`.

Note

If you do not provide a discovery URI, the following values must be provided instead. These settings are recommended for testing scenarios only. Discovery URI is strongly recommended for production usage of OIDC.

- `authorization-uri`: Authorization endpoint for the identity provider.
- `jwks`: Public signing keys to validate the ID token.

There is an additional optional test parameter for authorization code flow:

- `token-endpoint`: Token endpoint for the identity provider, for authorization code flow only.

Note

There is an option, `disable-auto-create`. When set to `true`, disables the default automatic creation of a CipherTrust Manager user when the user logs in via OIDC. Disallows an unknown user from logging in. Enforces that a member of the *admin* or *User Admins* group must manually create a CipherTrust Manager user before the user can log in.

3. Network Time Protocol (NTP) configuration is very important for successful OIDC authentication. Ensure that an NTP server is configured for CipherTrust Manager.

To login a user with OIDC authentication

A user can login to the identity provider through the CipherTrust Manager web console to the root domain.

Note

For OIDC users, the special characters "/" and "|" are allowed in the `username` field.

1. On the login page, select a registered identity provider from the **Identity Providers** dropdown.

Note

Any username or password entered from the CipherTrust Manager login page is ignored. CipherTrust Manager-level username/password authentication and OIDC authentication are mutually exclusive.

2. You are redirected to the identity provider's login page. At this point, enter your credentials.

Note

Sometimes your browser caches the previously logged in OIDC-authenticated user. Clear your browser cache if you want to login a different user through the OIDC flow.

After authentication with the identity provider succeeds, you are redirected back to the CipherTrust Manager web console.

To logout an OIDC-authenticated user

CipherTrust Manager's logout operations use single logout (known as single sign-out on some identity providers) for OIDC-authenticated users. This means that logging an OIDC user out of CipherTrust Manager also logs the user out at the identity provider, immediately logging the user out of all client applications to the identity provider.

To logout a user from CipherTrust Manager:

User Interface	Action
WebUI or embedded API Guide	Navigate to the WebUI, click on the username in the top right corner, and click Logout
External REST API client	Use <code>POST /v1/auth/logout</code> method. Consult the REST API specification for details on OIDC authentication options.

To view, edit, and assign permissions to an OIDC-authenticated user

By default, once the user logs into CipherTrust Manager at least once, a user account is created in CipherTrust Manager with the OIDC connection name displayed as the source. You can also [create a CipherTrust Manager user account manually prior to first login](#).

There is an OIDC connection setting called **Disable Automatic Creation of a User** which disables this default behavior and enforces that OIDC users are manually created on CipherTrust Manager before first login.

CipherTrust Manager administrators can control the allowed operations for the user with [group](#) and [domain](#) assignment in the same way as with local users. If the OIDC connection has a group claim configured, administrators can configure [group maps](#) to associate a group on the OIDC server to a local CipherTrust Manager group, which are applied when the user logs in.

Deleting an OIDC user on CipherTrust Manager removes those CipherTrust Manager permissions associated with the user. However, the user can still login to CipherTrust Manager as long as that user is still assigned to the application on the identity provider. The user must be deleted or unassigned at the identity provider application to completely remove any access to CipherTrust Manager.

Certificate Based Authentication for Users

The CipherTrust Manager authenticates a user's login request by verifying the username and password against its internal database. The CipherTrust Manager can also be configured to authenticate login requests using browser-based web certificates. This section elaborates upon the steps that you need to perform to enable Certificate based Authentication for logging in the CipherTrust Manager.

Note

Currently, this feature is applicable to local users only. LDAP users are not supported.

Step 1: Enable the "Certificate based Login" Option for a User

1. Log on to CipherTrust Manager as an administrator. Navigate to **Access Management > Users**.

2. Enable the "Certificate based Login" option for the user:

Note

For certificate based login to work in any domain for a user:

- a. Add the domain issuer CA of that user certificate as an external CA in the root domain. Ensure the `user_authentication` for that CA is enabled.
- b. Add this CA to the list of Trusted CAs of the Web interface in the root domain.
- c. Restart the web service (**Admin Settings --> Services**).

Note

The CipherTrust Manager supports fields such as Domain Component (DC) and Given Name (GN) in the Distinguished Name for certificate-based login. Local CAs do not support the DC and GN fields, so they cannot issue certificates with these fields. Use external CA for issuing certificates with DC and GN fields.

Example:

`DC=DomainComponent,C=IN,ST=State,L=Location,O=Thales,OU=CipherTrust,GN=GivenName,CN=CommonName`

Note

For root domain users, the CipherTrust Manager doesn't have the option to pass the target domain from UI. The same is applicable for Password and Certificate based login. To pass the domain in user request, use API or curl.

For domain users, pass the `auth_domain` name in the **Home Domain** field.

The image shows the Thales login interface. At the top is the Thales logo. Below it, there are two main login sections separated by an 'OR'.

First Login Section:

- Username:** A text input field with placeholder text 'username or connection/username'.
- Password:** A text input field with placeholder text 'password'.
- ☐ I am a domain user
- Log In** button.

Second Login Section:

- Login with certificate only** header.
- ☒ I am a domain user
- Home Domain:** A text input field with placeholder text 'Home Domain'.
- Log In with Certificate Only** button.

At the bottom, a note states: 'By clicking Log In, you are accepting [this EULA](#)'.

Specifying a Common Name (CN) is mandatory for this feature to work. The entities must be specified carefully in this field, and separated by commas (,).

For example:

O=Thales,OU=CipherTrust,CN=User_1

If Distinguished Name (DN) field contains values that are separated by comma, then those values must be followed by a backslash (\).

Caution

For example:

C=IN,ST=UP,L=Noida,O=CompanyName\,INC,OU=ENC,CN=test

As per RFC 5280, a multivalued RDN (for instance - OU) should be arranged in a sorted order (using octet string sort), that is, the values with the shorter lengths should come first followed by a lexicographic sorting.

Note

All the certificates created by the CipherTrust Manager follow this arrangement.

For example:

While issuing certificate, if user provides the following subject DN as an input:

O=Thales, OU=XipherTrust, OU=X, OU=CipherTrust, CN=User_1

The issued certificate will have the subject DN set as:

O=Thales, OU=X, OU=CipherTrust, OU=XipherTrust, CN=User_1

- **For existing users:**

- Click the action button for that user, then click **Manage**.
- Click **CONFIGURE CERTIFICATE LOGIN**. Select **Allow user to login using certificate**.
- Specify **Certificate Subject Distinguished Name** for the user.
- Click **Update Certificate Login**.

- **For new users:**

- Click **Create New User**. Specify **Username** and **Password** for the user.
- Select **Allow user to login using certificate**.
- Specify **Certificate Subject Distinguished Name** for the user.
- Click **Create**.

Note

If you select the checkbox **Allow user to login using certificate** only, it enables the certificate based authentication for logging in to the CipherTrust Manager. Consequently, the user will be restricted to login using browser-based web certificates. For more information, refer [Creating a User through GUI](#).

Step 2: Create and Download the Web Certificate

If using Local CA

- If using local CA, Go to **CA > Local** and select the desired CA. If CA doesn't exist, [create a new CA](#).
- Click **Issue Certificate**.

-
- a. Enter the **Common Name** for this certificate.

Note

This common name should be the same common name that you specified while creating the user.

- b. Select the desired algorithm (RSA or ECDSA).
- c. In the **Name** field, specify the same details that you specified in the `certificate_subject_dn` property of the user.

Note

If *subject DN* of a certificate and a CA are *same* then certificate is treated as a self signed certificate. However, the self signed certificates are not recognized by the Web UI of a CipherTrust Manager for certificate login.

- d. Click **Issue Certificate**.
 - e. Click **save private key** to download the `key.pem` file.
 - f. Click **Issue Certificate**. The newly created certificate is displayed in the certificates list.
3. Download the certificate issued by the local CA and save it at the same location where the private key is saved.
 4. Add the CA to the list of [Local Trusted CAs](#) of the Web interface.
 5. Restart the web service from the **Admin Settings --> Services** section. Restarting the web service can take few seconds.

If using External CA

1. Upload the external CA.
2. Navigate to **Admin Settings --> Interfaces**. The Interface Configuration page is displayed.
3. Click the action button on the Web Interface Configuration, select **View/Edit**.
4. Add the external CA to the list of [External Trusted CAs](#) of the Web interface.
5. Restart the web service from the **Admin Settings --> Services** section. Restarting the web service can take few seconds.

Step 3: Create and Install pkcs12 Formatted Certificate

1. Install OpenSSL on your machine.
2. Use the following command to convert the key and certificate into a pkcs12 formatted .pfx file:

```
openssl pkcs12 -export -out example.pfx -inkey key.pem -in certificate.pem
```

Where:

- **key.pem** is the private key
- **certificate.pem** is the certificate file
- **example.pfx** is the pkcs12 formatted web certificate that will be installed in the web browser

This creates a .pfx certificate (*example.pfx* in the above command) at the same location.

3. Go to the web browser's settings.
4. Import and install the .pfx certificate.

You can now use the web certificate for logging on to CipherTrust Manager. Before logging on, you will be prompted to select the web certificate at the login page.

Note

After performing the above steps, if you are not prompted by the browser to select the certificate at login page, then clear your browser's cookies and try again.

Note

CipherTrust Manager verifies the revocation status of the certificate using the Certificate Revocation List (CRL) method before login. If the certificate is revoked, the CipherTrust Manager returns an unauthorized error message stating "cannot log in the user with a revoked certificate". In case, CRL URL is not reachable, then the CipherTrust Manager logs a message and lets the user login successfully.

Managing Your Own User Account Settings

There are shortcuts for viewing and editing your own user account compared to [managing other users](#).

These settings are available for any user in the CipherTrust Manager, regardless of permission level. To view and manage settings for other user accounts, you must be part of the User Admins [group](#).

View Your User Details

1. Log into CipherTrust WebUI.
2. Click on your full name in the upper right corner.
3. Click **User Settings**.

The **My User Settings** page is displayed.

4. View the following values:
 - **Full Name** - A friendly name to display. This value is different than the username you provide to login.
 - **Email** - Your email address
 - **Source** - Can be **local**, or the name of an **LDAP** or **OIDC** connection depending on the source of your account.
 - **Created** - Creation date and time.
 - **Updated** - Date and time when your settings were last updated.
 - **ID** - The UUID associated with your account. This value is helpful for operations to update your account in the REST API. If your account's source is LDAP or OIDC, the connection name and the `|` character precede the UUID.
 - **Last Login** - Date and time when you last logged in.
 - **Last Failed Login** - The last date and time when someone tried unsuccessfully to login with your user name.
 - **Logins** - Total number of logins.
 - **Failed Logins** - Total failed number of logins.
 - **Expires At** - The date and time the user account will expire and you will no longer have access to CipherTrust Manager.

Note

This value is only visible if the user administrator configured your account with an expiration date. If the **Expires At** detail doesn't have a value, your account will never expire.

Update Your Full Name

1. Log into CipherTrust Manager CipherTrust WebUI.
2. Click on your full name in the upper right corner.
3. Click **User Settings**.

The **My User Settings** page is displayed.

4. In the **UPDATE NAME** section, type a new value in the **Full Name** field.

Note

This value is different than the username you provide to login.

5. Click **Update** under the **Full Name** field to apply the change.

Update Your Email Address

1. Log into CipherTrust WebUI.
2. Click on your full name in the upper right corner.
3. Click **User Settings**.

The **My User Settings** page is displayed.

4. In the **UPDATE EMAIL** section, type a new value in the **Email** field.
5. Click **Update** under the **Email** field to apply the change.

Change Your Password

Note

This operation is not applicable to LDAP and OIDC users.

1. Log into CipherTrust Manager web console GUI.
 2. Click on your full name in the upper right corner.
 3. Click **User Settings**.
-

The **My User Settings** page is displayed.

- Under the **CHANGE PASSWORD** section, provide the **Current Password**, a new value for the **Password** and confirm the new password with the **Password Match** field.

The new password must adhere to policy rules defining allowed password length and characters. The policy rules are listed below the **Password** and **Password Match** fields. The [user administrator sets the password policy](#).

- Click the **Change Password** button to apply the change.

You are immediately logged out, and must log in again with the new password.

Authentication Tokens

The CipherTrust Manager supports two types of authentication tokens:

- API Tokens (JWT)
- Refresh Tokens

Both authentication tokens can be issued (created) using username and password or client credentials, but the API Token (JWT) can alternatively be issued using a refresh token. API Tokens (JWT) are short lived tokens and are used for accessing the REST API as a "Bearer" token. API Tokens (JWT) are valid for 5 minutes.

Refresh tokens are long lived tokens and can be used as an alternative way to issue API Tokens (JWT). One typical use case for refresh tokens is for a long lived browser session where a user enters credentials to get a refresh token. This allows the browser to automatically issue new API Tokens (JWT) using the refresh token during the session.

Refresh Token Duration

Refresh tokens have both a timeout due to inactivity, and an absolute lifetime beyond which the token expires.

The following timeouts due to inactivity apply to the CipherTrust Manager user interfaces:

- CipherTrust Web UI and Embedded API Guide:** An idle browser session logs out the user after 10 minutes of inactivity, due to an additional timeout unrelated to the refresh token. The refresh token timeout is 30 minutes, and applies in the case that the user has closed the browser window without logging out. In that situation, the user can navigate back to the CipherTrust Manager URL and access the CipherTrust Web UI or API Guide without logging in again, if the refresh token is still active.
- ksctl CLI:** By default, refresh tokens do not timeout due to inactivity. If you manually issue a refresh token with the `ksctl tokens create` command, there is an optional parameter `--rt-unused-life` to set the timeout in minutes for inactivity.

By default, there is no absolute lifetime for refresh token issued through the CipherTrust Web UI or the `ksctl tokens create` command. The `ksctl login` command sets the lifetime at 30 days by default. Both `ksctl` commands have options to specify the refresh token lifetime.

There is a [system property](#), `MAXIMUM_REFRESH_TOKEN_LIFETIME`, accessible through the `v1/configs/`

properties/ API endpoint or the `ksctl properties` commands. By default, `MAXIMUM_REFRESH_TOKEN_LIFETIME` is not set. When set, this system property applies to every CipherTrust Manager user interface, and overrides any other configuration related to refresh token lifetime, preventing users from creating refresh tokens which have a longer lifetime than the set maximum.

API Tokens (JWT)

Tokens are string values used to authenticate calls to the REST API. You use an endpoint to trade a user credential for an API token:

```
GET /api/v1/auth/tokens/
```

Example: Token

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJkMzEzYzZlLTQ5MTU0OGJkNy01ZmMyYmUyNzk2YzEiLCJzdBWIIiOiJsb2Nhbmh4ImZjOGZlNy1hN2ZlLTQ1YzEtOWU1OS0zYmUxNTRkMTZjYmQiLCJpc3MiOiJreWxvIiwiaWF0IjoiIjoia3lsbyIsImN1c3QiOiJ3Z3JvdXBzIjpbImFkbWluIiwiaWF0IjoiIjE5LCJpYXQiOiJlODExMjY0ODAsImV4cCI6I6MTQ4MTEyNDU0MH0.xTWlkVnYTiCAzJ_ZL746F2HB5HAqxOdnHJ-Qyjq_K84
```

JWT Signing Key

The CipherTrust Manager 2.15 onward, the default JWT signing key is ECDSA(ES256). Previously it was HMAC(HS256).

To rotate a authentication key:

Example - for ECDSA

```
ksctl tokens rotate-auth-key --type ecdsa --curve p256
```

Example - for HMAC

```
ksctl tokens rotate-auth-key --type hmac
```

Note

- When you deploy a new instance of CipherTrust Manager 2.15 or above, it generates a new ECDSA authentication key. In case of upgrading to CipherTrust Manager 2.15 or above, the authentication key type remains the same as before.
- When you rotate the JWT signing key without passing authentication key type, the default rotation key type will be ECDSA(ES256).

Refresh tokens are longer lived tokens that can be used instead of user or client credentials for requesting a new API access token when the current API access token becomes invalid or expires. The refresh token might expire as well and should be renewed periodically depending on the nature of the client application.

Example: Refresh token

Example use cases for refresh tokens:

- ### To issue an API token (JWT) using ksctl:

- Issue using username and password:

Refresh token reply (example):

- Issue using a refresh token:

Response:

```
{
  "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJkOWNhZTVlOC0xNDJmLTQ2NmQwOTRlZS1iODI5YmQ4ZTc0YWQlLCJzdBWiiOiJsb2Nhbmhw5MTdiNTNmYy1kMjA5LTQ4OTUuYmFizS03NmM3MTF1NDI5ZGMiLCJpc3MiOiJreWxvIiwiaWF0Ijoi2023-09-01T12:00:00Z",
}
```



```
I6ImFkbWluIiwiaWF0IjoxNTMwODk1Nzg4LCJleHAiOjE1MzA4OTYwODh9.JMRtLevUUv5YOsxg4BJhHmt5pyzlvwxOTWLD49EzTU4",
  "duration": 300,
  "token_type": "Bearer"
}
```

- Assign the API Token (JWT) to the environment variable using a refresh token:

```
export KSCTL_JWT=$(ksctl tokens create --refresh-token e7UySu7fj8ZAwPKVIjEGwOTm
t8AidxFHK8Wc4tL0RiaHVf7sd3oBQ74y2M8pPhNI | jq -r ".jwt")
```

To refresh the refresh token using ksctl:

Use this command to refresh the refresh token without the username and password.

```
$ ./ksctl tokens create --issue-rt --refresh-token cIb6w1PzazBnJM35v9QyP6LWN70lGn
GfNVUMGQfzjrcY8Hfv9cJjuMIXFHqNio9g
```

Response:

```
{
  "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiI2NzI3ZWQ1Ni0yNjxxLTRkMz
ctOGNhMC1iYmJhMzNhMjBkNzMiLCJzZW50IjoxNTMwODk1Nzg4LCJleHAiOjE1MzA4OTYwODh9.JMRtLevUUv5YOsxg4BJhHmt5pyzlvwxOTWLD49EzTU4",
  "duration": 300,
  "token_type": "Bearer",
  "refresh_token": "1xy4tVzBzjrzsVho4xWQAtN5Whil3a71eH6nYMTi4VNSIEWXHv7NpKGAItdQR
Uip"
}
```

Revoke Refresh Token

A refresh token can be revoked by passing the refresh token and client id or user credentials to the following endpoints:

```
POST /api/v1/auth/revoke
```

A revoked refresh token can no longer be used to renew API tokens.

Deleting Refresh Token for Users with Multiple Logins

Users with active multiple logins can encounter a banner in the web console UI, "Multiple Logins Detected".

Warning

If your organization suspects the account is compromised, take actions to change the user credentials immediately.

- Change the [password](#). Users can change their own passwords, or an administrator in the User Admins or Admins [group](#) can do so for the user account. This action revokes all active tokens.
- For certificate-authenticated users, an administrator in the CA Admins or Admins group can [use a local certificate authority to revoke the certificate](#) for the user.

If you do not suspect an account compromise, you can selectively delete tokens for unnecessary user sessions.

A user might have active refresh tokens for the following login types:

- Web UI login with the user credentials
- ksctl CLI login with the user credentials
- API playground log with the user credentials
- NAE-XML clients using the user's credentials for user delegation or impersonation
- KMIP clients using the user's credentials for user delegation or impersonation

To delete refresh tokens:

1. Login to the API playground as an Application Administrator, such as `admin`.

Do not specify a domain or an `auth_domain`.

2. Obtain the user ID, which is different than the username. The user ID is the user's authentication type, such as `local|`, followed by a 32 character alphanumeric randomly generated value.

REST API

List all users, using the query parameter `username` to filter to the user's name. Copy the value of `"user_id"` in the response.

Syntax

```
GET /v1/usermgmt/users?username=<username_value>
```

Example command

```
GET /v1/usermgmt/users?username=user1
```

Web Console UI

- a. Login to the web console UI as a User Admin, such as `admin`.
 - b. Navigate to **Access Management>Users**.
 - c. Search for the user name in the list.
 - d. View the **ID** column.
 - e. Click **Copy** next to the value to copy.
3. List the authentication tokens for the user, using the user ID value as a query parameter to filter.

Syntax

```
GET /v1/auth/tokens?user_id=<user_id_value>
```

Example Command

```
GET /v1/auth/tokens?user_id="local|21bf2cb1-b535-46b9-aaa0-593def1ed385"
```

4. Examine the list tokens response to view what interfaces the tokens are associated with. This can help you identify which tokens are no longer needed.
 - The `labels` field indicates tokens issued for web console UI access and for NAE-XML and KMIP clients using the user's credentials. Potential values are `web-ui` and `nae-auth`.
 - `client_id` indicates the client identity associated with the token. The following public client identities are valid for CipherTrust Web UI and API playground tokens, ksctl CLI tokens, and tokens for **unregistered** NAE-XML clients acting as public clients:

client id	client name
c5890024-a6d4-408d-a592-5d4d5807c722	nae
17771cf2-f80b-4eb5-a19b-a2d0032179c3	web-ui and api-playground
5ffb6fac-2cb5-4b91-8183-e20ad3b62577	ksctl

Registered NAE-XML clients, and all KMIP clients, have individual client identities which were set up on client registration. To confirm the client identity and look up other details about these clients:

REST API

List all clients.

Syntax

```
GET /v1/client-management/clients
```

The `id` value for each client confirms the client identity.

Web Console UI

- a. Login to the web console UI as a Client Admin, such as `admin`.
 - b. Navigate to **Access Management>Client Hub**.
 - c. View the **ID** column. This represents valid client identity values.
 - d. Click on a client name to obtain more details about that client.
5. For tokens you wish to delete, look in the list all tokens response for `id` values, and retain those.
 6. Delete each token individually, using the `id` value for the token.

Syntax

```
DELETE /v1/auth/tokens/<token_id>
```

Example Command

```
DELETE /v1/auth/tokens/a52bf108-d214-40b8-8889-2a32dbc5e42c
```

Login Command

In the `ksctl` CLI, use the `login` command to create and save a long lived refresh token that makes it easier to access your CipherTrust Manager server. This method is safer than storing a password in the `config.yaml` file because a refresh token can easily be revoked and renewed if ever exposed. For more information on refresh tokens, refer to [Refresh Tokens](#).

Configuration file with a saved refresh token

The `login` command can be used as an alternative to creating a `config.yaml` file as defined in the [CLI Installation](#). When using the `login` command, if no `config.yaml` file exists, one is created in the `.ksctl` subdirectory of the user's home directory, or the configuration file that is specified in the `login` command is used. See examples below.

The `login` command requires:

- User name
- URL of the CipherTrust Manager instance
- Flag indicating that SSL verification should not take place

Example 1: When no `config.yaml` file exists

```
$ ksctl login --user <username> --url https://<ip_address_or_hostname> --noSSLverify
```

Note

Provide the name or the IP address of the host for the CipherTrust Manager instance.

You are then prompted for the password. If the password is successful, you will see the following message:

```
this login will expire in 30 days
```

This command also creates a config.yaml file similar to the following:

```
KSCTL_VERBOSITY: false
KSCTL_RESP: json
KSCTL_URL: https://ip_address_or_hostname
KSCTL_USERNAME: admin
KSCTL_REFRESH_TOKEN: icmtiodYvNfCAR5XXXODlgJjATkYPpYs3oHRXXXCdGWhGpSXXX70x3AQF1tkc
pGO
KSCTL_CLIENTID: ffbbd794-68a3-4da9-83bc-9b0dd7a3ee48
KSCTL_NOSSLVERIFY: true
```

Example 2: When specifying the config.yaml file to use

The following `login` command can be run to create a custom configuration file by adding the `configfile` variable:

```
$ ksctl login --user <username> --url https://<kylo-system> --noSSLverify --configfile localconfig.yaml
```

As in Example 1, you are prompted for the password. On successful creation of the refresh token, a custom configuration file is created with the name `localconfig.yaml`.

Example 3: When `config.yaml` file already exists

When the URL and the username or refresh token are already stored in a config.yaml file, the following command can be run:

```
$ ksctl login
```

Example 4: Initial `login` command using password

```
$ ksctl login --user <username>
```

You are prompted for a password. If successful, a refresh token with a default lifetime of 30 days is created. You will see the following message:

```
$ this login will expire in 30 days
```

The refresh token is written to the `config.yaml` file to be used by subsequent invocations of `ksctl`.

Note

The `config.yaml` file stores an additional information, `KSCTL_CLIENTID`. This is used by the `login` command and should not be deleted or modified.

Refresh token lifetime

If you do not specify the lifetime for a refresh token, a default of 30 days is used.

Example 1: Specifying a refresh token lifetime in days

Use the `--days` option to specify the number of days the token should be valid.

```
$ ksctl login --user admin --days 10
```

Returned information:

```
this login will expire in 10 days
```

Example 2: Specifying a refresh token lifetime in hours

Use the `--hours` option to specify the number of hours the token should be valid.

```
$ ksctl login --user admin --hours 10
```

Returned information:

```
this login will expire in 10 hours
```

Example 3: Specifying a refresh token lifetime in days and hours

Use both the `--days` and `--hours` options to specify the number of days and hours the token should be valid.

```
$ ksctl login --user admin --days 10 --hours 10
```

Returned information:

```
this login will expire in 10 days, 10 hours
```

Summary of `login` command options

The following options can be provided in the `login` command. Each option used overrides the corresponding value in the configuration file.

- `--user` : required only when there is no configuration file. If provided, the value is saved to the configuration file.
- `--password` : never required, never saved to the configuration file.
- `--url` : required only when there is no configuration file. If provided, the value is saved to the configuration file.
- `--nosslverify` : required when there is no configuration file. If provided, the value is saved to the configuration file.
- `--verbose` : never required, used during invocation, and never saved to the configuration file.

If you get the following error, it indicates that your ssl verification is being done and that you did not specify `-nosslverify` in the `login` command.

```
Post https://localhost/api/v1/auth/tokens: x509: certificate is not valid for any names, but wanted to match localhost
```

Revoke/refresh an existing token

If a token already exists in the configuration file when a `login` command is invoked, the existing token will be revoked/refreshed automatically.

Logout Command

The `logout` command reads the configuration file and revokes the login token found in it. This command does not require any options.

```
$ ksctl logout
```

If the login token is successfully revoked, the following message is returned:

```
login revoked
```

If there is no login refresh token or client id present in the file, the following message is returned:

```
login token and client id must exist in ksctl configuration file
```

Client Management

A client represents a machine where a supported product such as ProtectFile or CipherTrust Transparent Encryption (CTE) is installed. It can also represent a generic client using the REST, KMIP, or NAE protocol.

All clients with supported product installations (for example, ProtectFile, and CTE) can be managed on the CipherTrust Manager. An Administrator can register many clients with the CipherTrust Manager, view registered clients, view and modify details, revoke registrations, renew clients, and delete clients when they are no longer needed. We recommend to register a client whenever possible.

Client registration is supported with:

- CTE Agents
- CTE Userspace Agents
- CTE for Kubernetes Agents
- DPG
- Custom KMIP clients
- Custom NAE clients
- Custom REST API clients

Registering a client with CipherTrust Manager allows you to better track client activity, through the assigned client identity.

The [Loki audit records](#) contain a `client_id` value for every action a registered or public client takes.

ProtectFile, CTE, CipherTrust Data Protection Gateway (DPG), and KMIP clients need to be registered with the CipherTrust Manager to successfully authenticate and perform key and cryptographic operations. When successfully registered, the client is automatically added to the CipherTrust Manager. CipherTrust Manager currently allows unregistered clients of other types.

A client can be registered with the CipherTrust Manager by using a registration token generated on the CipherTrust Manager. This token is called the client registration token. The fingerprint of the server's web interface certificate is also required for registering ProtectFile and CTE Userspace clients with the CipherTrust Manager.

The CipherTrust Manager provides options to manage registration tokens and clients.

Note

For client authentication to work in any domain, the issuer CA of that client certificate must be added in the domain and `client_authentication` for that CA must be enabled. This is applicable only for the clients with `authentication_mode` set as `dn`.

Warning

As soon as a client is deleted from the CipherTrust Manager, all communication between the CipherTrust Manager and the client will stop immediately. It is recommended that the clients are decrypted before deletion, otherwise the encrypted data will become inaccessible.

Public Clients

The CipherTrust Manager pre-registers the following public clients:

- ksctl
- Web-UI and API playground
- NAE

These public clients are shipped with the hard-coded client ids and names as follows:

client id	client name
c5890024-a6d4-408d-a592-5d4d5807c722	nae
17771cf2-f80b-4eb5-a19b-a2d0032179c3	web-ui and api-playground
5ffb6fac-2cb5-4b91-8183-e20ad3b62577	ksctl

The client ids are pre-embedded in the respective public clients. When public clients call the `/v1/auth/tokens` endpoint API to generate a registration token, it includes the `client_id` parameter with the `password` grant type. The `client_id` and `client_name` are present in the issued JWT after the successful authentication.

These client ids are included in [Loki audit records](#).

Registering Clients

Note

During client registration process, if a certificate is presented, it undergoes a revocation check using the Certificate Revocation List (CRL). If the certificate is determined to be revoked, the registration process cannot proceed. An audit record is generated to notify the same. This applies to all client types and is relevant for both automatic and manual registration processes.

The following table lists the parameters that are required when managing a client on the CipherTrust Manager:

Parameter	Description
Name	Name for the client to display on the CipherTrust Manager.
Registration Token	Registration token to register the client with the CipherTrust Manager.

For the most security, the client also provides a Certificate Signing Request (CSR) to be signed by a CipherTrust Manager local certification authority (CA).

The high-level registration flow varies based on product or protocol. CTE, CTE-U, CTE for Kubernetes, custom KMIP clients, custom NAE clients, and custom REST API clients require you to create a client profile, which contains settings for the CA which issues the client certificate. Other connectors obtain registration tokens through the Application Data Protection tile, and manage settings for the CA there. For these connectors, a client profile is

created for display purposes after registration.

Note

On KMIP and REST interfaces, if the `cert_duration` parameter is specified in both the client profile and registration token, then preference is given to the one specified in the *client profile*. Otherwise, its value is set to the system's default value, that is, 730 days.

Note

On KMIP and REST interfaces, if the `ca-id` parameter is specified in both the client profile and registration token, then preference is given to the one specified in the *registration token*. Otherwise, its value is set to the system's default local CA value.

High-Level Registration Flow for CTE, DSF Hubs, and Custom Clients

Note

KMIP and NAE have the option to automatically register their clients using existing certificates. This follows a different process than the manual registration described here. Consult KMIP or NAE documentation for information on automatic registration.

For DSF Hubs clients, have a user in this DSF Hubs [group](#) perform these steps to ensure that the client is registered as a DSF Hubs client with the associated permissions.

1. [Create a client profile](#), which manages the CA that issue certificates for clients associated with the client profile. The three certificate authority settings options are, from most to least secure:
 - Provide a CSR issued by the client, to be signed by a CipherTrust Manager local CA.
 - Set an [external CA](#) which has been added to CipherTrust Manager.
 - If your client is not capable of generating a key pair or CSR, generate a CSR on CipherTrust Manager and use a local CA to sign it. Download the generated certificate and store it on the client.
2. [Create a registration token](#) associated with the client profile. Retain this registration token.

Note

DSF Hubs clients require a registration token generated through the `ksctl` CLI or REST API, because the

required `labels` parameter is not available in the CipherTrust WebUI. The `labels` value must be set to `"ncryptify-reserved/dsf-managed": "true"`.

3. Have the client issue a client registration request to CipherTrust Manager, including the registration token, and relevant certificate-related parameters. The way you initiate this request is specific to the client type, but in all cases, you provide the registration token.

The client-side registration for specific client types is described at the following locations:

- [Custom Rest API Client Registration](#)
 - [Installing and Registering CTE](#)
 - [Install and Configure CTE-U with CipherTrust Manager](#)
 - [Managing Kubernetes Clients \(CTE for Kubernetes\)](#)
 - [KMIP Client Registration](#)
 - [NAE Client Registration](#)
4. (DSF Hubs clients only) Assign the client to the DSF Hubs group. This operation is only available in the REST API or `ksctl` CLI.

```
ksctl groups addclient --clientid <id_of_registered_dsf_hubs_client> --name "DSF Hubs"
```

High-Level Registration Flow for Other Connectors

1. Add an application for a [supported client](#). As part of this process, you provide necessary CA and certificate settings, and receive a registration token.
2. Use the registration token with the client to register the client with CipherTrust Manager.

After registration, you can view clients' configurations and check their health status through the single pane of glass.

Refer to your client documentation for end-to-point registration and deployment flow.

Tokens

A token is a string that is used to [register clients](#) with the CipherTrust Manager. The CipherTrust Manager provides options to create tokens, view existing tokens, view and modify their details, and delete them when they are no longer required.

Note

The registration token can be created without CA Admins privilege. Such tokens can only be used with the default Local CA. To use the token with non-default CA, it is mandatory to add the user in the CA Admins group. This enables the user to view the list of CAs in the domain.

Creating a Registration Token

You can create a registration token through the CipherTrust WebUI, ksctl CLI or REST API.

Note

DSF Hubs clients require a registration token generated through the ksctl CLI or REST API, because the required `labels` parameter is not available in the CipherTrust WebUI.

Creating a Registration Token Using CipherTrust WebUI

A registration token can be created on the CipherTrust Manager. It is used when manually registering CipherTrust Manager clients with the CipherTrust Manager. A CipherTrust Manager administrator creates registration tokens.

To create a registration token for CTE, CTE-U, CTE for Kubernetes, KMIP, NAE and Custom REST Clients:

Note

For other connectors, registration token is created while adding an application for a [supported client](#).

1. Log on to the CipherTrust Manager Web UI as a member of the `admins` or `Client Admins` group.
2. In the left pane, click **Access Management > Registration Tokens**.
3. On the right, click **Add Registration Token**. The Create New Registration Token wizard is displayed. This is a three-step wizard.
4. Click **Begin** to start token creation. The Configure Token screen is displayed.
5. (Optional) Specify a **Name Prefix**. Prefix for the client name. This prefix is used to construct names for clients whose names are not specified during registration with the CipherTrust Manager using this token.
 - If the name prefix is specified as `ks_client`, client names will be constructed as `ks_client#`; for example, `ks_client1`, `ks_client2`, `ks_client3`, and so on.
 - If the name prefix is **not** specified, the CipherTrust Manager will construct a *random* name for clients.

However, if a client's name is specified during registration, this name prefix will not be used for that client.

6. Specify **Token lifetime**. This is the duration (in minutes, hours, or days) for which this token can be used for registering clients. For example, specify the lifetime as:
 - `1 minutes` for 1 minute
 - `2 hours` for 2 hours
 - `3 days` for 3 days
 - `unlimited` for a token that never expires.

By default, the token lifetime is `unlimited`. The token will *never* expire.

7. Specify **Client Capacity**. This is the maximum number of clients that can be registered using this registration token. The default capacity is `100` clients.
8. Click **Select Profile**. The Select Profile screen is displayed.
9. Select the desired **Client Profile** from the drop-down list. A client profile allows the CipherTrust Manager administrators to manage the local CAs that issue certificates for the clients associated with the given client profile.

If no client profile exists, you can create one before proceeding. Refer to [Managing Client Profiles](#) for details.

Note

- If you automate registration token creation using the `/api/v1/client-management/regtokens` API endpoint, it is recommended to provide value for the `client_management_profile_id` attribute that points to an existing client profile. KMIP and NAE clients can automate registration in this way.
- This is not enforced for backward compatibility reasons.

10. Click **Create Token**. The Create Token screen is displayed. The screen displays the generated registration token in **ASCII** and **Base64** encoding. The CipherTrust Manager accepts the registration token in ASCII format only. Select **Base64** if the client application accepts the token in **Base64** format only and converts the token to ASCII before sending the it to the CipherTrust Manager.
 1. Select the desired encoding.
 2. Click **Copy** next to the token. Save the copied token. This token will be used when registering and migrating clients.
 3. Click **Close**.

Note

While creating a registration token, the CA chain will be validated for the intermediate local CAs.

Managing Registration Token using ksctl

Note

It is recommended to use the value of `cert_duration` specified in the [Client Profile](#). Going forward, the `cert_duration` parameter will be deprecated from the registration token.

This section provides instructions to create and manage registration tokens on the CipherTrust Manager.

The following operations can be performed:

- Create/Get/Update/Delete registration tokens
- List all client profiles

Creating Registration Token

To create a registration token, run

Syntax

```
ksctl clientmgmt tokens create --ca_id <ca-id> --cert-duration <validity-duration-of-certificate> --client-mgmt-profile-id <client-profile-id> --life-time <lifetime-of-token> --max-clients <maximum-number-of-client> --name-prefix <name-prefix> [--labels ]
```

- `ca-id` - ID of the trusted Certificate Authority that is used to sign client certificate during registration process.
- `cert-duration` - duration (in days) for which the CipherTrust Manager client certificate is valid.
- `client-mgmt-profile-id` - id of the client profile.
- `labels` - Labels (key/value pairs) associated with the resource. The labels can be supplied as a list of comma separated key/value pairs of the form "[key=value,key2=value2...]".

DSF Hubs clients require the `labels` value of `"ncryptify-reserved/dsf-managed": "true"`. This label is also automatically applied to File Activity Monitoring (FAM) resources. FAM resources cannot be deleted or modified through CipherTrust Manager, only through the DSF platform.

Each part of a label, the prefix, name, and value, must begin and end with an alphanumeric value. Characters

between the beginning and end can contain alphanumeric characters, periods (.), hyphens (-), and underscores (_).

A key has an optional prefix of 253 characters or fewer, followed by a forward slash, and a required name of 63 characters or fewer. For example, a valid key is `sales.widgets.com/region`.

A value can be empty, or a value of 63 characters or fewer.

- `life-time` - duration (in minutes, hours, or days) for which this token can be used for registering clients.
- `max-clients` - maximum number of clients that can be registered using this registration token.
- `name-prefix` - prefix for the client name. This prefix is used to construct names for clients whose names are not specified during registration with the CipherTrust Manager using this token. However, if a client's name is specified during registration, this name prefix will not be used for that client.

Example Request

```
ksctl clientmgmt tokens create --ca-id "950f1ef4-7b47-4cbb-9bfb-adee2f2c0283" --cert-duration 2 --client-mgmt-profile-id "ee910055-b8fc-42c2-9c67-49bdbd0f67b2" --life-time "10h" --max-clients 50 --name-prefix "test_prefix"
```

Example Response

```
{
  "id": "669457e6-89cc-482d-8cb5-3b35f2b1a89c",
  "uri": "kylo:kylo:munshi:tokens:669457e6-89cc-482d-8cb5-3b35f2b1a89c",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-11-30T20:35:05.841579Z",
  "updatedAt": "2023-11-30T20:35:05.841579Z",
  "token": "4u9xQXsTknLcGT2byBrY4IJCOp40gWKIhEP7jxEUFJM27QDM1RgE26Fz43oSog1j",
  "valid_until": "2023-12-01T06:35:05.839849438Z",
  "max_clients": 50,
  "cert_duration": 2,
  "clients_registered": 0,
  "ca_id": "950f1ef4-7b47-4cbb-9bfb-adee2f2c0283",
  "name_prefix": "test_prefix",
  "label": null,
  "client_management_profile_id": "ee910055-b8fc-42c2-9c67-49bdbd0f67b2"
}
```

Getting Registration Token Details

To get details of a registration token, run:

Syntax

```
ksctl clientmgmt tokens get --token-id <token-id>
```

Example Request

```
ksctl clientmgmt tokens get --token-id "669457e6-89cc-482d-8cb5-3b35f2b1a89c"
```

Example Response

```
{
  "id": "669457e6-89cc-482d-8cb5-3b35f2b1a89c",
  "uri": "kylo:kylo:munshi:tokens:669457e6-89cc-482d-8cb5-3b35f2b1a89c",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-11-30T20:35:05.841579Z",
  "updatedAt": "2023-11-30T20:38:50.171348Z",
  "token": "4u9xQXsTknLcGT2byBrY4IJCOp40gWKIhEP7jxEUFJM27QDM1RgE26Fz43oSog1j",
  "valid_until": "2023-12-01T01:38:50.170988Z",
  "max_clients": 20,
  "cert_duration": 2,
  "clients_registered": 0,
  "ca_id": "",
  "name_prefix": "test_prefix",
  "label": null,
  "client_management_profile_id": "c5ce40bf-073c-49bb-977b-5933d6d36b71"
}
```

Updating Registration Tokens

Caution

In a mixed cluster version setups, *do not* reset the value of `cert-duration` specified during token creation for old nodes, this might lead to abnormal system behavior.

To update a registration token, run:

Syntax

```
ksctl clientmgmt tokens update --token-id <token-id> --cert-duration <validity-duration-of-certificate> --client-mgmt-profile-id <client-profile-id> --life-time <lifetime-of-token> --max-clients <maximum-number-of-client> --name-prefix <name-prefix>
```

Example Request

```
ksctl clientmgmt tokens update --token-id "669457e6-89cc-482d-8cb5-3b35f2b1a89c" --ca-id "950f1ef4-7b47-4cbb-9bfb-adee2f2c0283" --cert-duration 2 --life-time "5h" --max-clients 20 --client-mgmt-profile-id "c5ce40bf-073c-49bb-977b-5933d6d36b71"
```

Example Response


```
{
  "id": "669457e6-89cc-482d-8cb5-3b35f2b1a89c",
  "uri": "kylo:kylo:munshi:tokens:669457e6-89cc-482d-8cb5-3b35f2b1a89c",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-11-30T20:35:05.841579Z",
  "updatedAt": "2023-11-30T20:38:50.171348Z",
  "token": "4u9xQXsTknLcGT2byBrY4IJCOp40gWKIhEP7jxEUFJM27QDM1RgE26Fz43oSog1j",
  "valid_until": "2023-12-01T01:38:50.170988864Z",
  "max_clients": 20,
  "cert_duration": 2,
  "clients_registered": 0,
  "ca_id": "950f1ef4-7b47-4cbb-9bfb-adee2f2c0283",
  "name_prefix": "test_prefix",
  "label": null,
  "client_management_profile_id": "c5ce40bf-073c-49bb-977b-5933d6d36b71"
}
```

Getting List of Registration Tokens

To list all registration tokens, run;

Syntax

Example Request

```
ksctl clientmgmt tokens list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "ed307d73-695a-4c98-b1bd-a412d84b7b82",
      "uri": "kylo:kylo:munshi:tokens:ed307d73-695a-4c98-b1bd-a412d84b7b82",
      "account": "kylo:kylo:admin:accounts:kylo",
      "application": "ncryptify:gemalto:admin:apps:kylo",
      "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
      "createdAt": "2023-11-29T11:19:41.044238Z",
      "updatedAt": "2023-11-29T11:39:13.306853Z",
      "token": "UD21fsg4ZYJHGASOfWwJxPT6eqXFTLRgBBkDGxy5rrBcnGCs1uoM5MHluLZ65kW3",
      "valid_until": "0001-01-01T00:00:00Z",
      "max_clients": -1,
      "clients_registered": 5,
      "ca_id": "950f1ef4-7b47-4cbb-9bfb-adee2f2c0283",
      "label": null,
      "client_management_profile_id": "52d3041a-e958-4144-879e-eddbe6debc41"
    }
  ]
}
```

```
]
}
```

Deleting Registration Tokens

To delete a registration token, run:

Syntax

```
ksctl clientmgmt token delete --token-id <token-id>
```

Example Request

```
ksctl clientmgmt token delete --token-id "669457e6-89cc-482d-8cb5-3b35f2b1a89c"
```

Example Response

There will be no response if token is deleted successfully.

Getting the Fingerprint of the Server's "web" Interface Certificate

The fingerprint of the server's "web" interface certificate is unique. It is used when registering ProtectFile and CTE UserSpace clients with the CipherTrust Manager. A CipherTrust Manager administrator can provide you the fingerprint.

Fingerprint of the server's web interface certificate can be viewed on the GUI or the API playground.

On the API Playground

To get the fingerprint:

1. Acquire an authorization token.
2. In the left pane of the API playground, click **Client-Management/Tokens**.
3. Under **Client-Management/Tokens**, click **Web Certificate Fingerprint**. The **Web Certificate Fingerprint** section of the API playground is displayed in the right pane.
4. Click **GET**. View the fingerprint details in the response output.

The SHA256 and SHA512 fingerprints of the server's web interface certificate are displayed.

On the UI

To get the fingerprint:

1. Log on to the CipherTrust Manager as administrator.
2. In the left pane, click **Access Management > Registration Tokens**. The list of existing registration tokens with details such as their name, expiry, remaining usage count, and usage count is displayed.

3. On the right, under **Web Server Certificate Fingerprint**, the fingerprint is displayed. You may need to scroll down the page.

The UI displays the SHA256 fingerprint. To get the SHA512 fingerprint, use the API playground.

1. Copy the fingerprint.
2. Save the fingerprint. This fingerprint will be used when migrating clients.

Managing Client Profiles

A client profile represents the data that can be used to create or renew the client certificates. Client profiles allow the CipherTrust Manager administrators to manage the local or external CAs that issue certificates for clients associated with the given client profiles. As well, client profiles are required to issue registration tokens for KMIP, NAE, custom REST API, CTE, CTE Userspace, and CTE for Kubernetes clients.

This section provides instructions to create and manage client profiles on the CipherTrust Manager.

The following operations can be performed:

- Create/Get/Update/Delete client profiles
- List all client profiles

Creating Client Profiles

CipherTrust WebUI

1. In CipherTrust Manager, navigate to **Access Management > Add Client Profiles**.
2. Click **Add Client Profile**.
3. Enter **Profile Name** and **Description**.
4. Select **CA Type**: Local or External.
5. Select the respective Local or External CA in **Select <Local or External> CA**.
6. Enter the **Certificate Duration** for which the CA certificate will remain active.
7. (Optional) Expand the **CERTIFICATE SIGNING REQUEST OPTIONS** section and provide the following details:

Note

This section should be filled for non-CTE clients. The fields cannot be edited and are meant for new registrations, not for renewals.

Field	Description
Common Name	Enter common name of the client certificate.
Country	Enter country name.
State	Enter state name.
City	Enter city name.
Org Name	Enter organization name of the client certificate.
Org Unit	Enter organization unit of the client certificate.
UID	Enter the user identifier to be included in the subject name field.
Email	Enter the email.

8. Expand **CONFIDENTIAL COMPUTING** section and add the following details:

Note

This section should be filled only for CTE clients.

- Select **Attestation Authority Identifier**. Currently, only Intel Trust Authority (ITA) is supported.
- Select the **Attestation Type**. This is the technology that Attestation Authority uses to attest VMs on CTE agent.
- Select a non-admin connection in the **Attestation Connection** field.
- Select an admin connection in the **Admin Connection** field. By default, this field remains blank. Selecting a value in this field allows you to select policies in the **Policy Names** field.

Note

- Admin and non-admin type connections should belong to the same Attestation Authority.

- Select the policies in the **Policy Names** field. These policies are fetched from the Attestation Authority server. You can select multiple policies.
- Keep the default values for the remaining fields.
- (Optional) Click **Reset confidential computing form** to reset all the fields to the default values and input the information again.

9. Expand the **Groups** section to associate CipherTrust Manager **groups** with the client profile. This controls the operations the client is permitted to do. Click **+ Show All** to display all available CipherTrust Manager **groups** and a search bar for group names. Enable and disable checkboxes for desired groups to associate or disassociate the client profile.
10. Expand the **Labels** section.
 - To add a new label, enter a **Label Name** and optionally enter a **Label Value**. The allowed characters for both fields are letters, numbers, periods (.), hyphens (-), and underscores (_). **Label Name** also accepts forward slashes (/).
 - Click the **x** next to existing labels to delete them.
11. Click **Add Client Profile**.

After you have created a client profile, it can then issue **registration tokens**.

KSCTL CLI

To create a client profile, run:

Syntax

```
ksctl clientmgmt profiles create --ca_id <ca-identifier> --csr-params <csr-parameters> --groups <list of groups> --cert-duration <validity-duration-of-certificate>
```

- **ca_id** - ID of the trusted Certificate Authority that will be used to sign the client certificate during registration process.
- **csr-params** - the client certificate parameters.
- **groups** - list of the groups to which the client will be added while registration.
- **cert-duration** - duration (in days) for which the CipherTrust Manager client certificate is valid.

Note

During registration, the clients are added to the groups defined in the **groups** parameter as well as to the **All Clients** group by default.

Example Request

```
ksctl clientmgmt profiles create --ca_id 950f1ef4-7b47-4cbb-9bfb-adee2f2c0283 --csr_parameters "{\"csr_cn\": \"admin\"}" --groups "Key Admins,User Admins" --cert-duration 2
```

Example Response

```
{
  "id": "ee910055-b8fc-42c2-9c67-49bdbd0f67b2",
  "uri": "kylo:kylo:client-management:generic-client-profiles:ee910055-b8fc-42c2-9c67-49bdbd0f67b2",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-11-30T20:25:32.423768Z",
  "updatedAt": "2023-11-30T20:25:32.423768Z",
  "ca_id": "950f1ef4-7b47-4cbb-9bfb-adee2f2c0283",
  "csr_params": {
    "csr_cn": "admin"
  },
  "groups": [
    "Key Admins",
    "User Admins"
  ],
  "cert_duration": 2
}
```

After you have created a client profile, it can then issue [registration tokens](#).

Getting Client Profile Details

CipherTrust WebUI

To view client profile details:

1. In CipherTrust Manager, click **Access Management > Add Client Profile**.
2. Click the name of the profile you wish to view.

General details, Certificate Signing Request Options, Confidential Computing, Groups, and Labels are visible.

Detail	Category	Description	Editable
Profile Name	General details	The profile's autogenerated name	Yes
Description	General details	An optional, user-provided description	Yes
CA Type	General Details	Indicates whether the client profile is associated with a local or external certificate authority.	Yes
Select Local CA/Select External CA	General Details	Indicates the certificate authority associated with the client profile	Yes
Certificate Duration	General Details	Duration in days for which the client certificate should be valid	Yes
Common Name	Certificate Signing Request Options	Common name of the client certificate	Only in ksctl CLI and REST API

Detail	Category	Description	Editable
Country	Certificate Signing Request Options	Country name	Only in ksctl CLI and REST API
State	Certificate Signing Request Options	State name	Only in ksctl CLI and REST API
City	Certificate Signing Request Options	City name	Only in ksctl CLI and REST API
Org Name	Certificate Signing Request Options	Organization name of the client certificate	Only in ksctl CLI and REST API
Unit	Certificate Signing Request Options	Organization unit of the client certificate	Only in ksctl CLI and REST API
UID	Certificate Signing Request Options	User identifier attribute in the subject name field for a client or group of clients	Only in ksctl CLI and REST API
Email	Certificate Signing Request Options	E-mail address	Only in ksctl CLI and REST API
Attestation Authority Identifier	Confidential Computing	Applicable only to CTE clients. Authority or verifier to attest the trusted execution environments.	No, only one Attestation Authority Identifier is supported.
Cloud Provider	Confidential Computing	Applicable only to CTE clients. Valid values are "Azure" and "Google Cloud"	Yes
Attestation Connection	Confidential Computing	Applicable only to CTE clients. The CipherTrust Manager connection that communicates with the Attestation Authority.	Yes
Admin Connection	Confidential Computing	Applicable only to CTE clients. Optional admin-level CipherTrust Manager connection that communicates with the Attestation Authority, allowing CipherTrust Manager to fetch attestation policies.	Yes
Policy Type	Confidential Computing	Applicable only to CTE clients. Type of attestation policy.	No, only one Policy Type is supported.
Policy Names (Multiple selection)	Confidential Computing	Applicable only to CTE clients. Selected attestation policies	Yes
Groups	Groups	The CipherTrust Manager groups associated with the client profile. This	Yes

Detail	Category	Description	Editable
		controls the operations the client is permitted to do. By default, only selected groups associated with the client group are displayed. Click +Show All to display all available groups, and a search bar for group names. Click Show Selected to return to the default view	
Labels	Labels	Labels associated with the client profile, to help group resources. Existing labels are displayed in the format <code>:</code> . If there is no label value, the label displays as <code>:</code>	Yes

Refer to [Updating Client Profiles](#) for information on how to edit applicable client profile details.

KSCTL CLI

To get details of a client profile using id, run:

Syntax

```
ksctl clientmgmt profiles get --profile-id <client-profile-identifier>
```

Example Request

```
ksctl clientmgmt profiles get --profile-id ee910055-b8fc-42c2-9c67-49bdbd0f67b2
```

Example Response

```
{
  "id": "ee910055-b8fc-42c2-9c67-49bdbd0f67b2",
  "uri": "kylo:kylo:client-management:generic-client-profiles:ee910055-b8fc-42c2-9c67-49bdbd0f67b2",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-11-30T20:25:32.423768Z",
  "updatedAt": "2023-11-30T20:25:32.423768Z",
  "ca_id": "950f1ef4-7b47-4cbb-9bfb-adee2f2c0283",
  "csr_params": {
    "csr_cn": "admin"
  },
  "groups": [
    "Key Admins",
    "User Admins"
  ],
}
```



```
"cert_duration": 2
}
```

Updating Client Profiles

CipherTrust WebUI

1. In the CipherTrust Manager GUI, click **Access Management > Client Profiles**.
2. Click the ellipsis icon (...) corresponding to the client profile name you want to update.
3. Select **View/Edit**.
4. Update the values of **Profile Name**, **CA Type**, **Select Local CA/Select External CA**, **Description**, and **Certificate Duration** fields as needed.
5. Expand the **Groups** section. Click **+ Show All** to display all available CipherTrust Manager groups and a search bar for group names. Enable and disable checkboxes for desired groups to associate or disassociate the client profile.
6. Expand the **Labels** section.
 - Click the **x** next to existing labels to delete them.
 - To add a new label, enter a **Label Name** and optionally enter a **Label Value**. The allowed characters for both fields are letters, numbers, periods (.), hyphens (-), and underscores (_). **Label Name** also accepts forward slashes (/).
7. (For CTE clients) Expand the **Confidential Computing** section and modify the field values as needed.

Note

The Admin Connection field appears blank, but it actually holds the value that user entered during the creation of the client profile. You can select a different value and update this field, as required.

8. Click the **Update** button.

KSCTL CLI

To update a client profile, run:

Syntax

```
ksctl clientmgmt profiles update --profile-id <profile-id> --ca_id <ca-identifie
r> --csr-params <csr-parameters> --groups <list of groups> --cert-duration <valid
```

```
ity-duration-of-certificate>
```

Note

- Update of the list of groups applies only to newly registered clients. It will not impact the behavior of the existing clients.
- The updated list of groups will overwrite the existing list of groups in the profile.

Example Request

```
ksctl clientmgmt profiles update --profile-id ee910055-b8fc-42c2-9c67-49bdbd0f67b2 --cert-duration 6 --groups "Connection Admins,Client Admins"
```

The value defined in the `groups` parameter updates the list of the groups to which the client is added when registered.

Example Response

```
{
  "id": "ee910055-b8fc-42c2-9c67-49bdbd0f67b2",
  "uri": "kylo:kylo:client-management:generic-client-profiles:ee910055-b8fc-42c2-9c67-49bdbd0f67b2",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-11-30T20:25:32.423768Z",
  "updatedAt": "2023-11-30T20:29:57.156777Z",
  "ca_id": "950f1ef4-7b47-4cbb-9bfb-adee2f2c0283",
  "csr_params": {
    "csr_cn": "admin"
  },
  "groups": [
    "Connection Admins",
    "Client Admins"
  ],
  "cert_duration": 6
}
```

Getting List of Client Profiles

To list all client profiles, run;

Syntax

Example Request

```
ksctl clientmgmt profiles list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "ab1cb81a-ee07-4c1d-963e-24964c305271",
      "uri": "kylo:kylo:client-management:generic-client-profiles:ab1cb81a-ee07-4c1d-963e-24964c305271",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2025-07-28T06:17:57.071981Z",
      "updatedAt": "2025-07-28T06:22:43.318596Z",
      "name": "cmp-ab1cb81a-ee07-4c1d-963e-24964c305271",
      "ca_id": "950f1ef4-7b47-4cbb-9bfb-adee2f2c0283",
      "csr_params": {
        "csr_cn": "admin"
      },
      "groups": [
        "Connection Admins",
        "Client Admins"
      ],
      "cert_duration": 6
    }
  ]
}
```

Deleting Client Profiles

Note

- This operation is not available on the CipherTrust WebUI.
- Take care to delete any associated KMIP profiles or clients. They are not deleted automatically.

To delete a client profile, run:

Syntax

```
ksctl clientmgmt profiles delete --profile-id <profile-id>
```

Example Request

```
ksctl clientmgmt profiles delete --profile-id ee910055-b8fc-42c2-9c67-49bdbd0f67b2
```

Example Response

There will be no response if profile is deleted successfully.

Renewing Client Certificates

Note

During client renewal process, if a certificate is presented, it undergoes a revocation check using the Certificate Revocation List (CRL). If the certificate is determined to be revoked, the renewal process cannot proceed. An audit record is generated to notify the same.
This applies to all client types and is relevant for both automatic and manual renewal processes.

The CipherTrust Manager allows you to renew the certificate of a registered CipherTrust Manager client. The CipherTrust Manager provides the following renewal methods:

- **Admin's Renew Route** - A member of the `Client Admins` or `admin` group can renew the certificate for a client.
- **Self-renew Route** - An authenticated client that is a part of `All Clients` group can renew its own client certificate.

Note

- Self renewal is only supported for certificates signed by a Local CA of the CipherTrust Manager.
- The `grant_type` of the client authentication should be `client_credentials`.

After the successful renewal, the renewed certificate and the key (optional) will be shared in the response.

The authentication of the renewed 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.
- After the renewal of client certificates, the CipherTrust Manager continues to allow old client certificates unless they are revoked or expired.
- A CipherTrust Manager client certificate cannot be renewed if it is revoked.

The following table lists the parameters that are required to renew client certificates:

Parameter	Description
cert_duration	Duration (in days) for which the CipherTrust Manager client certificate is valid. The default duration is 730 days. It is recommended not to use a duration longer than that. The certificate duration shouldn't be more than the CA duration. However, if the certificate duration exceeds the CA duration, the certificate duration is automatically set to CA's duration. For admin renew route, if <code>cert_duration</code> is available in both the request and in the profile attached to the client, the value specified in the request is used. Otherwise, its value is set to the system's default value, that is, 730 days. For self-renew route, the <code>cert-duration</code> parameter is deprecated. Any value provided in this parameter will be ignored. If 'cert_duration' is provided in the profile, the value specified in the profile is used. Otherwise, the server default value will be used.
ca_id	ID of the trusted Certificate Authority to be used for the certificate renewal.
csr	CSR to be signed by one of the Local CAs (the CA which is the issuer of the current client certificate) of the CipherTrust Manager.
do_not_modify_subject_dn	Specifies if <code>Subject DN</code> in the CSR is allowed to be modified or not. If this flag is set to <code>true</code>, then <code>Subject DN</code> must be unique across all the CipherTrust Manager clients, otherwise renewal will not be allowed. The default value is <code>false</code>. For details, refer to the section Renew Local CA Client Certificates. Note Email ID is also part of subject DN of the certificate. An email ID mismatch can result in subject DN mismatch.
ext_cert	New client certificate signed by an external CA to renew an existing CipherTrust Manager client.
subject_dn_field_to_modify	This field is used to make <code>Subject DN</code> unique so that the client can be identified uniquely. The default value is <code>UID</code> . For details, refer to the section Renew Local CA Client Certificates .
subject_dn_template	CSR parameters used to generate a CSR for client certificate.

Parameter	Description
key_gen_params	Parameters (algorithm, size, private_key_bytes, and so on) are used to generate and optionally encrypt the private key.

Renew Local CA Client Certificates

A client certificate can be renewed using one of the following methods:

- **Client CSR**

You can pass the CSR with `do_not_modify_subject_dn` flag. This flag specifies if the `Subject DN` in the CSR is allowed to be modified by the CipherTrust Manager as part of renewal or not.

If `do_not_modify_subject_dn` is set to `false`, the server assigns `client_id` to `Subject DN` of the field decided by the `subject_dn_field_to_modify` parameter.

For example:

```
"/CN=admin/UID=1a26fd25-0c28-4afa-b392-a4fa970604a1"
```

This ensures that `Subject DN` is unique for each renewed client.

However, if `do_not_modify_subject_dn` is set to `true`, the client does not permit the server to modify `Subject DN` in the CSR. In this case, the client needs to provide a CSR with `Subject DN` which is unique on the CipherTrust Manager.

Note

It is highly recommended to set the `do_not_modify_subject_dn` field to `false` unless there are constraints preventing modifications in the CSR.

- **Server CSR**

The CSR parameters (`subject_dn_template`) and key generation parameters (`key_gen_params`) can be passed to generate and optionally encrypt the private key.

The server assigns `client_id` to `Subject DN` of the field decided by the `subject_dn_field_to_modify` parameter. This ensures that `Subject DN` is unique for each renewed client.

- **Without Client and Server CSRs**

When the client CSR and server CSR parameters are not provided, the renewal parameters are taken from an internal client profile used at the time of registration. If no associated profile is found, the renewal fails.

The server assigns `client_id` to Subject DN of the field decided by the `subject_dn_field_to_modify` parameter. This ensures that Subject DN is unique for each renewed client.

In case of renewing a client that is issued by a local CA, you can specify the certificate duration. The default value (duration) is 730 days.

Following are the possible values of the `subject_dn_field_to_modify` field:

- `UID (userid)`
- `CN (commonName)`
- `SN (serialNumber)`
- `DNQ (dnQualifier)`
- `OU (organizationalUnit)`

If none of the fields is selected, by default the `UID` is used to modify the Subject DN.

Warning

If any of these fields (except `OU`) is selected, the original field values will be overridden by the CipherTrust Manager.

If `OU` is selected, the `OU` will be appended to Subject DN with other attributes.

Note

It is recommended to select a field that is not already a part of CSR or CSR parameters (that is, select a not-in-use field).

Renew External CA Client Certificates

When renewing a client certificate that is issued by an external CA, you need to provide the new certificate signed by the external CA in the request. The renewed certificate must have a unique Subject DN for each client. Also, Subject DN of the new certificate should match Subject DN of the old client's certificate.

Renew Client Certificates using ksctl

Note

If `cert_duration` is available in both the request and in the profile attached to the client, the value specified in the request is used. Otherwise, its value is set to the system's default value, that is, 730 days.

To renew client certificate, run:

Syntax

```
ksctl clientmgmt clients renew --client-id <client-id> --cert-duration <Duration-in-days> --ext-cert <Certificate-file-signed-by-an-external-CA> --clientcsr <CSR-signed-by-local-CA> --subject-dn-field-to-modify <Subject-Distinguished-Name> --do-not-modify-subject-dn --alg <Algorithm> --cn <Common Name> --dns <Subject Alternative Names> --email <E-mail-Addresses> --enc-alg <Private Key Encryption Algorithm> --ips <IP-Addresses> --names <{O:organization, OU:organization unit, C:country, ST:state/province, L:location}> --pass <Password to PEM-encrypt the private key> --private-key-bytes <Private key bytes of the key> --size <Key size>
```

Example Request 1 for Local CA

```
ksctl clientmgmt clients renew --client-id 1a6b1e6c-bd80-4936-9db3-f85d1d547b01 --cert-duration 100 --alg ecdsa --dns *.thalesgroup.com --email contact@thalesgroup.com --enc-alg AES256 --ips 1.1.1.1 --names {O:Thales Group,OU:RnD,C:US,ST:MD,L:Belcamp} --pass KeySecure_1 --private-key-bytes -----BEGIN EC PRIVATE KEY-----\nMHcCAQEEIEj1liesPsLdk0tM7Jv87sruegOPdmji9SY3s3ncdckxqoAoGCCqGSM49\nAwEHoUQDQgAEL8cvuduRZs6e/vsttMlhi9HxV+0FzhCg/zHUmXNmyH5KlmQgoaq1\nvVfwnHqQk79lf+55WSLD7uUwaxhYwGHIapw==\n-----END EC PRIVATE KEY----- --size 256
```

Example Response 1

```
{
  "client_id": "1a6b1e6c-bd80-4936-9db3-f85d1d547b01",
  "key": "-----BEGIN RSA PRIVATE KEY-----\nMIIEowIBAAKCAQEAnyfmh0/39UJAPAJooSLBbwFila+OGBog1zVJ3nmtx5kvnUJQ\n\nnb4i7U+RKmvdqXnDcl+cVy+446jqwQQ4ixRKidmG0dexD3ErZ9JSMy2tbQsmd/Z4G\n\nn5SQgSzwWDHNknzJ7+MD022t7JNhRq9E4gNs7Wh26kvUh0lwPn5W9lGPAR1YpClvW\n\nnOyPYBVTlwSQZpgShzXMSTrVle9OqzdKg/Otjb8QvXFtxwstjB4GI+sjvo35+XqcG\n\nnCRnea+gcsIigAXkKZP+CvCrpC3rF9X9Bt8iYsCEyo/X0SZc1eAEE+3Mjc/zSMQ4h\n\nnyJwIQQ2su0+BPDINYVLlr6dtSN3eL3+KelDd7wIDAQABAoIBAF1Jt/gsSXz6vkfu\n\nnaYG5q4PzXeLHU1SRPN08IPZEgDX0dlJJInidvp+nuosm5rRALyx6iO4lZjqM9+7J\n\nn6Ft3Vr7o+iv8trITNboEsJozbSSypOaZkBUTndI7f/kAUkR2BPCDsnmagG3SVdzO\n\nnH753wKxKDKhDAvYYfBfgX2TxKy0kSMfaG+H/WrAv6wQEHilGRZDy8XYw/z4IlwJE\n\n\nnexAEDfk/mn/AUeITA8UYAJwJMGnY2Vp9+BlqNjzWpCnnamzWDf6I2bX7+AVxZIXs\n\n\nnupLX5uXTcx2Q/RqaAoNjK9/UllVnMwDXDOtavXkmAowYN68tn2vInuEtf3BGHhfT\n\n\nn4OVtRGECgYEA0n0544A0cWEhvDtelyGYTX8jOhaPuNkCr8bKetulKTlIcR4X1Uqa\n\n\nnJgbRt3H8Lpnd2jQ5iD/xfnuwHzr1IAOB64GuJgIuMhSzxtjqZa4c5IMS6EFvWIr\n\n\nk\n\nnmnQWaTqWCrc6M9YAmFGjcedSmMgRKJvFpWQqc4ZDrV8M7+JWxROJlmcCgYEAwZoQ\n\n\nnv+HH/3f/o76+AgNF12XOfTENqJxhwsaNVN6P3/GXX+9rtKh2S02sb16yTq+Llm2w\n\n\nnDMBHZHaJhumF+5JtkLWON+hfvQDsmKxtZ2c07303TMOOcOjxB1b+swgTG95Xv4et\n\n\nnHt0+LE9Wh6rTKPs8JzQBAl3VrWbQQxIkFoXwNzkCgYB0jA5YBST6eTY7jg2ZyksU\n\n\nnL+KMWs5BKj0op0ie/Wm4aLbLnkTIpEX15TTjCMcF78RDA
```



```

UUTRYx68G/qeDyV77t\nc6XD+tUmevyM1LhKk0AA9Pw7q3FEcbbkKoH2yLEqW6ip6EvwDtYbUGYXokSO
XqX4\nUB6KV8LHEqqRdQfJBS0A5QKBgDRpTwuDrI+E/vIHW00V9CLOxt3hf78bxuOB+RT2\nw6BuuPpBy
Li5unycZlGp7L+cin+I51YtX6Z06LpbUehPNdYtMHk2MUjD3U1ftKZ/\nrDcBgQ0JC/DJPyrch/xTvUiY
Is05H8bxQjmgqN5FgrDfguNiErVfR0+QO+VMbfSj\nAesRAoGBAIX03Vls5HiIZSpIjE2Ab9ojrgnbWPE
bIowOcjhBzKZM7zcLfF6vibO2\n5mfKp9JLZ+YoVggJispBBvnBXhzSSgcePyrVfp1DASQaGs8LtAb42o
r/7KZvnIzd\nNehsZhUOPD4Lnv6Wh5pzorjVbIbm315u01PKxD2OEB7k3G5xMRn+\n-----END RSA PR
IVATE KEY-----\n",

```

```

"csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIICczCAVSAQAwEDEOMAwGA1UEAxMF
dGFuZWwoggEiMA0GCSqGSIb3DQEBAQUA\nA4IBDwAwggEKAoIBAQCfJ+aHT/f1QkA8AmihIsFvAWKUD44
YGgbXNUneea3HmS+d\nQlBviLtT5Eqa92pecNyX5xXL7jjQorBBDiLFEqJ2YbR17EPcStn01IzLaItCyZ
39\nnngblJCBLPBYMc2SfMnv4wOjba3sk2FGr0TiA2ztaHbqS9SHSXA+flb2UY8BHVikK\nw9Y7I9gFVOX
BJBmmBKHNcxJOtWUT06rN0qd862Nvx9C9cW3HCy2MHgYj6yO+jfn5e\nnpwYJGd5r6BywiKABeQpk/4K8Ku
kLesX1f0G3yJiWITKj9fRjLzV4AQ7cyNz/Nix\nDiHiNAhBDay7T4E8Mg1hUsivp21I3d4vf4p6UN3vA
gmBAAGGhJAcBgkqhkiG9w0B\nCQ4xDzANMASGA1UdEQQEEMAKBADANBgkqhkiG9w0BAQsFAAOCAQEAEL6
Si44XlK3\neelk7wXthn+I+Pa0BIjHx90koBTW5RKLAAALsaisYjjAjAPJlfisK8aQMVS3imD\nnucSGp
CRgqhY1oOPOALTp/mwWTsDuranFd/hfNQc5DppqGGOXj5NHTC4si7nYPuRL\nhHqgn/qHN/gCgh984v4G0
7suJsxQ1zA5beeyxRORE9/Umspkyjss/olw+c/ekbc+G\nnr6+Fp41JwOhsH39kT/eLgF2LNDXDFWbuF2Q
13Y7jRVfT1LMRI0xuy00AncIMayMj\nnoIyQNbnFTYxSzsRhgtZ/85tXtoaLva2h/dy+21f8ufiAxZTUQW
PN914DgozGw4yN\nl9qKd9SKTw==\n-----END CERTIFICATE REQUEST-----\n",

```

```

"cert": "-----BEGIN CERTIFICATE-----\nMIIE5zCCAs+gAwIBAgIRAPPwaSrfpbwXqCG3dSB
AA6MwDQYJKoZIhvcNAQELBQAw\nWjELMAkGA1UEBhMCVVMxChAJBgNVBAGTAk1EMRAwDgYDVQQHEwZCW
xjYW1wMRAw\nDgYDVQQKEwZHZW1hbHRvMR0wGAYDVQQDEwZFLXZlTZW1cmUgUm9vdCBDQTAeFw0y\nnMTE
yMDExMDQ1NTZaFw0yNDA4MjcxMDQ1NTZaMEYxZjAMBgNVBAMTBXRhbmVqMTQw\nnMgYKCCZImiZPyLQGBAR
MkMWE2YjFlNmMtYmQ4MCO0TM2LTlkYjMtZjg1ZDFkNTQ3\nnYjAAMIIBIjANBgkqhkiG9w0BAQEFAAOCA
Q8AMIIBCgKCAQEAAnyfmh0/39UJAPAJ\nnoSLBbwFiLA+OGB0G1zVJ3nmtx5kvnUJQb4i7U+RKmvdqXnDc
l+cVy+446jqwQQ4i\nnXRKIdmG0dexD3ErZ9JSMY2tbQsmd/Z4G5SQgSzwWDHNknzJ7+MDo22t7JNhRq9E
4\nngNs7Wh26kvUh0lwPn5W9lGPAR1YpC1vW0yPYBVT1wSQZpgShzXMSTRvLE9OqzdKg\nn/Otjb8QvXFtx
wstjB4GI+sJvo35+XqGCRnea+gcsIigAXkKZP+CvCpC3rF9X9B\nnt8iYsCEyo/X0SZc1eAEE+3Mjc/z
SMQ4hyJwIQQ2su0+BPDINYVLr6dtSN3eL3+K\nnelDd7wIDAQAB04G7MIG4MA4GA1UdDwEB/wQEAwIDiD
ATBgNVHSUEDDAKBggrBgEF\nnBQCDAjAMBgNVHRMBAf8EAjAAMB8GA1UdIwQYMBaAFFjNnvvBFxGnHa+Ki
XmdOsib\nnf+z6MASGA1UdEQQEEMAKBADBVbGnVHR8ETjBMMEqgSKBGhkRodHRWoi8va2V5c2Vj\nndXJlLm
xvY2FsL2NybmVOWZmZWJlMTgtYjQxMi0MDYzLTg0NTMtMzYzNzBiZjQ4\nnZTg2LmNybdANBgkqhkiG9
w0BAQsFAAOCAQEAzSV/H1VR7KJ80DG3obKmGbIbs+0y\nnXkt6Js7BFjo58GlyguXtQYX+UTXr6OA5MiPW
YYQSY4RbXWX79lqmo1XQLuIpPN7R\nntpDFk3rn9WYWF7fKWFLLSMXPdz397cZJ5WHP93EootZyeDiSIqHm
tTpSW+cJ1l/j7\nnumhY7e58nJ/Uml5bHc5npLeXNcEbzbwDcMVtOZ7P8nATex04nXgQZemZ+VHRbVhn\nn
apKG5rXRAt10kg6ntqf8jfm4np9RN30gAhQaPIn+r5Cbvv/LbXkcwY1aY5KGv1KW\nn0rbB17CbeMQ1khh
KW518nXbxdNj1ZgVPCi+KGcYdg5VOg4zCCS7dWPHYfNmEkJHu\nnVhz7xeVwFLlXrdW7H6R4QhmBgPrzIw
qtnfPzbs+gurtzfdGMrlCFF+49KASCxMev\nncVSZzROHcpkQ0PLAUhTYCOib/ZJfi9u57HYy0d0pfnP44
0qr2whsVlwJvEN9GS/D\nnwEnumdN3VNTXE4NZ1XG/tFK4kiUXuM8vGMqCvJr+10yxMmZdX6Qgi6z6VebG
IJ4w\nncRt83FDou5qQmH7krjXV8cXyGtIebBPwXcCKO3iMZNxpbs3P9LQx4yqwShQB3DcI\nnM7MgMSgcZ
HFTqrQJ3G3+vPpX2dzBXEWAC35VKjmvDQaSPk/woyjL4mVn1IDyrIpn\nnnTNcbt1L9MVxlXw=\n-----E
ND CERTIFICATE-----\n",

```

```

"ca_cert": "-----BEGIN CERTIFICATE-----\nMIIGJDCBAygAwIBAgIQDOK/DLL6BXSyFbnl
yJAJDDANBgkqhkiG9w0BAQsFADBa\nnMQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUQxEDAOBgNVBACTB0J
lbGNhbXAXEDAO\nnBgNVBAoTB0dlbWFSdG8xGjAYBgNVBAMTEutleVNlY3VyZSBSb290IENBMB4XDTIx\nn
MTEzMDA1NDY0Ml0xDTI0MDMwNDA1NDY0Ml0wXDELMAkGA1UEBhMCVVMxChAJBgNV\nnBAGTAk1EMRAwDgYDVQQHEwZCW
xjYW1wMRAwDgYDVQQKEwZHZW1hbHRvMR0wGAYDVQQDEwZFLXZlTZW1cmUgUm9vdCBDQTAeFw0y
nMTE
yMDExMDQ1NTZaFw0yNDA4MjcxMDQ1NTZaMEYxZjAMBgNVBAMTBXRhbmVqMTQw\nnMgYKCCZImiZPyLQGBAR
MkMWE2YjFlNmMtYmQ4MCO0TM2LTlkYjMtZjg1ZDFkNTQ3\nnYjAAMIIBIjANBgkqhkiG9w0BAQEFAAOCAg8A\nnMIICGKCAQEA3zVqkxglhUkeYSGgzYM7v0vpjiERp6EPv
3vdJ+hCVgIEqj1V5vow\nd1X/mmEXBpZGDTZLnM50106Mfj8jzp1TGooNe5/hPHLJEPUnLGS2FFiSYZ4K
CZb7\nnZv84iRgEkl63/OU47QntmtwugPJg4SAQkBe6ekr7WrJr53Zlfs0N+c6ciKBA9mQCL9\nnos1av2Q3
f+WcFZrpbWEhrAyv1SEoletbjMtPQ92XGUjJuRcnp2rFuh1AR9NHGv8\nnQX11oLo2vOR7GuUVScI8ZNH
NXYkfoL5H6OeTh9es7clrm8lg3M0Ssgy9ph6F4/ej\nnHvIqHBZK7SdBjUuQz3rSD6mtWxOsnog37tvUsc
FsiYOMVrCmHS5xykISHHQhs5W\nnv3352Z/IcUN9BjvH/vVoQbMBSv4BZjy48GsSwe5lmCgTc/Z0jzuBY
LzuhAd02Y+2\nnxi/bhQvAFALmoXepbbtnGmNoLDFI1CpfDTfGPlWrHShg9yJQjW7DzoWJZrasSfb9m\nnhf
qriDoueT6PNEfFAE77CIdILPELWSsFmsT56FlUQj1jveATjvou0nJiAHDdbkY\nn
nTN1TtINR6w7uxBfiHdPIrtVqyNGT9qyHJXXa9axBY2Sbo/pDGvc0X0R2IMtjwU08\nnZ5FMqATI9H7P5bXGPTs5IXYAQTqTSSmn
91rB2LQZIJR6cvUf6R7eD3sCAwEAAaOB\nn4zCB4DAOBgNVHQ8BAf8EBAMCA4gwEwYDVR01BAwwCgYIKwY
BBQUHAAwEwDAYDVR0T\nnAQH/BAIwADAFBgNVHSMEGDAWgBRYzZ77wRcRpx2viol5nTrIm3/s+jAzBgNVHR
EE\nnLDAqghN3ZW1ua2V5c2VjdxJlLmxvY2FsgrNzdxBwb3J0QGdlbWFSdG8uY29tMFUG\nnA1UdHwROMeW

```

```
wSqBIOeAeGRGh0dHA6Ly9rZXlZWN1cmUubG9jYWwvY3Jscy85ZmZl\nYmUxOC1iNDEyLTQwNjMtODQ1My
0zNjM3MGJmNDhlODYuY3JsMA0GCSqGSIb3DQEBA\nCwUAA4ICAQCYi0/YtLHBnNIBj+IRGrGo3h1mFk+La
KR98D5dSBZd1Czj3/SMF0RQ\nnqDWi4KLu0Ro+8R+ok+slp19ZAV5wo6w1UMZM/IMwFND8VE4SrO1X4wjP
EdLmRWnO\nnjzHXUHNzEOozzwrSJOQP1wReY6DtZ0V/i7zS1KerctN6ZwQAIBM3ZQ3T+TvN4D8e\nncYfkP
Tcgo/HEBDjO9DhRxTE3wmUleKDpY4Xv9iPMfELnm36Hm47Ob0TxdK+KD+G1\nnNm+z9yWgJZ3POaNHROF4
itefEwjIqmTkf8BGNTe5e6+pRzHwr1An3AtkydufPnuX\nnQzveCajhQi5edQpUad72Lds9vgMcCUseeiU
F0dcjq9Kj5mJUFUnfIPuRPY36oDph\nnUJlT8P5Ttd7wngMoxeGpz+qrJxkrZ7AIFdquA9GM8VLMFdi9AL
IImnyNg4OPwjaK\nnJfCDurIZ0d8Ba4cYqqiHLbuD0/a+/xYGwRvoQwCOM70J6oeo1O1JVOzJ1dv9PEI\
nlJmQmT2XBH06wNV58xFQ3VPhtZM6JuvAQGd/o908qmOuDVhMcmMMu0FsJrCijxTc\nnjj9Q3XpwwZKAY
b+NduyIbQ2DqcFBMisV4tgyhdBGHTtD7DIL0y+Rx0fU9jwz0hny\nnNyBpD4LvK1Y4ObWnViiUAEQPzAE
z3LWz/U2J9YMZpI1yvyfB+Xy8g==\n-----END CERTIFICATE-----\n-----BEGIN CERTIFICAT
E-----\nMIIEFoTCCA4mgAwIBAgIRAPx8MV+j0NqVPb2EhQpI3IQwDQYJKoZIhvcNAQELBQAw\nnWjELMAk
GA1UEBhMCVVMxZCZAJBgNVBAGTAk1EMRAwDgYDVQQHEwdCZWxjYyYw1wMRAw\nnDgYDVQQKEwdHZW1hbHRvMR
owGAYDVQQDExF1ZlZWN1cmUgUm9vdCBDQTAeFw0y\nnMTEzMzAwNTQ2MzBaFw0zMTExMjgwNTQ2MzBaM
FoxCzAJBgNVBAYTA1VTMQswCQYD\nnVQqIEwJNRDEQMA4GA1UEBxMHQmVsY2FtcDEQMA4GA1UEChMHR2Vt
YWx0bzEaMBGg\nnA1UEAxMRS2V5U2VjdXJlIFJvb3QgQ0EwgGIIeA0GCSqGSIb3DQEBAQUAA4ICDwAw\nng
gIKAoICAQDb9AihTQfx/QoOuhyOEVmAfXF0W0EQhNAPGIazTdpHZ1fkpe9qqZAR\nneHPC5ME9no6N7Kqa
ggovHzNyYU/TA6tFhduS91bjI0W2aS4wAldGRLA7LIfcYnV8\nnXuBp0x2otyjf+tJ1UAfmZyWkAmlJkfb
cxHwXtBS0Sv676qd/DnyqxS4Spj4GyHTl\nnTJM1Q1utHjV6BiY2AcUAlKYWC4R7sKkf2dN8Ce3eyiI9RU
6UFijcnCicuWI9W3MS\nnX2E33RFgJy/etN3dBWxs9+xFYgAbc11ORZU1Im4xbhwP94hhXBbfIoaJHsDe
tL7\nnCoP2GXPvODXUntJR4iQnN4ITlwEK62hGfjGGdq2a+1Xqz25bB0Qzj2FHSeCeeAuQ\nnLWZKC/3Ccv
kTxBVSzzv66rzMKqmY9gSs8tenFbSuK58tGUxNR6HoklCBi/vsIokN\nnfHLBvbA9BYAHA4mLmoBBPoj2u
oGJphkm1J5J0YH4vvyLxG425Z8NP0WV1MdvL+OG\nn5NPJg/+kiANpcy/RDMMYsBY6dCbxdm7ojAtu9JM9
5/Z3V6VjQFa1NMCKP6gCzRVx\nnAoUQ9cKNUJHCIN+N0eSbBgM8/uSC1F/y5CiVLDEV3zKDxq/NZKPkUS3
YGime7c14\nnw3S39bi7uUr7NIdalPMzMrDtu4xNJY20vByJG2UeQTf2KH7ve4WFRQIDAQABo2Iw\nnYDAO
BgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUWM2e\nn+8EXEacdr4qJeZ06yJ
t/7PowHgYDVR0RBBcwFYETc3VwcG9ydeBnZW1hbHRvLmNv\nnbTANBgkqhkiG9w0BAQsFAAOCAGFAF4ZSN
iC4SRjiialyo8avfDVoXcqmFzJVE2Ym\nn9rPh8oAG44MHu3k5tnt1U82cnv2bLRSSdci+SRFb9CAunWPG
168D2D8LmUubjVBT\nnC7XxwTmaxXKxOEwdBk1VXjszw8kS3F17BguJ+9ZbobwphuszBwGWGqnT05QOTDd
M\nn/f/6VsJ5wG5mz+xBjiFqOT9DIzBpXb+TBOVnMa14Ie54BLOmhSo6y2HTfO8EYtG9\nn+FQEq3udXcuK
fhHGa563Fy1uiLPvD0xuYeJxmWCoTqWyRla06aYUG1LAksRLlJHD\nnwtHOKExJBVTm46ITVZqiQHweoPV
YnHLZQ6B+tF2EOblrURGZ3DcZJ9Kr6/xElyz2\nnvmjeF0iBMmVr83LdDCFo75TVquY1Y2+iMq4fpd5eh2
0vhvj5DMc9STDxNNAYiu31\nn2kDnGXJMSrZaCqb7SFw/X5Lh0zImHZUoaqMxQz1zH19xGfIGWXosX7X4M
KZhWHBK\nnM2ZNGIgh9Kub8bdEKDP7SID/Mph7P3eDkAX8auHTRPQEM+iyTkWRx6ohvjpjgI+S\nnawbGlR
FwX5BAk+epiWVLimr644aLyAZh8tf5rT4EcbqtWeQQwpJZDhmd+ykk42Kn\nnnDRcs5LpMNPX5IqsdZ0D8
dPhhpShqheG+C7pCE6SgEZM8LtwHprKp1XjkIK9g5Yg\nnRFAuSN8=\n-----END CERTIFICATE-----
\n"
}
```

Example Request 2 for Local CA

```
ksctl clientmgmt clients renew --client-id 400a1593-c80c-48cb-86c6-466c1f0b7664
--cert-duration 100
```

Example Response

```
{
  "client_id": "400a1593-c80c-48cb-86c6-466c1f0b7664",
  "key": "-----BEGIN RSA PRIVATE KEY-----\nMIIEPqAIBAAKCAQEA6OYYStxJNTZPqJItHuiV
vGM2ruKrMQVuhzPGn321QTJroKrA\n4gjoI4jEig7i3f4MGHzaVfRRKm7+ciRJniAUyCivohtFXe3DEyK
EzspftaokejAy\nn+8NkTZOEsDDpWRujf7VXZTF2be3MtOo1ie/dyuCwajmVL7t/DQU9C+I/gTHGAMz7\nn
x4vAlfiolxNEMcDwgKcFOCoT4surirdZNGZpzI6h1QVovcfsGMVotiMsPL7a3svc\nnS6EgrBJkYTSYuwy
YGJ6CSwMm5Cyhmflbkt4lsVAhHJGTZeGwzrGW0Ku0tFDSAYgF\nnHrKpJw6QqfrhY2uiKfG2X41ZYMV4A4
v7DMPYSQIDAQABaoIBAQDfSTqhCvFlatz8\nnUKJrQutlqf7UjWumfaNySX0YyETkLN/JjYmfysGu+jOf
}
```



```

hEPP\nHxbduQQo61spiRwRnwpyeGhAGjIzzFKB9T0cLSLusuYWlG6LexjOe0+GqsUtlFdJ\nniaWBii3AE
RALxqQKMhnn0drjTfU1Xxa+h2Oxji8L+32MHgXZT8wKPR3hCrg1Mll+\nrrrHg+bA8Q1kYE5GhKpijRGiK
8PmB085CWStJSTwDflb9ELbjQFAYpX7jgt3ueDj9\nmMDSL5QHET79C8e6g9PCbi2Je8PMro1DADX9CcK
pJhxZ21/bp8a7i9bpmde07CG\nfoszxKP0c9KjVzJy3S8qBcsSoOg9L1eZG0/gQpTRMqa3WUJ1CMxbXW
qeAAOVr3o\nnrz2Wtp4S5d0sFwnMloq7Fq417WlpuFRqdzDyeXQkiUKli77cIVKUKn+wHNuR81aP\nnpip
ph71rnGKV5rY/+c5ZfqQVDsGqc6UKLy/5If+Yo9u41bQdyfczdGZ7EVdeRIhe\nnHWYem4VBSb7HIPJu1S
sQ73rGkXh1Dx1CFEjcmvOwmUCUsLbBi7d65n0J+VeGwZgi\nnho/ZTne1v6XjACipJ+eLXe6XsHWtE8BWA
uY/j0gFFSsm0ktmCXyZsRC0CAwEAAoAB\nnzjCByzAOBgNVHQ8BAf8EBAMCA4gwEwYDVR01BAAwCgYIKwYB
BQUHAWewDAYDVR0T\nAQH/BAIwADAfBgNVHSMEGDAwGtDoXJ+X/v5BV+01fJsQ4ESkZERCDAAeBgNVHRE
E\nnFzAVgRNzdXBwb3J0QGdlbWFSdG8uY29tMFUGA1UdHwROMEwwSqBIOEaGRGH0dHA6\nnLy9rZXlZWN1
cmUubG9jYWwvY3Jscy8wNTQ5N2Y2Ny1jZDM4LTRiYWQtYjUxMS01\nnN2RlM2E4MzQwZGEuY3J5MA0GCSq
GSIB3DQEBECwUAA4ICAQCGI568+o6bcESMM73n\nn18N5Fuj4pKsnuNG1FeUvIBE/5Y+IwvMBasM5PMznp
8u+ZmWwN1Cx6kZsRk39UxG\nnh0j9RPMOMkCND0UoU+Kr573vQhXRicCyUprsb+Gbk9uWbmLbEHNhDulP
Yef8B8M\nnZiMrVa7ZGbr4Sn95KDonN1LaQzwc4pNNgKV9RY7b7yGuWqTSzWnfZ+7PE702ybi/\nNm/CiPm
KiDpI/zBFqB+8YXnoVuzK65DkKcuCf3VaDLdJDklOgt5OeAqXMnJLP6Aof\nnb1skPBzIY8SdPrWduOWE4
lmqkIoG/Z24if/Gz2vzq/fFOpebp44arjVN+N6psmKU\nnGndaJRItMaYjka3Vj90dZWbUEIiewQFqtHNm
Ovm0aU+5yFglH5cZ7XEIjDOVanxn\nn6hn8H8qp9kXVzNe3NdU2Mqtrejcal69PEzu206RPxhuk13YhefW
FelN8HqHZQqPI\nne7veDVKYwgGSXYWdhBeKbkfMd8xcQAH3SmZklRlOwKr0NiJ6ZVP03E99dUtvwiWm\nn
5ZMT1e8sY9sQfPk9sKtTaYhNFaD4Au8JZSwipW47Ns2qW4AdgKa+qvYj7cvLzSmQ\nn4ocyYutg/vwtBAP
9+tSEelP6xqhFJUxjTidwMfGpc5KycYASBpolupZxoErly5rC\nnxG0bXUDXI+FQBwy7Kp12AyoeO
A==\n-----END CERTIFICATE-----\n-----BEGIN CERTIFICATE-----\nmIIFoDCCA4igAwIBAgIQ
aTaqi2oHz012L3YTFudOhjANBgkqhkiG9w0BAQsFADBA\nmQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUQ
xEDAQBgNVBACTB0JlbnhXaXEDAO\nBgNVBAoTB0dlbWFSdG8xGjAYBgNVBAMTEutleVNlY3VyZSBSb2
90IENBMB4XDTIx\nmMTExNzA5Mzc1NFoXDTMxMTEwNTA5Mzc1NFowWjELMAkGA1UEBhMCVVMxZzA5BjBjNV\
nBAGTAK1EMRAwDgYDVQQHEwdCZWxjYw1wMRAwDgYDVQQKEwdHZW1hbHRvMRowGAYD\nnVQQDEwFLZXl1ZWN1cmUgUm9vdCBDQTCcAiIwDQYJKoZIhvcNAQEBBQADggIPADCC\nnAgoCggIBAJu/0CD4wQekPWqbidlB1
DT3SOp8qWVzThnOqf9c8YcMMptlD8ca2AJG\nnwv8yq6UP5xKPzIXSvP38MvtdQdkk/NS9/k23CQt5XxYF
ExgTy7bQhPNfuf3wp6iH\nnD9/remnttB4yeunc+XwgFEJ48zXdlWwU8BVkpWK4DV7FB/30/GX0QpQvrCp
Affgj\nnNEMVasgAtDBlDe9wG1/YATncSP4SrXjmUKXmCy4g9PDIkdy+Yde3uclmDok1o3UN\nnE3c7tSUD
uc8TJtZUXXn67Xld87ocleCoRoPrUmpH2HrFIRzvZKy/SS9G5qM0JA1Y\nnvzqAy/3ntXpJO3X2FvRh3sH
bDmIrAQv1GWSMUedBQRMFPA8V3gKN86FQfovnQzhc\nnQylfmV7cG2e+2m8Q8FudEeoPfi93CXRd9hqWNE
Yr+ValfsQb4CR7w1rjsQl3Wo6j\nnVAKxug4ficyEWYVrQUuz3xF2+86iXJsquCzK1P0vRTiW33Agu3Xh
qZjSzTH8y1P\nnASnZgBp2+Avbn/RFHU66c6y+sqAltk8wMkHf2da7LH4Q6kAgPkm2WyBUcOi2kxYN\nnXm
jLFY7sMREDVf2UYek1Mw5XdxsTSOP6ije4+8M8I3CbqhVzWpDS3SwBZVSCU+VD\nn4+p3hupy5TcQ78Tv0
fUOgod0pwn0KHnQaDGDLaPhoi0WtrA7ttM1AgMBAAGjYjBg\nnMA4GA1UdDwEB/wQEAWIBBjAPBgNVHRMB
Af8EBTADAQH/MB0GA1UdDgQWBBDToXJ+\nX/v5BV+01fJsQ4ESkZERCDAAeBgNVHREEFzAVgRNzdXBwb3J0
QGdlbWFSdG8uY29t\nnMA0GCSqGSIB3DQEBECwUAA4ICAQBdeVQhdJJehxaChK3VF4MxxRgdRAjgjG8Otj
Qe\ngdziluPZNNiHD0t3QV/WYta5+ChDpbtSMf+HWyIKCo0IekljRHNjHz2LqnXO/Rnq\nnqGVbZ6ctPi
N/kF44naDezMxE6VEoa4XepbAIPiQa9WxohtLN2GH1GNXFLsPFRXxt\nnVE0Do5hBRH41QagdC1cIRwwFy
kRENZmwQhkgTDPN5NsZWEtzEoXJTRiOxNyGv8t\nnzWv4dWm+fNjJfeGqdCuetv2RhF5RYb0TkecER0Tp
Xc2/6wgFWOpAKvB1jDNXJFjg\nnzJK0K+4Kkuwx0odI0brD/NdmivfW6gneMfKMuxEpHc7AnVNst+OL4U
6oky9vCOz\nnvMyb3ysF7fXnE7ltRWgg8YrlqHucP+kAkU0XvNv664N8qlMb6gjFqb+XG2njrzHA\nnHSNn
DwRwNdjb0410kMgkBzo8v6doIjJcZGZQy0RGuzz8mse0KeqaCQwgALEB+UW9\nnN6dReEK03W7IS50JK6B
u2F82zA6teCKVWWhncDVfmDS+UYYXYs5DRmOp7a+Ly1s\nnEb9aDFDkEW1foheJakWXQAJdhWAgJYO4lM
p7DKunFB5QunPWp38rX1hC1A4CKAkg\nnxXVj92YLoLmTQWOHeG3IQ8mJZ1z2a86RqvWBSQzeMqA1VJRFE
a/QE6p7c/oK0ks+\nn+fpqqQ==\n-----END CERTIFICATE-----\n"
}

```

Example Request 3 for Local CA

```

ksctl clientmgmt clients renew --client-id 47e5049f-770c-4be8-85d2-51bc9c39f0ba
--subject-dn-field-to-modify UID

```

Example Response 3


```
ksctl clientmgmt clients renew --client-id 1e04f15f-0de7-4e13-9c5d-2a84e26ee947
--clientcsr ./csr.pem
```

```
{
  "client_id": "1e04f15f-0de7-4e13-9c5d-2a84e26ee947",
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\n\nmIIEcjCCAawICAQAwNTEzMDEGA1UEAxMq
dG9rZW4yMDk5YWI5MjktMTk0Zi00ZThi\n\nLTkwZWMTNTk1Mjc5NTgzYWNlMIIBIjANBgkqhkiG9w0BAQE
FAAOCAQ8AMIIBCGKC\n\nAQEAsKK02wJspkm8eeVJZ6r9aexy4Nk+7vlaMsVgTBjI2yZrtgbkbNFQah1cwK
ny\n\nC3ZNF5X4USFGHFs+msgoCEFUb3leMQb7WS6yPsRgM4Z4E3WIu9bDNBP9MUXuSE1k\n\nnz6uv9qbDLD9
TVghkMIIU6wJuCS3OidIww8ZtZEkhQvU9lQpmGSOAhWYonCdRvNaC\n\nnoqKGbDrFRU+7siK8+L0rea+lcN
1frdWjkyB//GzuhhqJ2zitxRo+Zukx1PAS59Gi\n\n2sAnWJTmAdw/eCyMJo2HG9R2dKmAtmMx+dgO9nNnB
0+FK0sEC1SHTNwaGALmX5rT\n\nn1IdDxeHUrjE5K/7dLC32qcqC1QIDAQABoAAwDQYJKoZIhvcNAQELBQAD
ggEBAABJ\n\nnUpqzjZwgq+RV0ph33GMK2s2fyv4tbOQ+jwdsWfQrYVdBjEvxu9xD1/Ou9QN+4L2T\n\nnhXFfs1
WLyDWYxZyCu+4vPVYLBxZEnOqZblixiljmJ8wykYk2dyttNPOQo22ZPLTHx\n\n\nnviMZ81qZu/oGGk4ia9YB
svjMLPsbFzW34xWLTiEYgiNDlPhsZxQbxOwQ6H2VsCAQ\n\n\n6c/cF0LkbZZvKunrNb//KnfF7VmPFhZQkdt
uagwAX/lLdymVGockFLviYcxTP6nO\n\n\n7xUzUVcBRadZib+8dQYeRnUW2+qAQeWffLFqQrFG9xr8UWQt8S
hDWuTwZpWnA9V\n\n\n8JmUnCIUouoaJWZxP+Y=\n\n\n-----END CERTIFICATE REQUEST-----",
  "cert": "-----BEGIN CERTIFICATE-----\n\nmIIE/jCCAuagAwIBAgIQIzY+3boBkdC35RH2NMx
gfDANBgkqhkiG9w0BAQsFADBa\n\n\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUQxEDAOBgNVBACjB0JlBGM
NhBxAXEDAO\n\n\nBgNVBAoTB0dlbWFSdG8xGjAYBgNVBAMTEutleVNlY3VyZSBSb290IENBMB4XDTIx\n\n\nnMTA
```

```

zMTEwNTQzM1oXDTIzMTAzMTEwNTQzM1owazEzMDEGA1UEAxMqdG9rZW4yMDk5\nYWI5MjktMTk0Zi00ZT
hiLTkwZWmtNTk1Mjc5NTgzYWNlMTQwMgYKZCImiZPyLGQB\nARMkMWUwNGYxNWYtMGRlNjY00ZTEzLTl1JN
WQtMmE4NGUyNmVlOTQ3MIIIBIjANBgkq\nnhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsKKO2wJspkm8eeVJ
Z6r9aexy4Nk+7v1a\nmSvGTBjI2yZrtgbkbNFQah1cwKnyC3ZNF5X4USFGHFs+msgoCEFUB31eMQb7WS6
y\nPsrGm4Z4E3Wiu9bDNBP9MUXuSE1kz6uv9qbDL9TVghkMIiU6wJuCS3OidIww8Zt\nzEkhQvU91Qpm
GSOAhWYonCdRvNaCoqKGBDrFRU+7siK8+L0rea+1cn1frdWjkYB/\n/GzuhhqJ2zitzRo+Zukx1PAS59G
i2sAnWJTmAdw/eCyMJo2HG9R2dKmAtmMx+dgO\n9nNnB0+FK0sEC1SHTNwaGALmX5rTlIdDxeHUrjE5K/
7dLC32qcqC1QIDAQABo4Gu\nmIGrMA4GA1UdDwEB/wQEAWIDiDATBgNVHSUEDDAKBggrBgEFBQcDAjAMB
gNVHRMB\nAf8EAjAAMB8GA1UdIwQYMBaAFK5TxeYX0HEIee7MBusV6YZRlpiJMFUGA1UdHwRo\nmEwwSq
BIOeAeGRGh0dHA6Ly9rZXl1ZWN1cmUubG9jYWwvY3Jscy9jNTA2MjY4Zi0x\nYzUzLTRiOWUtODk1ZS1iZ
WZjZjg0MTgyYjUuY3JsaMA0GCSqGSIb3DQEBCwUAA4IC\nAQAYMDPTgxvfrtgv1SeBkvJUKTAfaonnUe5K
T6nb/AixPCPY3qNRRZ6nhq1GMHro\nVBzoFAupdr8INAP/N81x8WUGMrBJ6nuHulzf5KnXTNaa5M26BD1
oCIugNyBrxIuf\nndlczzF5qPBb7JGIzOcpemvOVi4ObHkWzw4S12MIn+4fIXhv6G8vH10QgGNYK5Iy7\n
bc6+BwVEQmXITfEaWrkmeVxjZKnqiI78qjoxqGKb67G9ob+oCJPVVJg2Ex67n/9k\nnq32G1Oqlc7Oh4Zg
j7tWm6Qvng8sn14VfCyfSd017QZrB456197OZ8qQhBhdWw5MY\nn77IbHp5w600/zqQFahBxSiQfpreKZb
UFY26NE6ghmEKlu93i3CKYbmEeibn2sjyV\nn9zzvTD5pFd0mPoeTZKBXddsK8mcqZhiAGvkKRUYONJkRS
RMUMqzBjzCaO4t7xcDU\nnRAstoeFu4FbvtrHo79/qZFASnQm2JskpmTvyXj4uS7j+W6SH/w0mkjTk3XIX
IXMK\nT4guoP8Zk+Dpfftcs73WRY4AjBqIa7y2dL2VyevWTe6S5bJHNMLLHqxOM3bDPbx\nn14GJpi+q4
4/fu85eZSDWo23zAFfh6DTg/UhTbIV6OKmbPH6mvFetG5tU2YnmX55d\nnJYAu7jmgBLs7iIGV4kEyyUKh
jzGjZKlyMVL6CnHj7PdXGg==\n-----END CERTIFICATE-----\n",
"ca_cert": "-----BEGIN CERTIFICATE-----\nmIIFoTCCA4mgAwIBAgIRAP7Ey0klaW058b3E
HqrNyS4wDQYJKoZIhvcNAQELBQAw\nwJELMAkGA1UEBhMCVVMxMjY4Zi0x\nYzUzLTRiOWUtODk1ZS1iZ
CZWxjYw1wMRAw\nDgYDVQQKEwdHZW1hbHRvMR0wGAYDVQQDEwFLZXl1ZWN1cmUubG9jYWwvY3Jscy9jNTA2MjY4Zi0x\nYzUzLTRiOWUtODk1ZS1iZ
MTEwMzExMDQxMDZaFw0zMTEwMjY4Zi0xYzUzLTRiOWUtODk1ZS1iZ\nMTEwMzExMDQxMDZaFw0zMTEwMjY4Zi0xYzUzLTRiOWUtODk1ZS1iZ\nGA1UEBhMCVVMxMjY4Zi0xYzUzLTRiOWUtODk1ZS1iZ\nGA1UEChMHR2VtYXw0b2EaMBGg\na1UEAxMRS2V5U2VjZjJ1IFJvb3QgQ0
EwgGIIeMA0GCSqGSIb3DQEBAQUAA4ICDwAw\nnggIKAoICAQDGA3nahTIT/WlDzURGLgFDN0kDwwDn+fuJ4
Vak5Jy5MW73elibQoRP\nXdcM5v7AYR197IjhrGTOprTxqxXmLhjY+XDZ1bPCEzzE6ZKNtHpDL6U147DF
Yc7H\nnq4TfkLpV7fWnXjleZCEhU8Z8XNjqGndnh4tq6MJ2C+C2t1NbbhmlYdSRwULzt+md\nnbeu074Ko6
DGEJ26rNGfTyAm79k1+YamcWbk1T+ct3bhn
ce6pf2MM3jI4cWLBmXwS\nnaKReZmf9uBFGbVhjMnWG
IxBgGDnY6mgEHLXSqEvhw93EEhOHNYOf8Hy1HQTNdPSY\nnMDtkI5kQ0/XjEH3dSSFjwOv1XJGryx2W5Bq
D0UQHXSIIzOm+ao/+URKj/LcZ+V6y\nn4rGJU7jrPwiYsDqxxEMEz6o17QHfuQBfoGFqRizvgPNN75WYhT
0TgG3BmTW3KAmD\nn0zEM+nvwMDPELU7SgzX1Ufj83XqoCLsGLzWH4InY3S0H1T7HVATay2zgjpKH/Sp8\n
nOTh71LFARyGo9PvxI3tdjZNYMD+/rXA5qA5fmq94DlGikJHMZ5n2NP+1Q+I+rsv\nnpmtKDZ6QQ1Wzpo
ORGM8KOHkGjYkHwv5NdmhxgCqzzVS1HGnWrcqiKuwWw1Dh7VI\nnYYgo+INGKdBNqnLwI77vdjGWkCgjY
i9uIXtNz4n0zH60CreKDgOpfQIDAQABo2Iw\nnYDAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAAQ/BAUwAwE
B/zAdBgNVHQ4EFgQURlPF\nn5hfQcQh57swG6xXphlGwmIkwHgYDVR0RBBcwFYETc3VvcG9ydeBnZw1hbH
RvLmNv\nnbTANBgkqhkiG9w0BAQsFAAOCAQEACEE+oAGb2HTTEvshvAu4iSJe3Z3mx31Ei3/o\nnMSEkyo
P/aotxfrVsgFNQNTpVvGfCJJxBCO6YZUnCLH267Aa2wGClguL/NESUdaK4\nnX7HXZhAK4Bhc+eFGKjdUz
VSg650ToYnq1IVdZ+AO6JII9xHAQthZoHu2c3ylrT/k\nn/phd8Ro23sfFYs47zkk169vL/A4pKofgSjfp
LhrVuEOFFZRFcl8nPt9tSUNXmQcN\nnzYvDnfep4Ka81KIrMmxtocuHdDQoZvUy6U+Akrf9GNiGYUBfNCn
xFQflmXKArknL\nn3RqBjZWwS0zKnSN4QV/1CKH01C5xnkviRTNNoYbSL5mRvuGiJ7+iGRu0c5NFoBM3\n
nuVA9Sar5VZ8xS780ONB2ud4Ci1cWdGhOqGPJugSo/qSXJuKYH7Y8W+T/Aaqq961z\nnwdSlr17tQVkm50/
eyNRsWo91xu3Gbnf/ccdsFADAvwzyJTU8Wum8yFEHPkAhx2r7\nn4fEBVi94LLz4msqGvsIBU/GUWieDVw
PysZzMVABFGpS1jFYltZ3MvkJJn5h8CasA\nnvOPeqLUdGSG/P7aHzyJR58s3f3Y15/tQgM5McyOvBtK01
SOKWRrmvMpF8wwiNwik\nn0gz2OzryK/BkXO7iNO4GRFNj/ynmMXFZuSw1L8HRLjU0yEjwXKLZmGpUFzgp
W9LX\nnJW+cMJE=\n-----END CERTIFICATE-----\n"
}

```

Example Request 5 for Local CA

```

ksctl clientmgmt clients renew --client-id 1e04f15f-0de7-4e13-9c5d-2a84e26ee947
--clientcsr ./csr.pem --do-not-modify-subject-dn

```

The CSR file must be in the PEM format as:

Example Response 5

135


```
ksctl clientmgmt clients renew --client-id b6a182a8-4f4f-4d0f-a04f-ff191d1471b5
--ext-cert ./cert.pem
```

-----BEGIN CERTIFICATE-----
MIIEuTCCAqGgAwIBAgIQQW5X4dR3NWdQSsID8AZKtZANBgkqhkiG9w0BAQsFADBa
MQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUQxEDAOBgNVBACTB0J1bG9hbnR3X2Ew
BgNVBAoTB0dlbW93dG8xGjAYBgNVBAMTEUtleVN1Y3VyZSBSb290IENBMB4XDTIx
MDUwNjA0MTkzN1oXDTIzMDUwNjA0MTkzN1owJjEOMAwGA1UEAxMFYWRtaW4xZDAS
BgoJKiaJk/IsZAEBEwQxMjMxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEA80JhDvdDwRjsbu7s6a3i3sPlgXtzx6W5oE1ReqfYsJPyYmWfaTDn3lS0rUI8
Y+1h3QUovyTKcZiGzPqbEV18MOJvH3IXzTgl/vfyng6nuIV3XU8Yg/8Ic+u52wMv
70GYmziMLiHaE/tnM1VGUxd2SmJ+dbVkp6CL4kWa5j1J+GMjVxkwFZbC8+LXwbmI
6kSOanbgVaksNdvaa8e0lOoO+76APTk8Jv/5sUB7QVUpNEk1yyJ6yU107MGGRZF
Q65DNFKVkyqya2zkB5vKS5cypEC3Uv5Yd8th17Q2Mfr1FJJCOSsF8+zv6XnHTFz1o
8pN6VZSzUkHOUDZbMxxnOiV6SQIDAQABo4GuMIGrMA4GA1UdDwEB/wQEAwIDiDAT
BgNVHSUEDDAKBggrBgEFBQcDAjAUBgNVHRMBAf8EAjAAMB8GA1UdIwQYMBAAFCGv
Zl+ix1Auz/iCYyKLs95b02RhDMFUGA1UdHwROMewwSQBIoEaGRGHdHA6Ly9rZXlZ
qwn1cmUzGjY9YjYwvY3J3cys84ZWM2MWIzNC04NWRLLTQyMzEtYTRhMC1jZjZjMwVm
NzZmYTtyY3J3sJMA0GCSqGSIb3DQEBCwUAA4ICAQBSFYog1y5cUen7y+t3gOGnJx3Z
8ZtRkRTNAD1C06a7fKmlqMmVO18dpERCn5MIef0dggdydqzFzIxUDXjHTGfFppHtkv
4JtDPe6Mx0Esxz/25zJVRp2B77+v4VVWPqZx07Yr5r6+4UI4M6TZDNAICF/ooYC+

```
{
  "client_id": "b6a182a8-4f4f-4d0f-a04f-ff191d1471b5",
  "key": "-----BEGIN RSA PRIVATE KEY-----\nMIIEoQIBAAKCAQEAtMyZxEVwy0M/FtVKjWhkXeSxaGP8vlsfQuZJldgCFqmp72eF4IO6v8L+Xy/3qL26RcY9o2V51yafm6pRHZP+F1r+KeSW3i0v515R1WWL+q05y55i\Ny9DmF6pFvB8kFvNCvhoysXj+tlxaMGXZnrAGz+v/oxnhjCad6MQOH81SV6Qb/1I7nf2zXDgi194nCWFm2s+Gg47KUJiprkzWiA+4Ku+KtmuehphjQhRGY2GptfCXn6bx\nn1DhjH+Q51Y/9JZEpyAN66hXU1nADQKBCyZp74te7anMss7Ye+k23Dsr9tuOAPvwy8EdvFXlo6Q8WY5ceuGc5rDIOAAIq6kp4/RSDCQIDAQABoB4wHAYJKoZIhvcNAQKOMQ8wDTALBgNV\nnHREEBDACgQAwDQYJKoZIhvcNAQELBQADggEBAJDa0vLtJ3fEYV0J6UukzVo6gGoc\nnuz6wT6GgLonCye4coAcYDW1V1Lv6/y0DgWiTQLJiv13NLTxQDvk6jXJ4Ep3+vNMf\nnFW70Luwfbj0naFnSI/X7JSEyNc76V693pqFgVslD2/mzPmZr1JidI5sT7xibUGMK\nnndbh6veo94I3ZDvcvV9WdqZfSBFA71+hpms+ug0z6KaCE3SGJcD5/rO4uTFT2byS\nngblUPjqCkoXQtTwRPgOYjXUG3s9sqQfm5mMrOkj7Ni03tr/TQKHjawghM0Z5xLec\nnTfxcLZZ3vsJ/kvgOr6gbrsXM1HArEnT9zpW2nFRaCxZYq1LbLwDUQtTdW/I=\n-----END RSA PRIVATE KEY-----\n",
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIICmDCCAYACAQAwNTEzMDEGA1UEAxMqdg9rZW4xYTM2NzZmMDQTMjFhNi00ZGMx\nnLWIyNDU0OGMzMDFiMzhjNzZhMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC\nnAQEAtMyZxEVwy0M/FtVKjWhkXeSxaGP8vlsfQuZJldgCFqmp72eF4IO6v8L+Xy/3\nnqL26RcY9o2V51yafm6pRHZP+F1r+KeSW3i0v515R1WWL+q05y55iY9DmF6pFvB8k\nnFvNCvhoysXj+tlxaMGXZnrAGz+v/oxnhjCad6MQOH81SV6Qb/1I7nf2zXDgi194\nnCWFm2s+Gg47KUJiprkzWiA+4Ku+KtmuehphjQhRGY2GptfCXn6bx\nn1DhjH+Q51Y/9\nnJZEpyAN66hXU1nADQKBCyZp74te7anMss7Ye+k23Dsr9tuOAPvwy8EdvFXlo6Q8W\nnY5ceuGc5rDIOAAIq6kp4/RSDCQIDAQABoB4wHAYJKoZIhvcNAQKOMQ8wDTALBgNV\nnHREEBDACgQAwDQYJKoZIhvcNAQELBQADggEBAJDa0vLtJ3fEYV0J6UukzVo6gGoc\nnuz6wT6GgLonCye4coAcYDW1V1Lv6/y0DgWiTQLJiv13NLTxQDvk6jXJ4Ep3+vNMf\nnFW70Luwfbj0naFnSI/X7JSEyNc76V693pqFgVslD2/mzPmZr1JidI5sT7xibUGMK\nnndbh6veo94I3ZDvcvV9WdqZfSBFA71+hpms+ug0z6KaCE3SGJcD5/rO4uTFT2byS\nngblUPjqCkoXQtTwRPgOYjXUG3s9sqQfm5mMrOkj7Ni03tr/TQKHjawghM0Z5xLec\nnTfxcLZZ3vsJ/kvgOr6gbrsXM1HArEnT9zpW2nFRaCxZYq1LbLwDUQtTdW/I=\n-----END CERTIFICATE REQUEST-----\n",
  "cert": "-----BEGIN CERTIFICATE-----\nMIIFCzCCAvOgAwIBAgIQan15qavOd6PS4Yduu3hJLTANBgkqhkiG9w0BAQsFADBa\nnWQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUQxIDAeBgNVBAMTB0JlbGEnbHhXAxEDAO\nnBgNVBAoTB0dlbWFTcDg8xGgYAYGBNVEBAMTEuVn1Y3VySBSB290IENBMB4XDTIx\nnMTEwNzEwNTEyMloXDTIzMTEwNzEwNTEyMloZaEzMDGA1UEAxMqdG9rZW4xYTM2\nnNzZmMDQTMjFhNi00ZGMxLWIyNDU0OGMzMDFiMzhjNzZhMIIIBIjANBgkqhkiG9w0BAQEF\nAAOCAQ8AMIIBCgKCAQEAtMyZxEVwy0M/FtVKjWhkXeSxaGP8vlsfQuZJldgCFqmp72eF4IO6v8L+Xy/3qL26RcY9o2V51yafm6pRHZP+F1r+KeSW3i0v515R1WWL+q05y55iY9DmF6pFvB8k\nnFvNCvhoysXj+tlxaMGXZnrAGz+v/oxnhjCad6MQOH81SV6Qb/1I7nf2zXDgi194\nnCWFm2s+Gg47KUJiprkzWiA+4Ku+KtmuehphjQhRGY2GptfCXn6bx\nn1DhjH+Q51Y/9\nnJZEpyAN66hXU1nADQKBCyZp74te7anMss7Ye+k23Dsr9tuOAPvwy8EdvFXlo6Q8W\nnY5ceuGc5rDIOAAIq6kp4/RSDCQIDAQABoB4wHAYJKoZIhvcNAQKOMQ8wDTALBgNV\nnHREEBDACgQAwDQYJKoZIhvcNAQELBQADggEBAJDa0vLtJ3fEYV0J6UukzVo6gGoc\nnuz6wT6GgLonCye4coAcYDW1V1Lv6/y0DgWiTQLJiv13NLTxQDvk6jXJ4Ep3+vNMf\nnFW70Luwfbj0naFnSI/X7JSEyNc76V693pqFgVslD2/mzPmZr1JidI5sT7xibUGMK\nnndbh6veo94I3ZDvcvV9WdqZfSBFA71+hpms+ug0z6KaCE3SGJcD5/rO4uTFT2byS\nngblUPjqCkoXQtTwRPgOYjXUG3s9sqQfm5mMrOkj7Ni03tr/TQKHjawghM0Z5xLec\nnTfxcLZZ3vsJ/kvgOr6gbrsXM1HArEnT9zpW2nFRaCxZYq1LbLwDUQtTdW/I=\n-----END CERTIFICATE-----\n"
```



```
Af8EBTADAQH/MB0GA1UdDgQWBBD0XJ+\nX/v5BV+01fJsQ4ESkZERCDaEbgNVHREEFzAVgRNzdXBwb3J
0QGdlbWfSdG8uY29t\nMA0GCSqGSIb3DQEBCwUAA4ICAQBdeVQhdJJehxaChK3VF4MxxRgdRAjgjG8Otj
Qe\ngdziluPZNNiHDOt3QV/WYta5+ChDpbtSMf+HWyIKCo0IekljRHNjHz2LqnxO/Rnq\nqGVBZ6ctPi
N/kF44naDezMxE6VEoa4XepbAIpIqA9WxohtLN2GHlGNXFLsPFRXxt\nVE0Do5hBRH41QagdC1cIRwwFy
kRENZmwQhkGTdDPN5NsZWEtzEoXJTRiOxNyGv8t\nzWV4dWm+fNjJfeGqdCuetv2RhF5RYb0TkecER0Tp
Xc2/6wgFWOpAKvB1jDNXJFjg\nzJK0K+4Kkuwx0odI0brD/NdmivfW6gneMfKMuxEpHc7AnVNst+OL4U
6oky9vCOz\nvMyb3ysF7fXnE7ltRWgg8YrlqHucP+kAkU0XvNv664N8qlMb6gjFqb+xG2njrzHA\nHsNn
DwRwNdjB0410kMgkBzo8v6doIjJcZGZQy0RGuzz8mse0KeqaCQwgAlEB+UW9\nN6dReEK03W7IS50JK6B
u2F82zA6teCKVWWhncDVfmDS+UYYXYs5DRmOp7a+Ly1s\nEb9aDFDkEW1foheJakWXQAJdhWAgJYO4lM
p7DKunFB5QunPWp38rX1hC1A4CKAkG\nxXVj92YLoLmTQWOHeG3IQ8mJZ1z2a86RqvWBSQzeMqAlVJRFE
a/QE6p7c/oK0ks+\n+fpqqQ==\n-----END CERTIFICATE-----\n"
}
```

Self-Renew Client Certificates using ksctl

Note

The `cert-duration` parameter is deprecated. Any value provided in this parameter will be ignored. If `cert-duration` is provided in the profile, the value specified in the profile is used. Otherwise, the server default value will be used.

To self-renew client certificate, run:

Syntax

```
ksctl clientmgmt clients self renew --clientcsr <CSR-signed-by-Local-CA> --subject-dn-field-to-modify <Subject-Distinguished-Name> --do-not-modify-subject-dn --alg <Algorithm> --cn <Common Name> --dns <Subject Alternative Names> --email <E-mail-Addresses> --enc-alg <Private key encryption algorithm> --ips <IP-Addresses> --names {<O:organization, OU:organization unit, C:country, ST:state/province, L:location> --pass <Password to PEM-encrypt the private key> --private-key-bytes <Private Key bytes of the key> --size <Key size>
```

Example Request 1

```
ksctl clientmgmt clients self renew --alg ecdsa --dns *.thalesgroup.com --email contact@thalesgroup.com --enc-alg AES256 --ips 1.1.1.1 --names {O:Thales Group,O:RnD,C:US,ST:MD,L:Belcamp} --pass KeySecure_1 --private-key-bytes -----BEGIN EC PRIVATE KEY-----\nMHcCAQEEIEj1liesPsLdk0tM7Jv87sruegOPdmji9SY3s3ncdckxqoAoGCCqGSM49\nAwEHoUQDQgAEL8cvuduRZs6e/vsttMlhi9HxV+0FzhCg/zHUmXNmyH5KlmQgoaql\nvfwHqQk79lf+55WSLD7uUwaxhYwGHIapw==\n-----END EC PRIVATE KEY----- --size 256
```

Example Response 1

```
{
  "client_id": "1a6b1e6c-bd80-4936-9db3-f85d1d547b01",
```



```

1bGNhbXAXeEDAO\nBgNVBAoTB0dlbWFSdG8xGjAYBgNVBAMTEUtleVNlY3VyZSBSb290IENBMB4XDTIx\n
MTEzMDA1NDY0MlOxDTI0MDMwNDA1NDY0MlOwXDELMakGA1UEBhMCVVMxZzAJBgNV\n
DVQOHEwdCZWxjYWlwMRAwDgYDVQOKEwdHZW1hbHRvMRwwGgYD\n
FsmIICijANBgkqhkiG9w0BAQEFAAOCAg8A\n
3vDj+hCVgIEqjlV5vow\nd1X/mmEXBpzGDTZLnM50106Mfj8jzp1TGooNe5/hPHLJEPUnLGS2FFiSYZ4K
CZb7\nnzV84iRkL63/OU47QntmtwuqPJg4SAQkBe6ekr7WrJr53ZlfsON+c6ciKBA9MqCL9\n
f+WcFZrpbWEehrAyyvlSEoletbjMtPQ92XGUjJuRcnp2rFuh1AR9NHGv8\n
NXyKfoL5H6OeTh9es7clrm8lg3M0Ssgy9pH6F4/ej\n
FsiYOMVrCmHS5xykISHHQhs5W\n
LzuhAd02Y+2\nqriDoueT6PNEfFAE77CidILPELWSsFmsT56FlUQj1jveATjvou0nJiAHDdbkY/\n
dPIrtVqyNGT9qyHJXXa9axBY2Sbo/pDGvc0X0R2IMtjwU08\n
91rB2LQZIJR6cvUf6R7eD3sCAwEAAaOB\n
BBQUHAWewDAYDVR0T\n
EE\nwSqBIOEaGRGh0dHA6Ly9rZXlZWN1cmUubG9jYWwvY3Jscy85ZmZl\n
0znjM3MGJmNDhlODYuY3JsmA0GCSqGSIb3DQEB\n
KR98D5dSBZdlCzj3/SMF0RQ\n
EdLmRWnO\n
Tcgo/HEBDjO9DhRxTE3wmUleKDpY4Xv9iPMfELnm36Hm47Ob0Txdk+KD+G1\n
itefEwjicmTkf8BGNTe5e6+pRzHwrlAn3AtkydufPnuX\n
F0dcjq9Kj5mJUFUnfIPuRPY36oDph\n
IImnyNg4OPwjaK\n
nlJmQMt2XBH06wNV58xFAQ3VPhtZM6JuvAQGD/o908qmOuDVhMcmMMu0FsJrCijxTc\n
b+NduyIbQ2DqcFBMisV4tgyhdBGHTtD7DIL0y+Rx0fU9jwz0hny\n
z3LWz/U2J9YMZpIlyvyfB+Xy8g==\n
E-----END CERTIFICATE-----\n
E-----\n
GA1UEBhMCVVMxZzAJBgNVBAGTAK1EMRAwDgYDVQOHEwdCZWxjYWlwMRAw\n
owGAYDVQOQDEwFLZXlTZWZlcmUgUm9vdCBDQTAeFw0y\n
FoxCzAJBgNVBAYTA1VTMqswCQYD\n
YWw0bzEaMBGGA1UEAaMRS2V5U2Vj\n
gIKAoICAQDb9AihTQfx/QoOuhYOEVmAfXF0W0EQhNAPGIAzTdpHZ1fkpe9qqZAR\n
qgovHzNyYU/TA6tFhduS91bjIOW2aS4wAlDGRLA7LIfcYnV8\n
cxHwXtBS0SV676qd/DnyqxS4Spj4GyHT1\n
6UFijcnCicuWI9W3MS\n
tL7\nkTxBVSzvv66rzMKqmY9gSs8tenFbSuK58tGUxNR6HokLCBi/vsIokN\n
oGJphkm1J5J0YH4vvvYlxG425Z8NP0WVlMdvL+OG\n
5/3ZV6VjQFalmMcKp6gCzRVx\n
YGime7c14\n
BgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUWM2e\n
t/7PowHgYDVR0RBBCwFYETc3VwcG9ydeBnZW1hbHRvLmNv\n
ic4SRjia1yo8avfDVoXcqmFZJVE2Ym\n
l68D2D8LmUubjVBT\n
M\nfhHGA563FyluiLPvD0xuYeJxmwCoTqWyRla06aYUG1LAksRLlJHD\n
YnHLZQ6B+tF2EOblrURGZ3DcZJ9Kr6/xElyz2\n
0vhvj5DMc9STDxNNAYiu31\n
KZhWHBK\n
FwX5BAk+epiWVLimr644aLyAZh8tf5rT4EcbqtWeQQwpJZDhmd+ykk42Kn\n
dPhhpShqheG+C7pCE6SgEZM8LtwHprKp1XjkIK9g5Yg\n
\n"
}

```

Example Request 2

```
ksctl clientmgmt clients self renew --subject-dn-field-to-modify CN
```

Example Response 2

```
{
  "client_id": "1e04f15f-0de7-4e13-9c5d-2a84e26ee947",
  "key": "-----BEGIN RSA PRIVATE KEY-----\nMIIIEogIBAAKCAQEAt/6mVPE+16G0WcwXDnM5X
VX8yyg3XqkHDXWjigdk0K3OLtUX\ndweXQ9wTm/DUr7njEXJ+qM7tw NP/D5fA9V08+/qMjOGBafI0
uS4TGp3SXMzJ5KDc/n6zbNoV7+lbAG/x4RpVpPDK0Mzd/4w0YnkREmFx/I7oqGVLrjm/Qdz1FC/qLqOUM
p\n9DNcbNV03IKhV0e96rH0n197qhSTUDtykF2VYwLRrYxKTygKo2pn1Hgi2+4VMjH6\nnhWtM32raJFXT
QdiTUrauM9lOySnTrXvtlKJmY1Fc7akSTLaJTtKSR2z6XiEmruOJ\n8B1Xpkhq3X6iBf1fgye77+uWuG
UW7xPpKu3TQIDAQABAoIBAHAQMrCz0FH1Upk2\nnyXi3STTa7w4euSdXdkL3Fk4gitZT2XQWPoumKZRB4a
TwGK2v9CFPDU1MIqrKMSbk\nhCRAU8bv2dgCjHkv6UeW6BqTrnW8RTD9c75iTz14VEcy6h9FSguqWRw
K+yBC/jMf\nJun1Cf2Gr0ACLXaxEa/1OH+NFSOoPh+FPISxYq7/XhrmLevlgNby1UQLMbFi2b7\nnRVE4
cd2POwIIzN15LN1SvRtxBYj+r4Y75f046sgCUD9r2gP3PT/2ENcx3r57JR4b\nn7Ghmuzgj25HR7ki0zJH
jgH7hNX+1Yh3VGWUNjqPU4IeAeB2jTGpcOTAVp/o4vtGY\nnFb5UZMECgYEA2cm9j0QVQ61y4TE554510O
MQ10gMZQd18H8w3bwxmyMkSXbFOaha\nnM5arMB1K0AhRpyvffsL66lnQPhLXmvVl rpboZaZzJpK0t89Op
b2D9mkP7fA15E0X\nnX+SwSmO2z0XkfmENL9/dlEpE/1uDuYxtv8BgHdbplhwv+0k3bfzdue90CgYEA2Ec
I\nnvqPfzPELHvIW68U6U/mhOZHirDlURGV10sCu1t2Yp9e6P40SkV1Wc56azzeVnA3\nnBZwAVWwUBX
S+b+AN8YLJicp+G6Z88Xm5lIJcUEGLZu3+v4jmIf8ovxzSkOedfDVg\nnT2P6tG7EBxnLj14t1lwvEl8gK
kFFvhTkaQK6jECgYBGTrdwWbVXrNXqVv7vfvDe\nn1YbpSowfbWIoqpb27jtBbv3W7Y3MCEpeXySSmtI9
CIJ7qlEzVfS01RvB/kk2UERS\nnMXGC51JDQJcww68LGHxSfyZSVPrA4hZKscQrr7cjgNe76M7eBrI8niF
aIowP71Z7\ntvYpjmwaRcgU0meq0RVMzQKBgDQ6+2amxr7pup+PiTRzFNa/gAkqX64/UVIBsCGo\ncNF0
ZiOtH0ufUFT7aLq2faP2QC+Zy//jy8TVakvRBmiZ0GQ51H7/mvLsgxIB9oph\nn7Q9IuzwFT71Lv2A1cfu
tZ0CtXsAHV+n0WDBI67hXR2xfmN6XOZsxBuCNgN1FyF9/\nz3WRAoGAaFogxxNUbCWqkLxQeNKeeQmPXW
MsC5Ad8+Uf6aSN+rAH1WsuYQPs5fkP\nniUvUgZwHgk2TSEwwYpYtAOeDFiEya9IdeJyKYpxTKm2dM9ZBv
n5NFTAJTfrrwhrtM\nnK24ojzMxf+VpLSpCCGp7rrrKb80KQLPFA4FVgc/SfI2LFclq0jk=\nn-----END R
SA PRIVATE KEY-----\n",
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nnMIICmDCCAYACAQAwnTEzMDEGA1UEAxMqd
G9rZW4yMDk5YWI5MjktMTk0Zi00ZThi\nnLTkwZWMtNTk1Mjc5NTgzYWNlMTIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKC\nnAQEAt/6mVPE+16G0WcwXDnM5XVX8yyg3XqkHDXWjigdk0K3OLtUXdweXQ9wTm/D
U\nnr7njEXJ+qM7twNP/D5fA9V08+/qMjOGBafI0uS4TGp3SXMzJ5KDc6zbNoV7+lbAG\n/n/x4RpVpPDK0M
zd/4w0YnkREmFx/I7oqGVLrjm/Qdz1FC/qLqOUMP9DNcbNV03IKh\nnV0e96rH0n197qhSTUDtykF2VYwL
RrYxKTygKo2pn1Hgi2+4VMjH6hWtM32raJFXT\nnQdiTUrauM9lOySnTrXvtlKJmY1Fc7akSTLaJTtKSR2
z6XiEmruOJ8B1Xpkhq3X6\nniBf1fgye77+uWuGUW7xPpKu3TQIDAQAB0B4wHAYJKoZIhvcNAQELAQOMQ8wD
TALBgNV\nnHREEBDACgQAwDQYJKoZIhvcNAQELBQADggEBAAoI0CSQQ9Kn7mpwuoA5vElF4zOp\ncAFb7O
F7xp8gChJiRhotWfUTO41WmQfEMXGuBgdUcfGgHaNSMrH55C9RoKM44d5k\nnDz+jgps20IotKQFDpgh5Z
HN/TZu6Fxxd50UchJ+mTDu6zikf441LCwOZFp6211E4\nnaBpaHuiYls9DPCXBibex3N3cS8EFBUrm0/Ty
d+yhAhv6NFSd90BOPVHbz1HBYkSf\nnk7ZJzZn3OJK1P6td+2SqEcK0tu20hnDkylcbFs5R3qt2zBMAC+B
RYPROvJ11LN+h\nn2u3E8PFPkRlAOXby2GF0nWFXzYZI6Osy58RKS6Bbx3gJR/kzxU/Z/OJnkV
8=\nn-----END CERTIFICATE REQUEST-----\n",
  "cert": "-----BEGIN CERTIFICATE-----\nnMIIFCzCCAvOgAwIBAgIQILRWHfQfJGJG6dTgAMy
HkzANBgkqhkiG9w0BAQsFADBAnMQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUQxEDAOBgNVBACTB0Jlbg
NhbXAxEADAOnBgNVBAoTB0dlbWFSdG8xGjAYBgNVBAMTEUtleVn1Y3VyZSBSb290IENBMB4XDTEwMTA
zMTEzMTIwN1oXDTEwMTAzMTEzMTIwN1owazEzMDEGA1UEAxMqdG9rZW4yMDk5YWI5MjktMTk0Zi00ZT
hiLTkwZWMtNTk1Mjc5NTgzYWNlMTQwMgYKCCZImiZPyLGQB\nnARMkMWUwNGYxNWYtMGRlNy00ZTEzLTlJN
WQtMmE4NGUyNmVlOTQ3MIIBIjANBgkq\nnhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAt/6mVPE+16G0WcwX
DnM5XVX8yyg3XqkH\nnDXWjigdk0K3OLtUXdweXQ9wTm/DUr7njEXJ+qM7twNP/D5fA9V08+/qMjOGBafI
0\nnuS4TGp3SXMzJ5KDc6zbNoV7+lbAG/x4RpVpPDK0Mzd/4w0YnkREmFx/I7oqGVLrj\nnm/Qdz1FC/qLq
OUMP9DNcbNV03IKhV0e96rH0n197qhSTUDtykF2VYwLRrYxKTygK\nno2pn1Hgi2+4VMjH6hWtM32raJFX
TQdiTUrauM9lOySnTrXvtlKJmY1Fc7akSTLaJ\nnTtKSR2z6XiEmruOJ8B1Xpkhq3X6iBf1fgye77+uWu
GUW7xPpKu3TQIDAQAB0G7\nnMIG4MA4GA1UdDwEB/wQEAwIDiDATBgNVHSUEDAKBggrBgEFBQcDAjAMB
gNVHRMB\nnAf8EAjAAMB8GA1UdIwQYMBAAFK5TxeYX0HEIee7MBusV6YZRlpiJMASGA1UdEQQE\nnMAKBAD
BVBgNVHR8ETjBMMEqgSKBGhkRodHRwOi8va2V5c2VjdXJlLmxvY2FsL2Ny\nnbHMyYzUwNjI2OGYtMWMwM1M
y00Yj11LTg5NWUwYmVmY2Y4NDE4MmI1LmNyYDANBgkq\nnhkiG9w0BAQsFAAOCAgEAIYJoz9y6kR3jGEao
```

```
ksctl clientmgmt clients self renew --clientcsr ./csr.pem
```

```
-----BEGIN CERTIFICATE REQUEST-----
MIICEjCCAWICAQAwNTEzMDEGA1UEAxMqdG9rZW4yMDk5YWI5MjktMTk0Zi00ZThi
LTkwZW50bnR1bWVudC5NTGZzYWNlMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEAsKKO2wJspkm8eeVJZ6r9aexy4Nk+7vlaMsVgTBjI2yZrtgbkbnfQah1cwKny
C3ZNF5X4USFGHfs+msgoCEFU3leMQb7WS6yPsRgM4Z4E3WIu9bDNBP9MUXuSE1k
z6uv9qbDLD9TVghkMIIU6wJuCS3OidIww8ZtZEkhQvU91QpmGSOAhWYonCdRvNaC
oqKGbDrfRU+7siK8+L0rea+1cn1frdWjkyB//GzuhhqJ2zitxRo+Zukx1PAS59Gi
2sAnWJTMAdw/eCyMJo2HG9R2dKmAtmMx+dgO9nNnB0+FK0sEC1SHTNwaGALmX5rT
1IdDxeHUrjE5K/7dLC32qcqC1QIDAQABoAAwDQYJKoZIhvcNAQELBQADggEBAAbJ
UpqzjZwgq+RV0ph33GMK2s2fyv4tbOQ+jwdsWFQrYVdBjevXu9xD1/Ou9QN+4L2T
hXFs1WLyDWYxZyCu+4vPVYLBXzEnOqZblxilmJ8wykYk2dyttNPOQo22ZPLTHx
```


[illegible]

```

3BmTW3KAmD\n0zEM+nvwMDPElU7SgzX1Ufj83XqoCLsGLzWH4InY3S0HlT7HVATay2zgjpK/Sp8\nOTh
t71LFARyGo9PvxI3tdjZNYMd+/rXA5qA5fmq94DlGikJHMZ5n2NP+lQ+I+rsv\npmtKDZ6QQ1WzpOORGM
8KOHkJgYkHwv5NdmhxgCqzzVS1HGnWrcqiKuwWw1Dh7VI\nYYgo+INGKdBNqnLwI77vdjGWkCgjYi9uI
XtNz4n0zH60CreKDgOpfQIDAQABo2Iw\nYDAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAd
BgNVHQ4EFgQUr1PF\n5hfQcQh57swG6xXphlGWmIkWgYDVR0RBBCwFYETc3VwcG9ydEBnZW1hbHRvLmN
v\nbTANBgkqhkiG9w0BAQsFAAOCAgEACEE+oAGb2HTtEvshvAu4iSJe3Z3mx31Ei3/o\nmSEkyoP/aotx
frVsgFNQntPvvGfCJJxBCO6YZUnCLHZ67Aa2wGClguL/NESUdaK4\nX7HXZhAK4Bhc+eFGKjdUzVSg65O
ToYnq1IVdZ+AO6JII9xHAQthZoHu2c3ylrT/k\n/phd8Ro23sfFYs47zkk169vL/A4pKofgSjfPLhrVuE
OFFZRFcl8nPt9tSUNXmQcN\nzYvDnfep4Ka81KIrMmxtocuHdDQoZvUy6U+Akrf9GNIgYUBfNCnxFQflm
XKArknL\n3RqBjZWwS0zKnSN4QV/1CKH0lC5xnkviRTNNoYbSL5mRvuGiJ7+iGRu0c5NFoBM3\nnuVA9Sa
r5VZ8xS780ONB2ud4Ci1cWdGhOqGPJugSo/qSXJuKYH7Y8W+T/Aaqq961z\nnwdSlr17tQVkm50/eyNRsW
o91xu3Gbnf/ccdsFADAvwzyJTU8Wum8yFEHPkAHx2r7\n4fEBVi94LLz4msqGvsIBU/GUWieDVwPysZzM
VABFGpS1jFYltZ3MvkJjN5h8CasA\nvOPEqLUdGSG/P7aHzyJR58s3f3Y15/tQgM5McyOvBtK0LSOKWRr
mvMpF8wwiNwik\n0gz2OzryK/BkXO7iNO4GRFNj/ynmMXFZuSwlL8HRLjU0yEjwXKLZmGpUFzgpW9LX\n
JW+cMJE=\n-----END CERTIFICATE-----\n"
}

```

Example Request 4

```

ksctl clientmgmt clients self renew --clientcsr ./csr.pem --do-not-modify-subjec
t-dn

```

Example Response 4

```

{
  "client_id": "1e04f15f-0de7-4e13-9c5d-2a84e26ee947",
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIIECejCCAWICAQAwNTEzMDEGA1UEAxMq
dG9rZW4yMDk5YWI5MjktMTk0Zi00ZThi\nLTkwZWMTNTk1Mjc5NTgzYWNlMTIIBIjANBgkqhkiG9w0BAQE
FAAOCAQ8AMIIBCgKC\nAQEAsKKO2wJspkm8eeVJZ6r9aexy4Nk+7vlaMsVgTBjI2yZrtgbkbNFQah1cwK
ny\nC3ZNF5X4USFGHFs+mSgoCEfU3leMQb7WS6yPsRgM4Z4E3WIu9bDNBp9MUXuSE1k\nnz6uv9qbDLD9
TVghkMIIU6wJuCS3OidIww8ZtZEkhQvU91QpmGSOAhWYonCdRvNaC\nnoqKGBDrFRU+7siK8+L0rea+lcn
1frdWjkYB//GzuhhqJ2zitxRo+Zukx1PAS59Gi\n2sAnWJTmAdw/eCyMJo2HG9R2dKmAtmMx+dgO9nNnB
0+FK0sEC1SHTNWaGALmX5rT\nn1IdDxeHUrjE5K/7dLC32qcqC1QIDAQABoAAwDQYJKoZIhvcNAQELBQAD
ggEBAABJ\nUpqzjZwgq+RV0ph33GMK2s2fyv4tbOQ+JwdsWFQRyVdBjEvxu9xD1/Ou9QN+4L2T\nnhXFs1
WLyDWYxZyCu+4vPVYLBXzEnOqZblilxilmJ8wykYk2dyttNPOQo22ZPLTHx\nnviMZ81qZu/oGGk4ia9YB
svjMLPsbFzW34xWLTiEYgiNDlPhsZxQbxOwQ6H2VsCAQ\n6c/cF0LkbZZvKunrNb//KnfF7VmPFhZQkdt
uagwAX/1LdymVGockFLviYcxTP6nO\n7xUzUVcBRAdZib+8dQYerNUW2+qAQeWFFLFqQrFG9xr8UWQt8S
hDWuTwZPvWnA9V\n8JmUnCIUuoaoJWZxP+Y=\n-----END CERTIFICATE REQUEST-----",
  "cert": "-----BEGIN CERTIFICATE-----\nMIIEIjCCAuagAwIBAgIQc7nFh3ZoAiv0UpnsQji
JiTANBgkqhkiG9w0BAQsFAADBa\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUQxEDAOBgNVBACTB0JlbG
NhbxAXAEDAO\nBgNVBAoTB0dlbWFSdG8xGjAYBgNVBAMTEUtleVn1Y3VyZSBSb290IENBMB4XDTE\nzMTEwNTUxNVowXDTIzMTAzMTEwNTUxNVowazEzMDEGA1UEAxMqdG9rZW4yMDk5\nYWI5MjktMTk0Zi00ZThi\nLTkwZWMTNTk1Mjc5NTgzYWNlMTQwMgYKCCZImiZPYLQGB\nARMkMWUwNGYxNWYtMGRlNy00ZTEzLTljN
WQtMmE4NGUyNmVlOTQ3MIIBIjANBgkq\nnhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsKKO2wJspkm8eeVJ
Z6r9aexy4Nk+7vla\nnMsVgTBjI2yZrtgbkbNFQah1cwKnyC3ZNF5X4USFGHFs+mSgoCEfU3leMQb7WS6
y\nnPsRgM4Z4E3WIu9bDNBp9MUXuSE1kz6uv9qbDLD9TVghkMIIU6wJuCS3OidIww8Zt\nnZEkhQvU91Qpm
GSOAhWYonCdRvNaCoqKGBDrFRU+7siK8+L0rea+lcn1frdWjkYB\n/n/GzuhhqJ2zitxRo+Zukx1PAS59G
i2sAnWJTmAdw/eCyMJo2HG9R2dKmAtmMx+dgO\n9nNnB0+FK0sEC1SHTNWaGALmX5rTlIdDxeHUrjE5K/
7dLC32qcqC1QIDAQABo4Gu\nnMIGrMA4GA1UdDwEB/wQEAwIDiDATBgNVHSUEDDAKBggrBgEFBQcDAjAMB
gNVHRMB\nAf8EAjAAMB8GA1UdIwQYMBaAFK5TxeYX0HEIee7MBusV6YZRlpiJMFUGA1UdHwRo\nnMEwwSq
BIoEaGRGH0dHA6Ly9rZX1zZW51cmUubG9jYWwvY3Jscy9jNTA2MjY4Zi0x\nnYzUzLTRiOWUtODk1ZS1iZ
WZjZjg0MTgyYjUuY3JSMa0GCSqGSIB3DQEBCwUAA4IC\nAQCVzKyj6GK9hmFdz3p6EdAp0WyItAPwTRZ3
zF/byj9aj\nnDnLfpU7IUjwleuoF9\nA38k1RPTouuydEIXZZ+Fj1/ymZvQo5Bx/4Kbhjoss5PDrbqoCI1
NXk3BRUXAKFxV\nnd+U8MXvYdXV/YjlxUNypMbOY6apWZnxbrNXw8MF+LUqHBaG7YknrmQq36DUJSUgi\nn

```

Upon reconnection, CipherTrust Manager automatically updates the client with the new certificate details.

Client Certificate Expiration Check

The CipherTrust Manager inspects the expiration date of all the registered client's certificates everyday, at a preset system time to log the record.

The CipherTrust Manager then creates list of certificates based on their expiration date:

- Certificates whose expiration dates are within 91 days.

This list is logged in the **Records** section once every week.

- Certificates whose expiration dates are within 7 days.

This list is logged in the **Records** section once every day.

- Certificates that are already expired.

This list is logged in the **Records** section once every day.

You can also create alarm triggers for these records. For more details, go to [Creating Alarm Trigger for Client Certificate Expiration](#).

Authenticating to the REST API

The REST API is hosted at this base URL: `https://{addr}/api/v1`

The specification is available [here](#).

You can use the REST interface via **curl**, or from the API Guide embedded in the CipherTrust Manager UI.

Authenticating to the Embedded API Guide

The embedded API Guide shares authentication with the CipherTrust Web UI. To authenticate to both the API Guide and the CipherTrust Web UI, visit `https://{ciphertrust_manager_IP_or_hostname}` in a web browser. Alternatively, click the **Log In** link in the top right corner of the API Guide when you are unauthenticated.

The simplest login is with a username and password. This authenticates a local user to the root domain. There can be additional login options to [authenticate to another domain](#), [authenticate with a certificate](#), [authenticate using LDAP](#), or [authenticate using OIDC](#). These options require additional user management configuration from a user in the User Admins group.

After login, the API Guide is accessible through the **API** button in the upper right corner.

There, you can make API calls in the established authentication context. You are logged out after 10 minutes of inactivity with the API Guide open. If you close the browser window, you are logged out after 30 minutes of your last login.

Manual API Token Generation for User Credentials

You can generate an API token with a user's username and password for a given domain, outside of the

embedded API guide. This is useful if you have created a custom REST API client.

To copy and paste the following example commands, set an environment variable to point to your CipherTrust Manager instance:

```
$ export KSCTL_URL=https://{addr} # sh/bash
$ set -x KSCTL_URL=https://{addr} # fish
```

For example, this command will use the root admin's credentials to create an API token:

```
$ curl -k -X POST $KSCCTL_URL/api/v1/auth/tokens/ \
-H "Content-Type: application/json" \
-d '{"name": "admin", "password": "admin"}'
```

Note

By default, this command creates a token for the root domain. You can specify a [child domain](#) to log in to with `{"name\":"domain_user\","password\":"domain_user_password\","domain\":"domain name\"}"`

The response:

```
{
  "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJkMzEzYzZlLTQ1YzEtOWU1OS0zYmUxNTRkMTZjYmQiLCJpc3MiOiJreWxvIiwiaWF0Ijoi2023-09-01T00:00:00Z",
  "duration": 300
}
```

Copy the value of the `jwt` property into another environment variable:

```
(sh) $ export KSTCL_JWT=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJkMzEzYzZwZS01NmYyLTQ5MTU0UGJkNy01ZmMyYmUyNzk2YzEiLCJzdBWIIiOiJsb2NhbmHxImZ2JjOGZlNy1hN2ZlLTQ1YzEtOWU1OS0zYmUxNTRkMTZjYmQiLCJpc3MiOiJreWxvIiwiaWYWNjIjoia3lsbyIsImNlc3QiOnsiZ3JvdXBzIjpbImFkbWluIl19LCJpYXQoIjE0ODExMjQxMzcsImV4cCI6MTQ0MTEyNDQzN30.Z9GU0YyOImHmDpmbZx66vL0NLQfeLvNGkyGkZpVbfx4
```

```
(fish)$ set -x KSTCL_JWT eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJkMzEzYzZwZS01NmYyLTQ5MTU0UGJkNy01ZmMyYmUyNzk2YzEiLCJzdBWIIiOiJsb2NhbmHxImZ2JjOGZlNy1hN2ZlLTQ1YzEtOWU1OS0zYmUxNTRkMTZjYmQiLCJpc3MiOiJreWxvIiwiaWYWNjIjoia3lsbyIsImNlc3QiOnsiZ3JvdXBzIjpbImFkbWluIl19LCJpYXQoIjE0ODExMjQxMzcsImV4cCI6MTQ0MTEyNDQzN30.Z9GU0YyOImHmDpmbZx66vL0NLQfeLvNGkyGkZpVbfx4
```

Using the **jq** tool, we can fetch, extract, and export the token in a single shell command:

```
(sh) $ export KSCTL_JWT=$(curl -k -X POST $KSCTL_URL/api/v1/auth/tokens/ \
-H "Content-Type: application/json" \
-d "{\"name\":\"admin\",\"password\":\"admin\"}"
```

```
n\"}" | jq -r '.jwt')
(fish)$ set -x KSCTL_JWT (curl -k -X POST $KSCTL_URL/api/v1/auth/tokens/ \
-H "Content-Type: application/json" \
-d "{\"name\":\"admin\", \"password\":\"admin\"}" | jq -r '.jwt')
```

We can use that API token to make other calls:

```
$ curl -k "$KSCTL_URL/api/v1/vault/keys2?limit=20" \
-H "Authorization: Bearer $KSCTL_JWT"
```

REST API Client Registration and API Token Generation for Client Credentials

You can register a REST API client to improve monitoring and traceability on the client's actions. Registered clients can exchange their client identities and client certificates for an API token for authentication through the REST API.

Registration is especially beneficial for automated clients which act without a person initiating operations.

If you register the REST client, its client identity is recorded in the [audit records](#) for any operations it performs using that identity.

Registering a REST API client

Registered clients require a client certificate which CipherTrust Manager recognizes. There are three options for creating a suitable certificate, in order of most to least secure:

- Provide a certificate signing request (CSR) issued by the client, to be signed by a CipherTrust Manager local certificate authority (CA).
- Sign a client-produced CSR with an external CA. The [external CA must be added](#) to CipherTrust Manager.
- If your client is not capable of generating a key pair or CSR, you can generate a CSR on CipherTrust Manager and use a local CA to sign it. You must then download the generated certificate and store it on the client.

The high level steps for registering are:

1. If possible, create a certificate signing request (CSRs) on the client.
2. Create a [client profile](#) on CipherTrust Manager.

The client profile controls the CA and certificate settings used to create or renew client certificates for a particular client.

Note

If you need CipherTrust Manager to generate a CSR, we recommend providing CSR parameters when creating the client profile, and then providing the resulting client certificate in the registration request. If you provide CSR parameters at both stages, the client certificate returned in the registration response takes precedence over the certificate returned on client profile creation.

3. Create a [registration token](#) on CipherTrust Manager using the client profile. You must retain the registration token.

The registration token is a randomly generated character string used to register a client.

4. With the REST client, make a `POST /v1/client-management/clients` to register the client.

Provide the registration token and certificate-related parameters. The specification file and API Guide embedded in the CipherTrust Manager GUI has full schema descriptions for this endpoint. The example request demonstrates providing a CSR for CipherTrust Manager to sign.

Example request

```
curl -k "$KSCTL_URL/api/v1/client-management/clients/" -H 'Content-Type: application/json' -H 'accept: application/json' --data-binary '${\n  "name": "REST_CLIENT",\n  "registration_token": "12345",\n  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIHNMHUCAQAwEzERMA8GA1UEAxMIVGVzdCBDU1IwWTATBgqhkhjOPQIBBgqhkhjO\n\\nPQMBBwNCAATndOIgsTp7m4bOuixxuAt2XQ3oZqp8th/woAo51z7RiAAGdm7IfB1w\n\\n7uWr8o5PWKBatXIgvPp8hvrWHQPHCfcLoAAwCgYIKoZIZj0EAwIDSAAwRQIgVyz\\nhFGCKV460fNJC0vC48gI268B68Xr6osFoy9Ouw8CIQCWN1LtcyxpIvul3XF1Pj7l\\navEeIqDBcfD6VHhbnpO2Ag==\n-----END CERTIFICATE REQUEST-----",\n  "subject_dn_field_to_modify": "DN Q",\n  "do_not_modify_subject_dn": false\n}' --compressed
```

Retain the ID value in the response. This represents the client identity required to obtain a token.

API Token Generation with Client Identity

To issue an API token for a client identity, you must specify `grant_type:"client_credential"` and `client_id:<client_identity>` in the `data` part of the request. The client identity is obtained during [client registration](#).

In addition, the public certificate file path and private key file path are provided in the request to ensure that the client certificate is verified. These values correspond with the client certificate configuration that took place when creating a client profile and registering the client.

For example, this command generates an API token for the client 123456789, provided the presented identity and client certificate match the expected values in the client registration:


```
$ curl -k -X POST $KSCCTL_URL/api/v1/auth/tokens/ -H "Content-Type: application/json" --cert ".ssh/id_rsa.pub" --key ".ssh/id_rsa" -d "{\"grant_type\":\"client_credentials\",\"client_id\":\"123456789\"}"
```

Note

Clients always login to the root domain.

The response:

```
{
  "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJkMzEzZjOGZlNy1hN2ZlLTQ1YzEtOWU1OS0zYmUxNTRkMTZjYmQiLCJpc3MiOiJreWxvIiwiaWF0IjoiY3I3JvdXBzIjpbImFkbWluI119LCJpYXQoIjE0ODExMjQ4MTEyNDQzN30uZ9GU0YyOImHMDPmbZx66vL0NLQfeLvNGkyGkZpVbfx4",
  "duration": 300
}
```

Copy the value of the jwt property into another environment variable:

```
(sh) $ export KSCCTL_JWT=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJkMzEzZjOGZlNy1hN2ZlLTQ1YzEtOWU1OS0zYmUxNTRkMTZjYmQiLCJpc3MiOiJreWxvIiwiaWF0IjoiY3I3JvdXBzIjpbImFkbWluI119LCJpYXQoIjE0ODExMjQ4MTEyNDQzN30uZ9GU0YyOImHMDPmbZx66vL0NLQfeLvNGkyGkZpVbfx4
(fish)$ set -x KSCCTL_JWT eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJkMzEzZjOGZlNy1hN2ZlLTQ1YzEtOWU1OS0zYmUxNTRkMTZjYmQiLCJpc3MiOiJreWxvIiwiaWF0IjoiY3I3JvdXBzIjpbImFkbWluI119LCJpYXQoIjE0ODExMjQ4MTEyNDQzN30uZ9GU0YyOImHMDPmbZx66vL0NLQfeLvNGkyGkZpVbfx4
```

We can use that API token to make other calls:

```
$ curl -k "$KSCCTL_URL/api/v1/vault/keys2?limit=20" -H "Authorization: Bearer $KSCCTL_JWT"
```

Attribute-based Access Control Permissions for Resources

This section provides the complete list of ABAC permissions required by a CipherTrust Manager user/client to perform operations on resources.

Resources

Backup	Certificate Authority	Client Management	Cluster
Keys	Connection Manager	Proxies	DNS Hosts
Domains	Groups	Interfaces	Connections
Logs	Log Forwarders	Quorum	Records
Users	Scheduler	Servers	SNMP
Policies	Migrations	Licensing	HSM
Miscellaneous	Network	ProtectApp	Secrets

Backup

Backup Keys

Operations	Required Permissions
create backup key	CreateBackupkey CreateBackupkeyDomain
List backup key	ReadBackupkey ReadBackupkeyDomain
Download backup key	DownloadBackupkey DownloadBackupkeyDomain
Upload backup key	UploadBackupkey UploadBackupkeyDomain

Backup

Operations	Required Permissions
create backup	CreateBackup CreateBackupDomain
list backup	ReadBackup ReadBackupDomain
Delete backup	DeleteBackup DeleteBackupDomain
Download backup	DownloadBackup DownloadBackupDomain
Restore backup	RestoreBackup RestoreBackupDomain
upload backup	UploadBackup UploadBackupDomain

Operations	Required Permissions
Backup Status	StatusBackup StatusBackupDomain

SCP Backup

Operations	Required Permissions
Get public key for SCP connection	DownloadSCPPublicKey
Rotate public key for SCP connection	RotateSCPKey
SCP the specified backup	DownloadBackup DownloadBackupDomain
SCP the specified backup to a host from specified connection	DownloadSCPPublicKey
Get the recent status of SCPs of a specific backup	No Permission needed
Get the status of the backup transferred via SCP of a given backup id	No permission needed

Certificate Authority

Operations	Required Permissions
Create CSR	GenerateCSR
Upload external CA	UploadExternalCA
List/Get external CA	ReadExternalCA
Delete external CA	ReadExternalCA DeleteExternalCA
Update external CA	UpdateExternalCA
Create local CA	CreateLocalCA
List/Get local CA	ReadLocalCA
Delete local CA	ReadLocalCA DeleteLocalCA
Update local CA	UpdateLocalCA
Self-sign a local CA	ReadLocalCA SelfSignLocalCA
Install a local CA	ReadLocalCa InstallLocalCA
Issue a certificate	IssueCertificate
List/Get certificates issued by local CA	ReadCertificate ReadLocalCA
Delete certificates issued by local CA	ReadCertificate

Operations	Required Permissions
	DeleteCertificate
Revoke certificates issued by local CA	ReadCertificate ReadLocalCA
Resume certificates local CA	ReadCertificate ReadLocalCA

Client Management

Clients

Operations	Required Permissions
Register a client	No permission needed
List/Get client	ReadClient
Update client	UpdateClient
Delete client	DeleteClient
Revoke client	UpdateClient
Renew client	ReadClient UpdateClient
Self client renewal	ReadClientSelf UpdateClientSelf
Add impersonated user	ReadClient UpdateClient
Delete impersonated user	ReadClient UpdateClient
Get impersonated users	ReadClient GetImpersonatedUser

Tokens

Operations	Required Permissions
Create regtoken	RegistrationTokenCreate
List/Get regtoken	RegistrationTokenRead
Update regtoken	RegistrationTokenRead RegistrationTokenUpdate
Delete regtoken	RegistrationTokenRead RegistrationTokenDelete
Get web certificate fingerprint	No Permission needed

Profiles

Operations	Required Permissions
Create client profile	AddGenericClientProfile
List/Get client profile	ReadGenericClientProfile
Update client profile	ReadGenericClientProfile UpdateGenericClientProfile
Delete client profile	ReadGenericClientProfile DeleteGenericClientProfile

KMIP Client Management

Operations	Required Permissions
List/Get kmip profiles	ReadKmpClientProfile
Create kmip profile	CreateKmpClientProfile
Delete kmip profile	DeleteKmpClientProfile
Register kmip client	No permission needed
List kmip clients	ReadKmpClient
Get clients count	ReadKmpClientCount
Delete kmip client	DeleteKmpClient
Create regtokens	No permission needed

Cluster

Operations	Required Permissions
Get cluster info	ReadCluster
Delete cluster	ReadCluster DeleteClusterNode
Create cluster	CreateCluster
Create cluster CSR	CreateClusterCSR
Join cluster	JoinCluster
List cluster errors	ReadCluster
Get cluster summary	ReadCluster

Cluster Nodes

Operations	Required Permissions
List/Get cluster nodes	ReadClusterNode
Sign certificate	CreateClusterNode
Update cluster nodes	ReadClusterNode UpdateClusterNode
Delete Cluster nodes	ReadClusterNode DeleteClusterNode

Keys

Operations	Required Permissions
List/Get keys	ReadKey
Create key	CreateLink ReadKey UseKey (for internal requests) ExportKey CreateKey
Update key	UpdateKey UpdateKeyAppMeta ReadKey
Delete key	DeleteKey ReadKey
List versions of a key	ReadKey
Create version of a key	CreateKeyVersion ExportKey
Destroy a key	DeleteKey ReadKey
Archive a key	ArchiveKey ReadKey
Recover a key	RecoverKey ReadKey
Revoke a key	RevokeKey ReadKey
Reactivate a key	ReActivateKey ReadKey
Export a key	ExportKey ReadKey
Clone a key	CloneKey UseKey ReadKey

Operations	Required Permissions
List key labels	ReadKey
Query keys	ReadKey

MKEK

Operations	Required Permissions
List/Get Mkeys	ReadMkek
Rotate Mkek	RotateMkek

Root of Trust Keys

Operations	Required Permissions
List/Get rot-keys	ReadRoTKey
Delete rot-keys	DeleteRoTKey
Rotate rot-keys	RotateRoTKey

SSH Keys

Operations	Required Permissions
Add ssh keys	No permission needed

Connection Managers

AWS Connections

Operations	Required Permissions
Create AWS connection	CreateConnectionPermission
List/Get AWS connection	ReadConnectionPermission
Delete AWS connection	DeleteConnectionPermission
Update AWS connection	UpdateConnectionPermission
Test existing AWS connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

Azure Connections

Operations	Required Permissions
Create Azure connection	CreateConnectionPermission
List/Get Azure connection	ReadConnectionPermission
Delete Azure connection	DeleteConnectionPermission
Update Azure connection	UpdateConnectionPermission
Test existing Azure connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

CM Connections

Operations	Required Permissions
Create CM connection	CreateConnectionPermission
List/Get CM connection	ReadConnectionPermission
Delete CM connection	DeleteConnectionPermission
Update CM connection	UpdateConnectionPermission
Test existing CM connection	TestConnectionPermission
Test CM connection parameters	TestConnectionPermission

GCP Connections

Operations	Required Permissions
Create GCP connection	CreateConnectionPermission
List/Get GCP connection	ReadConnectionPermission
Delete GCP connection	DeleteConnectionPermission
Update GCP connection	UpdateConnectionPermission
Test existing GCP connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

DSM Connections

Operations	Required Permissions
Create DSM connection	CreateConnectionPermission
List/Get DSM connection	ReadConnectionPermission
Delete DSM connection	DeleteConnectionPermission
Update DSM connection	UpdateConnectionPermission

Operations	Required Permissions
Test existing DSM connection	TestConnectionPermission ReadConnectionPermission
Test connection parameters	TestConnectionPermission
Add a new DSM node	CreateConnectionPermission ReadConnectionPermission
List/Get DSM nodes	ReadConnectionPermission ReadConnectionPermission
Delete DSM node	ReadConnectionPermission,DeleteConnectionPermission
Update DSM node	UpdateConnectionPermission

Hadoop Connections

Operations	Required Permissions
Create Hadoop connection	CreateConnectionPermission
List/Get Hadoop connection	ReadConnectionPermission
Delete Hadoop connection	DeleteConnectionPermission
Update Hadoop connection	UpdateConnectionPermission
Test existing Hadoop connection	TestConnectionPermission ReadConnectionPermission
Test connection parameters	TestConnectionPermission
Add a new Hadoop node	CreateConnectionPermission ReadConnectionPermission
List/Get Hadoop nodes	ReadConnectionPermission ReadConnectionPermission
Delete Hadoop node	ReadConnectionPermission DeleteConnectionPermission
Update Hadoop node	UpdateConnectionPermission

Elasticsearch Connections

Operations	Required Permissions
Create Elasticsearch connection	CreateConnectionPermission
List/Get Elasticsearch connection	ReadConnectionPermission
Delete Elasticsearch connection	DeleteConnectionPermission
Update Elasticsearch connection	UpdateConnectionPermission
Test existing Elasticsearch connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

External CM Server

Operations	Required Permissions
Create External CM connection	CreateConnectionPermission
List/Get External CM connection	ReadConnectionPermission
Delete External CM connection	ReadConnectionPermission DeleteConnectionPermission
Add a new CM node	ReadConnectionPermission UpdateConnectionPermission
List/Get CM nodes	ReadConnectionPermission
Delete CM node	ReadConnectionPermission UpdateConnectionPermission
Update CM node	ReadConnectionPermission UpdateConnectionPermission
Add a trusted CA	ReadConnectionPermission UpdateConnectionPermission
Delete trusted CA details	UpdateConnectionPermission

LDAP Connections

Operations	Required Permissions
Create LDAP connection	CreateConnectionPermission
List/Get LDAP connection	ReadConnectionPermission
Delete LDAP connection	DeleteConnectionPermission
Update LDAP connection	UpdateConnectionPermission
Test existing LDAP connection	TestConnectionPermission ReadConnectionPermission
Test connection parameters	TestConnectionPermission

Loki Connections

Operations	Required Permissions
Create Loki connection	CreateConnectionPermission
List/Get Loki connection	ReadConnectionPermission
Delete Loki connection	DeleteConnectionPermission
Update Loki connection	UpdateConnectionPermission
Test existing Loki connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

Luna Network HSM Connections

Operations	Required Permissions
Create Luna network HSM connection	CreateConnectionPermission
List/Get Luna network connection	ReadConnectionPermission
Delete Luna network connection	ReadConnectionPermission DeleteConnectionPermission
Update Luna network connection	UpdateConnectionPermission
Test existing Luna network connection	ReadConnectionPermission TestConnectionPermission
Test connection parameters	ReadConnectionPermission TestConnectionPermission
Add a new Partition into Luna network	ReadConnectionPermission UpdateConnectionPermission
Delete a partition node	ReadConnectionPermission UpdateConnectionPermission
Get connection status	ReadConnectionPermission

Luna Network HSM STC Partition

Operations	Required Permissions
Register a new Luna Network HSM STC partition	CreateConnectionHSMSTCPartitionPermission
List/Get STC partitions	ReadConnectionHSMSTCPartitionPermission
Delete STC partitions	ReadConnectionHSMSTCPartitionPermission DeleteConnectionHSMSTCPartitionPermission

Luna Network HSM Servers

Operations	Required Permissions
Create Luna network HSM Servers connection	CreateConnectionHSMServerPermission
List/Get Luna HSM Servers	ReadConnectionHSMServerPermission
Delete Luna HSM Servers	DeleteConnectionHSMServerPermission
Force Delete Luna HSM Server	DeleteConnectionHSMServerPermission
Enable STC for Luna HSM Server	ReadConnectionHSMServerPermission UpdateConnectionHSMServerPermission
Disable STC for Luna HSM Server	ReadConnectionHSMServerPermission UpdateConnectionHSMServerPermission
Add a new Partition into Luna network	
Get Info about Luna client	

OIDC Connections

Operations	Required Permissions
Create OIDC connection	CreateConnectionPermission
List/Get OIDC connection	ReadConnectionPermission
Update OIDC connection	UpdateConnectionPermission
Delete OIDC connection	DeleteConnectionPermission

Oracle Connections

Operations	Required Permissions
Create Oracle connection	CreateConnectionPermission
List/Get Oracle connection	ReadConnectionPermission
Delete Oracle connection	DeleteConnectionPermission
Update Oracle connection	UpdateConnectionPermission
Test existing Oracle connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

SAP Data Custodian Connections

Operations	Required Permissions
Create SAP Data Custodian connection	CreateConnectionPermission
List/Get SAP Data Custodian connection	ReadConnectionPermission
Delete SAP Data Custodian connection	DeleteConnectionPermission
Update SAP Data Custodian connection	UpdateConnectionPermission
Test existing SAP Data Custodian connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

SCP Connections

Operations	Required Permissions
Create SCP connection	CreateConnectionPermission
List/Get SCP connection	ReadConnectionPermission
Delete SCP connection	DeleteConnectionPermission
Update SCP connection	UpdateConnectionPermission
Test existing SCP connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

SMB Connections

Operations	Required Permissions
Create SMB connection	CreateConnectionPermission
List/Get SMB connection	ReadConnectionPermission
Delete SMB connection	DeleteConnectionPermission
Update SMB connection	UpdateConnectionPermission
Test existing SMB connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

Salesforce Connections

Operations	Required Permissions
Create Salesforce connection	CreateConnectionPermission
List/Get Salesforce connection	ReadConnectionPermission
Delete Salesforce connection	DeleteConnectionPermission
Update Salesforce connection	UpdateConnectionPermission
Test existing Salesforce connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

Syslog Connections

Operations	Required Permissions
Create Syslog connection	CreateConnectionPermission
List/Get Syslog connection	ReadConnectionPermission
Delete Syslog connection	DeleteConnectionPermission
Update Syslog connection	UpdateConnectionPermission
Test existing Syslog connection	TestConnectionPermission
Test connection parameters	TestConnectionPermission

CSR Creation

Operations	Required Permissions
Create connection CSR	CreateConnectionsCSR

Proxies & Allowed Proxies

Proxies

Operations	Required Permissions
Set proxy	AddProxy
Update proxy	UpdateProxy
Get proxy	ReadProxy
Delete proxy	ReadProxy DeleteProxy
Test proxy	TestProxy

Allowed Proxies

Operations	Required Permissions
Whitelist proxy (POST)	AddProxyProtocolAllowedProxies
List/Get whitelisted proxies	ReadProxyProtocolAllowedProxies
Delete whitelisted proxy	ReadProxyProtocolAllowedProxies DeleteProxyProtocolAllowedProxies
Update Whitelisted proxy	ReadProxyProtocolAllowedProxies UpdateProxyProtocolAllowedProxies
Reset proxies	ResetProxyProtocolAllowedProxies

DNS Hosts

Operations	Required Permissions
Create DNS host entry	CreateDNSHost
List/Get DNS host entries	ReadDNSHost
Update a DNS host entry	ReadDNSHost UpdateDNSHost
Delete a DNS entry	ReadDNSHost DeleteDNSHost

Domains

Operations	Required Permissions
List/Get domains	ReadDomain
Create domains	CreateDomain

Operations	Required Permissions
Update domain	ReadDomain UpdateDomain
Delete domain	DeleteDomain
List/Get domain KEKs	ReadDomainKEK
Rotate KEK	RotateDomainKEK
Retry domain KEK rotation	RotateDomainKEK
Enable syslog messages redirection (legacy syslog)	EnableDisableSyslogRedirection
Disable syslog messages redirection (legacy syslog)	EnableDisableSyslogRedirection
Syslog messages redirection status (legacy syslog)	ReadSyslogRedirectionStatus
Enable log forwarder redirection	EnableDisableDomainLogRedirection
Disable log forwarder redirection	EnableDisableDomainLogRedirection
Log forwarder redirection status	ReadDomainLogRedirectionStatus
Get domain info of current logged in user	ReadDomain

Groups & Groupmaps

Groups

Operations	Required Permissions
List/Get groups	ReadGroup
Create group	CreateGroup
Delete group	ReadGroup DeleteGroup
Update group	ReadGroup UpdateGroup
Add a user to a existing group	AddUserToGroup
Remove user from a group	ReadUser ReadGroup RemoveUserFromGroup
Add a client to group	ReadClient ReadGroup AddClient
Remove client from group	ReadClient ReadGroup RemoveClient

GroupMaps

Operations	Required Permissions
List/Get groupmaps	ReadGroupMapping
Create groupmaps	CreateGroupMapping
Delete groupmaps	ReadGroupMapping DeleteGroupMapping
Update groupmap	ReadGroupMapping UpdateGroupMapping

Interfaces

Operations	Required Permissions
List/Get interfaces	ReadInterfaceConfiguration
List/Get interfaces	ReadInterfaceConfiguration
Add an interface	CreateInterfaceConfiguration
Update interface	UpdateInterfaceConfiguration
Delete interface	DeleteInterfaceConfiguration
Get interface's certificate	ReadInterfaceCertificate
Put a certificate on interface	ReplaceInterfaceCertificate
Enable interface	UpdateInterfaceConfiguration
Disable interface	UpdateInterfaceConfiguration
Restore interface TLS Ciphers	UpdateInterfaceConfiguration
Get interface's CSR	GetInterfaceCSR
Generate CSR for an interface	GenerateInterfaceCSR
Auto-generate a server certificate for an interface	ReadLocalCA GenerateCSR IssueCertificate
Copy interface's certificate	ReadInterfaceCertificate
Get upcoming certificate	ReadUpcomingInterfaceCertificate
Put upcoming certificate	StoreUpcomingInterfaceCertificate
Delete upcoming certificate	DeleteUpcomingInterfaceCertificate
Apply upcoming certificate	ReplaceInterfaceCertificate

Connections

Operations	Required Permissions
List/Get connection	GetLDAPConnection
Test connection	CreateLDAPConnection
Create connection	CreateLDAPConnection
Delete connection	GetLDAPConnection DeleteLDAPConnection
Update connection	GetLDAPConnection UpdateLDAPConnection
Refresh connection	GetLDAPConnection UpdateLDAPConnection
List/Get users of Connection	ReadUser
List users LDAP	ReadLdapUser
List groups LDAP	ReadLdapGroup

Syslog Connections

Operations	Required Permissions
List/Get syslogs	ReadSyslogConnection
Add syslog connection	CreateSyslogConnection
Delete syslog connection	ReadSyslogConnection DeleteSyslogConnection
Update syslog connection	ReadSyslogConnection UpdateSyslogConnection

Logs

Operations	Required Permissions
Log download	DownloadClientAuditLogs DownloadServerAuditLogs DownloadActivityLogs DownloadDebugLogs
Download nae activity logs	DownloadActivityLogs
Download KMIP activity logs	DownloadActivityLogs
Download Web activity logs	DownloadActivityLogs
Download debug logs	DownloadDebugLogs
Download all logs	DownloadLogs
Get log level	No permission needed

Operations	Required Permissions
Set log level	UpdateLogLevel

Log Forwarders

Operations	Required Permissions
List/Get log forwarders	ReadLogForwarderConnection DeleteLogForwarderConnection
Add log forwarder	CreateLogForwarderConnection
Delete log forwarder	ReadLogForwarderConnection UpdateLogForwarderConnection
Update log forwarder	ReadLogForwarderConnection UpdateLogForwarderConnection

Loki Configuration

Operations	Required Permissions
Get loki config	ReadLokiConfiguration
Update loki config	UpdateLokiConfiguration

Prometheus Metrics

Operations	Required Permissions
Get metrics	No permission needed
Enable metrics collection	Enable Prometheus Metrics Collection
Disable metrics collection	Disable Prometheus Metrics Collection
Renew metrics collection token	Renew Prometheus Metrics API token
get status	Get Prometheus Metrics Status

Quorum

Operations	Required Permissions
List quorums	VoteQuorum ReadQuorum
Delete a quorum	ReadQuorum DeleteQuorum
Activate a quorum	ReadQuorum ActivateQuorum,

Operations	Required Permissions
Approve a quorum	ReadQuorum VoteQuorum
Deny a quorum	ReadQuorum VoteQuorum
Revoke the vote cast for the quorum	ReadQuorum
List resources of the quorum	ReadQuorum
Activate a quorum policy	ReadPolicy CreatePolicy CreatePolicyAppMeta ReadAttachedPolicy CreateAttachedPolicy CreatePolicyAppMeta
Deactivate quorum policy	ReadPolicy DeletePolicy
Status quorum policy	ReadAttachedPolicy ReadPolicy
List/Get quorum profiles	ReadQuorumProfiles
Update quorum profile	UpdateQuorumProfile ReadAttachedPolicy DeleteAttachedPolicy CreateAttachedPolicy CreatePolicyAppMeta
Get quorum	ReadQuorum

Records

Operations	Required Permissions
List/Get records	ReadRecord
List/Get client records	ReadClientRecord
List loki audit records	ReadRecord
Create alarm configuration	CreateRecordAlarmConfig
List alarm configuration	ListRecordAlarmConfigs
Update alarm configuration	ReadRecordAlarmConfig UpdateRecordAlarmConfig
Delete alarm configuration	ReadRecordAlarmConfig UpdateRecordAlarmConfig
get alarm configuration	ReadRecordAlarmConfig DeleteRecordAlarmConfig

Users

Operations	Required Permissions
List/Get users	ReadUser
Create user	CreateUser
Delete user	ReadUser DeleteUser
Update user	ReadUser UpdateUser
Get current user's info	ReadUser
Update current user	UpdateUser
Change password	No permission needed
change global password policy	UpdatePasswordPolicy
Get global password policy	No permission needed
List password policies	ReadPasswordPolicy
Create a password policy	CreatePasswordPolicy
change password policy	ReadPasswordPolicy UpdateCustomPasswordPolicy
Get password policy	ReadPasswordPolicy
Delete password policy	DeletePasswordPolicy
Get password policy of current logged in user	ReadSelfPasswordPolicy

Scheduler

Operations	Required Permissions
List/Get job configurations	ReadJob
Create job configurations	CreateJob
Update job configurations	UpdateJob
Delete job configurations	DeleteJob
Run now a job configuration	CreateJob
List/Get jobs	ReadJob
Delete job	DeleteJob
List supported operation types	No permission needed

Servers

SMTP Servers

Operations	Required Permissions
List/Get SMTP servers	ReadSMTPServer
Add SMTP server	AddSMTPServer ReadSMTPServer UpdateSMTPServer
Delete SMTP server	DeleteSMTPServer
SMTP test mail	ReadSMTPServer

NTP Servers

Operations	Required Permissions
List/Get NTP servers	ReadNTP
Add NTP server	CreateNTP
Delete NTP server	ReadNTP DeleteNTP
Get status NTP server	ReadNTP

SNMP

Operations	Required Permissions
Get SNMP info attributes	No permission needed
List the SNMP communities	ReadSnmpConfiguration
Add SNMP communities	AddSnmpConfiguration
Get SNMP community configuration info	ReadSnmpConfiguration
Update SNMP community configuration	UpdateSnmpConfiguration
Delete SNMP community configuration	DeleteSnmpConfiguration
List/Get SNMP user's configuration info	ReadSnmpConfiguration
add SNMP users	AddSnmpConfiguration
Update SNMP user	UpdateSnmpConfiguration
Delete SNMP user	DeleteSnmpConfiguration
List/get SNMP management stations	ReadSnmpConfiguration
Add SNMP management stations	AddSnmpConfiguration
Update management stations	UpdateSnmpConfiguration

Operations	Required Permissions
Delete management stations	DeleteSnmpConfiguration

Policies

Key Policies

Operations	Required Permissions
List/Get key policies	ReadKeyPolicy
Create a key policy	CreateKeyPolicy
Update a key policy	ReadKeyPolicy UpdateKeyPolicy
Delete a key policy	ReadKeyPolicy DeleteKeyPolicy

Policies

Operations	Required Permissions
List/Get policies	ReadPolicy
Create policies	CreatePolicy CreatePolicyAppMeta
Delete policy	ReadPolicy DeletePolicy

Policy Attachments

Operations	Required Permissions
List/Get policy attachments	ReadAttachedPolicy
Attach policy	CreateAttachedPolicy CreatePolicyAppMeta
Detach policy	ReadAttachedPolicy DeleteAttachedPolicy

Migrations

Operations	Required Permissions
Upload migrations	MigrationsUpload
List/Get migrations	MigrationsList
Delete migration	MigrationsDelete

Operations	Required Permissions
Migrate a migration	MigrationsMigrate
Migration data download	MigrationsDownload
Generate and upload DSM key migration file	MigrationsDownload
Get log of ongoing or last migration	MigrationsLog
Get status of ongoing or last migration	MigrationsStatus
Get containers from migration file	MigrationsGet

Migration Split Keys

Operations	Required Permissions
Create migration split keys	MigrationSplitKeysCreate MigrationSplitKeysList
List/get migration split keys	MigrationSplitKeysList
Delete migration split key	MigrationSplitKeysDelete
Add a share	MigrationSplitKeysPatch
Delete a share	MigrationSplitKeysPatch
Modify a share	MigrationSplitKeysPatch

Licensing

Operations	Required Permissions
List/Get licenses	ReadLicense
Add license	AddLicense
Delete license	ReadLicense DeleteLicense
List features of all licenses	ReadLicense
List available trials	ReadTrial_License
Get trial by id	ReadTrial_License
Activate a trial	ActivateTrial_License
Deactivate a trial	DeactivateTrial_License

HSM

HSM Servers

Operations	Required Permissions
Setup to HSM server	CreateHSM
Add HSM server	CreateHSM
List/Get HSM servers	ReadHSM
Delete HSM server	ReadHSM DeleteHSM

HSM Clients

Operations	Required Permissions
Get Luna STC client identity file	ReadHSM

Miscellaneous

Banners

Operations	Required Permissions
Get pre-auth banner	No permission needed
set pre-auth banner	UpdateLoginBanner
get post-auth banner	ReadPostAuthLoginBanner
set post-auth banner	UpdateLoginBanner

Alarms

Operations	Required Permissions
List alarms	ListAlarms
Clear alarms	ClearAlarms
Acknowledge alarms	AcknowledgeAlarms

Services

Operations	Required Permissions
Reset services	ResetServices
Restart services	RestartServices

Operations	Required Permissions
Status of services	No permission needed

Templates

Operations	Required Permissions
List/Get templates	ReadKey
Create templates	CreateKey
Update template	UpdateKey UpdateKeyAppMeta
Delete template	DeleteKey

Properties

Operations	Required Permissions
List/Get properties	ReadProperties
Update properties	UpdateProperties
Reset properties	ResetProperties

Notifications

Operations	Required Permissions
List/Get email addresses	ReadEmailAddress
Add email address	AddEmailAddress
Delete email address	AddEmailAddress

Query Permissions

Operations	Required Permissions
Query permissions	No permissions needed

Links

Operations	Required Permissions
List/Get links	ReadLink
Create links	CreateLink
Update link	UpdateLink
Delete link	DeleteLink

Info

Operations	Required Permissions
Get system info	ReadSystemInfo
Set system info	UpdateSystemInfo

Identity Providers

Operations	Required Permissions
List identity providers associated with CM	No permission needed
Login (post)	No permission needed
Login (Get)	No permission needed
oidc-callback (post)	No permission needed

Disc Encryption

Operations	Required Permissions
Get encryption status of a server	No permission needed
Encrypt Server on next reboot	No permission needed

Crypto

Operations	Required Permissions
Format preserving encrypt	ReadKey,UseKey,EncryptWithKey
Format preserving decrypt	ReadKey,UseKey,DecryptWithKey
Encrypt	ReadKey UseKey EncryptWithKey
Decrypt	ReadKey UseKey DecryptWithKey
Re-encrypt	ReadKey UseKey EncryptWithKey DecryptWithKey
MAC	ReadKey UseKey MACWithKey
MAC verify	ReadKey UseKey MACVerifyWithKey

Operations	Required Permissions
Sign	ReadKey UseKey SignWithKey
Sign verify	ReadKey UseKey SignVerifyWithKey
Encryptonite	ReadKey UseKey EncryptWithKey
Decryptonite	ReadKey UseKey DecryptWithKey
Generate Hash	No permission needed
Random	No permission needed

Network

Operations	Required Permissions
Ping a host	PingHostOrAddress
Trace-routing a host	TracerouteHostOrAddress
Check port availability on a remote host	CheckportHostOrAddress
List/Get network interfaces	ListNetworkInterfaces
Update network interface	ModifyNetworkInterface
DNS lookup query	Lookup Network Name or Address

ProtectApp

Operations	Required Permissions
List/Get ProtectApp profiles	ReadProtectAppClientProfile
Create ProtectApp profile	CreateProtectAppClientProfile
Delete ProtectApp profile	DeleteProtectAppClientProfile
Register ProtectApp client	RegisterProtectAppClient
List/Get ProtectApp clients	ReadProtectAppClient
Delete ProtectApp client	ReadProtectAppClient DeleteProtectAppClient

Secrets

Operations	Required Permissions
List/Get secrets	ReadKey
Create secrets	CreateKey
Export secrets	ExportKey
Update secret	UpdateKey UpdateKeyAppMeta
Delete secret	DeleteKey
Destroy Secret	DeleteKey

Password Policy

CipherTrust Manager local **users** have password policies. The CipherTrust Manager provides one universal password policy for all local **users** across all domains. This policy is known as the *global policy* and is applied to all users by default.

Note

- The *global policy* cannot be deleted.
- Password policies do not apply to users managed through an LDAP or OIDC connection.

However, the CipherTrust Manager also facilitates members of the **admin** and **user admin** groups to create custom password policies for new or existing users. These policies can be assigned to the selected users based on the requirement.

Custom policies are specific to the **domain** they are created in. You can only view, assign, or edit a custom policy if you are logged into its domain.

Allowed password length and characters

These are the password policy settings that define the allowed password length and characters:

- Default: Minimum number of characters: 8
- Default: Maximum number of characters: 30
- Default: Minimum number of uppercase characters: 1
- Default: Minimum number of lowercase characters: 1
- Default: Minimum number of digits: 1

- Default: Minimum number of other characters: 1

"Other characters" includes any character supported by the user's keyboard besides letters and digits. Regional keyboard characters, such as currency symbols, are allowed.

Password history

The CipherTrust Manager retains the user's password history to prevent users from reusing their passwords.

Default: Password History: 5.

- The minimum value for history is '0'. This value prevents the users from reusing their current password.
- With the default value of 5, the user is prevented from reusing their current password and 4 previous passwords.
- The maximum value for password history is 20.

User account lockout thresholds

User's account can be temporarily locked out for a specified duration after a specified number of user authentication failure attempts. By default, there is no lockout. A lockout is applicable only for users in a local account. It is not valid for LDAP or OIDC users.

Password Expiration

User passwords can be set to expire after the configured number of days from the last password change. By default, the password never expires, indicated with a password lifetime set to 0.

In addition, you can set a password expiry reminder during a defined period of time. With this setting, when a user's password is close to expiry, a banner displays on user login to the CipherTrust Manager GUI. The banner indicates the number of days until password expiry.

If an [SMTP](#) server is configured and the user has a non-default, valid email address, CipherTrust Manager also emails a password expiry reminder to the user. The timing and frequency of the email reminders depends on the user password expiry [scheduled job](#). By default, the scheduled job sends emails every day at 12 AM UTC. The scheduled job attempts to send emails to every local user in every domain that has these password policy settings, that is within the reminder period before password expiry. If the job cannot send an email to a user, it generates a [server record](#) in the user's domain. A record is generated, for example, when the email is undeliverable due to the user having an invalid email address.

There are two components to the password expiry reminder setting: enabling the reminder, and if enabled, setting the time period for CipherTrust Manager to display the banner and send emails. In the UI, the time period is referred to as **Days from expiration date**. Valid values for this time period are between 0 and 30, defaulting to 14 for new password policies. A value of 0 means that no reminder is displayed or sent.

Note

The password expiry reminder feature was introduced in CipherTrust Manager 2.15. If you upgrade from a CipherTrust Manager version lower than 2.15, existing password policies are given a **Days from**

expiration date value of 0.

Password Policy Management through the Web Console UI

You can create and delete custom password policies through the UI. You can also edit several settings of both the global password policy and custom password policies, and assign a password policy to a user.

Edit Valid Password Values

1. Log into CipherTrust Manager GUI as an Admin with user management permissions.

Note

As custom password policies are domain-specific, you might have to enable **I am a domain user** and specify a **Home Domain** as you log in, to access these resources in a non-root domain.

2. Navigate to **Admin Settings > Password Policies**.
3. Click the name of the policy you wish to view or edit.
4. Edit the following available settings as desired.
 - Minimum number of characters
 - Minimum lowercase characters
 - Minimum number of digits
 - Maximum number of characters
 - Minimum uppercase characters
 - Minimum number of special characters - This includes any character supported by the user's keyboard besides letters and digits. Regional keyboard characters, such as currency symbols, are allowed.
 - Unlimited characters - when enabled, you cannot set a maximum number of characters.
 - Password History Threshold - determines the number of previous passwords that a user cannot reuse. See [password history](#) for details and examples.
5. (Optional) Enter a test password in the **Test Password** field to validate potential passwords.
6. Click **Update Password Policy** to apply the changes.

Edit Account Lockout Settings

1. Log into CipherTrust Manager GUI as an Admin with user management permissions.

Note

As custom password policies are domain-specific, you might have to enable **I am a domain user** and specify a **Home Domain** as you log in, to access these resources in a non-root domain.

2. Navigate to **Admin Settings > Password Policies**.
3. Click the name of the policy you wish to view or edit.
4. Edit the following available settings as desired.
 - Disable account lockout
 - Number of failed attempts allowed before lockout
 - Lockout time in minutes
 - Add Lockout - This setting allows you to specify different lockout durations in minutes for subsequent failed attempts. For example, if the number of failed attempts allowed before lockout is 4, and the lockout time for the 5th failed attempt is 1 minute, you can add another lockout for the 6th attempt, with a different duration. Click the x next to the last Lockout to remove it. There is no limit on the number of lockout durations.
5. Click **Update Password Policy** to apply the changes.

Edit Password Expiry Settings

1. Log into CipherTrust Manager GUI as an Admin with user management permissions.

Note

As custom password policies are domain-specific, you might have to enable **I am a domain user** and specify a **Home Domain** as you log in, to access these resources in a non-root domain.

2. Navigate to **Admin Settings > Password Policies**.
 3. Click the name of the policy you wish to view or edit.
 4. Edit the following available settings as desired.
-

- Days before password expire - 0 means passwords never expire.
- Send password expiry reminder - Enable to email periodic [password expiry reminders](#) reminders to users with this password policy, and to display a banner on the CipherTrust Manager GUI during specified time period.

Note

An [SMTP](#) server must be configured for CipherTrust Manager to send emails. As well, the timing and frequency of the email reminders depends on the user password expiry [scheduled job](#). By default, emails are sent every day at 12 AM UTC.

- Days from expiration date - This setting indicates when CipherTrust Manager will start displaying and sending password expiry reminders. This value must be lower than the **Days before password expiry**. Valid values are between 0 and 30, defaulting to 14 for new password policies.

5. Click **Update Password Policy** to apply the changes.

Edit Allowed Password Change Frequency

1. Log into CipherTrust Manager GUI as an Admin with user management permissions.

Note

As custom password policies are domain-specific, you might have to enable **I am a domain user** and specify a **Home Domain** as you log in, to access these resources in a non-root domain.

2. Navigate to **Admin Settings > Password Policies**.
3. Click the name of the policy you wish to view or edit.
4. Edit the **Change password in minimum days** value. This indicates the minimum amount of time that must pass between a user's password changes.
5. Click **Update Password Policy** to apply the changes.

Create a Custom Password Policy

The only required value to create a password policy is a name. If desired, you can keep the defaults for [allowed password length and characters](#), [password history](#) (5 passwords), [account lockout](#) (no lockout), [password expiration](#) (no expiry), or time between password changes (1 day).

1. Log into CipherTrust Manager GUI as an Admin with user management permissions.

Note

As custom password policies are domain-specific, ensure you log in to the domain where you wish to access and apply the new custom password policy in the future. To create a custom policy in a non-root domain, enable **I am a domain user** and specify a **Home Domain** as you log in.

2. Navigate to **Admin Settings > Password Policies**.
3. Click **+ Create Password Policy**.
4. Enter a **Password Policy Name**.
5. Adjust the valid password values as desired.
 - Minimum number of characters
 - Minimum lowercase characters
 - Minimum number of digits
 - Maximum number of characters
 - Minimum uppercase characters
 - Minimum number of special characters - This includes any character supported by the user's keyboard besides letters and digits. Regional keyboard characters, such as currency symbols, are allowed.
 - Unlimited characters - when enabled, you cannot set a maximum number of characters.
 - Password History Threshold - determines the number of previous passwords that a user cannot reuse. See [password history](#) for details and examples.
6. (Optional) Enter a test password in the **Test Password** field to validate potential passwords.
7. Adjust Account Lockout Settings as desired.
 - Enable account lockout by unchecking the **Disable account lockout** checkbox.
 - Number of failed attempts allowed before lockout
 - Lockout time in minutes
 - Add Lockout - This setting allows you to specify different lockout durations in minutes for subsequent failed attempts. For example, if the number of failed attempts allowed before lockout is 4, and the lockout time for the 5th failed attempt is 1 minute, you can add another lockout for the 6th attempt, with a different duration. Click the x next to the last Lockout to remove it. There is no limit on the number of lockout durations.

8. Adjust password expiry settings, if desired.

- Days before password expire - 0 means passwords never expire
- Send password expiry reminder - Enable to email periodic [password expiry reminders](#) reminders to users with this password policy, and to display a banner on the CipherTrust Manager GUI during specified time period.

Note

An [SMTP](#) server must be configured for CipherTrust Manager to send emails. As well, the timing and frequency of the email reminders depends on the user password expiry [scheduled job](#). By default, emails are sent every day at 12 AM UTC.

- Days from expiration date - This setting indicates when CipherTrust Manager will start displaying and sending password expiry reminders. This value must be lower than the **Days before password expiry**. Valid values are between 0 and 30, defaulting to 14 for new password policies.

9. Edit the **Change password in minimum days** value, if desired. This indicates the minimum amount of time that must pass between a user's password changes.

10. Click **Create Password Policy** to finalize.

Delete a Custom Password Policy

Note

- You can only delete custom password policies. You cannot delete the global password policy.
- After you delete a custom password policy, all users with the policy are re-assigned to the global password policy.

1. Log into CipherTrust Manager GUI as an Admin with user management permissions.

Note

As custom password policies are domain-specific, you might have to enable **I am a domain user** and specify a **Home Domain** as you log in, to access these resources in a non-root domain.

2. Navigate to **Admin Settings > Password Policies**.
3. Find the name of the custom password policy, click the associated overflow icon (), and select **Delete**.

A confirmation pops up.

4. Read the confirmation, and click **Delete**.

Assign a Password Policy to a User

Local users are assigned a password policy on [creation](#) through the **Password Policy** setting. By default, they are assigned the global password policy.

You can re-assign an existing user to a different password policy. This requires a password change for the user, to ensure the password conforms to the assigned password policy.

1. Log into CipherTrust Manager GUI as an Admin with user management permissions.

Note

As custom password policies and users are domain-specific, you might have to enable **I am a domain user** and specify a **Home Domain** as you log in, to access these resources in a non-root domain.

2. Navigate to **Access Management > Users**.
3. Click the desired username. Use the search function to find a username, if needed.
4. Expand **Products and Permissions**
5. Select the desired **Password Policy** from the dropdown menu.
6. Set a **Password** and **Password Match**. The new password must conform to the new password policy.
7. Click **Reset** to assign the password policy and update the user's password.
8. Provide the new password to the user.

ksctl Password Policy Management

All password policy operations are also available through the [ksctl CLI](#).

To set the entire password policy

You can set the entire password policy. See the "[CLI Guide](#)" embedded in CipherTrust Manager for details on each parameter.

```
-f, --failed-logins-lockout-thresholds    list of lockout durations in minutes f
```

```

or failed login attempts.
-L, --lifetime      maximum lifetime of the user password.
-b, --history       number of past passwords saved. This sets how frequently old p
asswords can be reused.
-t, --minlength     minimum length of the password.
-z, --maxlength     maximum length of the password.
-m, --minupper      minimum number of upper case letters.
-w, --minlower      minimum number of lower case letters.
-d, --mindig        minimum number of digits.
-o, --minother      minimum number of other characters.
--pwdchnghdays     maximum lifetime of the password in days.

```

To set the allowed password length and characters in one string:

```

ksctl users pwdpolicy update --minlength 8 --maxlength 30 --minupper 1 --minlower
1 --minother 1 --mindig 1

```

To update a single aspect of the password policy

```

ksctl users pwdpolicy update --maxlength 100

```

To set password to expire in 30 days

```

-L, --lifetime      maximum lifetime of the user password.
ksctl users pwdpolicy update --lifetime 30

```

To set password to never expire

```

ksctl users pwdpolicy update --lifetime 0

```

To set the user account lockout thresholds

```

-f, --failed-logins-lockout-thresholds List of lockout durations in minutes for f
ailed login attempts.

```

In this example, values [0, 5, 30] means that the first failed login attempt, with lockout duration of zero, will not lockout the user account. The second failed login attempt will lockout the account for 5 minutes. The third and subsequent failed login attempts will lockout for 30 minutes.

```

ksctl users pwdpolicy update -f [0, 5, 30]

```

Note

If you set these thresholds while a user is locked out, the active lockout period must complete before the new thresholds are applied to that user's authentication attempts.

To disable user account lockout

To disable user account lockout, set an empty array '[]'.

```
ksctl users pwdpolicy update -f []
```

To unlock a user account

A user account that has been locked due to failed log in attempts can be unlocked by an Application Administrator.

```
ksctl users modify --id "local|c9161a90-0838-469b-87e9-726d8c539f3f" -u
```

To change the password history

```
-b, --history      number of past passwords saved. This sets how frequently old p  
asswords can be reused
```

To set the password history to the value 7, issue this command:

```
ksctl users pwdpolicy update --history 7
```

Assigning custom password policies

The CipherTrust Manager allows creating multiple password policies for the users. These custom password policies can be explicitly assigned to users using the `password_policy` field while creating or updating the user details.

```
password_policy: "custom_policy <Name of the custom policy>"
```

Example Request 1 (creating a user with custom password policy)

```
ksctl users create --username "testuser1" --password-policy custom_policy --pwor  
d "pass@A123"
```

Example Response

```
{  
  "created_at": "2023-02-21T04:48:40.447587Z",
```

```

"email": "testuser1@local",
"last_login": null,
"logins_count": 0,
"name": "testuser1",
"nickname": "testuser1",
"updated_at": "2023-02-21T04:48:40.447587Z",
"user_id": "local|d943506b-d3f9-4774-b5c3-c812c2eb4b9c",
"username": "testuser1",
"failed_logins_count": 0,
"account_lockout_at": null,
"failed_logins_initial_attempt_at": null,
"last_failed_login_at": null,
"password_changed_at": "2023-02-21T04:48:40.432542Z",
"password_change_required": false,
"certificate_subject_dn": "",
"enable_cert_auth": false,
"auth_domain": "00000000-0000-0000-0000-000000000000",
"login_flags": {
  "prevent_ui_login": false
},
"allowed_auth_methods": [
  "password"
],
"allowed_client_types": [
  "unregistered",
  "public",
  "confidential"
],
"password_policy": "custom_policy"
}

```

Example Request 2 (assigning custom password policy to an existing user)

```

ksctl users modify --id "local|d943506b-d3f9-4774-b5c3-c812c2eb4b9c" --password-policy test_policy --pword "Pass@A123"

```

Example Response

```

{
  "created_at": "2023-02-21T04:48:40.447587Z",
  "email": "testuser1@local",
  "last_login": null,
  "logins_count": 0,
  "name": "testuser1",
  "nickname": "testuser1",
  "updated_at": "2023-02-21T05:05:50.564412Z",
  "user_id": "local|d943506b-d3f9-4774-b5c3-c812c2eb4b9c",
  "username": "testuser1",
  "failed_logins_count": 0,
  "account_lockout_at": null,
  "failed_logins_initial_attempt_at": null,
  "last_failed_login_at": null,
  "password_changed_at": "2023-02-21T05:05:50.562118Z",
  "password_change_required": false,
  "certificate_subject_dn": "",
  "enable_cert_auth": false,

```

```

    "login_flags": {
      "prevent_ui_login": false
    },
    "allowed_auth_methods": [
      "password"
    ],
    "allowed_client_types": [
      "unregistered",
      "public",
      "confidential"
    ],
    "password_policy": "test_policy"
  }

```

To know the default values for custom password policies, refer to [allowed password length and characters](#).

Managing custom password policies

The following operations can be performed:

- Create/Get/Change/Delete custom password policies
- List all password policies

For parameters details, refer to [To set the entire password policy](#).

Creating custom password policies

To create a custom password policy, run:

Example Request

```

ksctl users pwdpolicy create --policy-name "custom_policy" --minupper 1 --minlower 1 --mindig 1 --minother 0 --minlength 8 --maxlength 30 --lifetime 30 --failed-logins-lockout-thresholds "[0,0,30]" --history 0 --pwdchngdays 1

```

Example Response

```

{
  "inclusive_min_upper_case": 1,
  "inclusive_min_lower_case": 1,
  "inclusive_min_digits": 1,
  "inclusive_min_other": 0,
  "inclusive_min_total_length": 8,
  "inclusive_max_total_length": 30,
  "password_history_threshold": 0,
  "failed_logins_lockout_thresholds": [
    0,
    0,
    30
  ],
  "password_lifetime": 30,
  "policy_name": "custom_policy",

```

```
"password_change_min_days": 1
}
```

Getting details of custom password policies

To get details of a custom password policy, run:

Example Request 1 (with policy name)

```
ksctl users pwdpolicy get --policy-name custom_policy
```

Example Response

```
{
  "inclusive_min_upper_case": 1,
  "inclusive_min_lower_case": 1,
  "inclusive_min_digits": 1,
  "inclusive_min_other": 0,
  "inclusive_min_total_length": 8,
  "inclusive_max_total_length": 30,
  "password_history_threshold": 0,
  "failed_logins_lockout_thresholds": [
    0,
    0,
    30
  ],
  "password_lifetime": 30,
  "policy_name": "custom_policy",
  "password_change_min_days": 1
}
```

If the *policy name* is not specified in the request, then the applied password policy is fetched. By default, the global password policy is applied to the users.

Example Request 2 (without policy name)

```
ksctl users pwdpolicy get
```

Example Response

```
{
  "inclusive_min_upper_case": 1,
  "inclusive_min_lower_case": 1,
  "inclusive_min_digits": 1,
  "inclusive_min_other": 0,
  "inclusive_min_total_length": 8,
  "inclusive_max_total_length": 30,
  "password_history_threshold": 0,
  "failed_logins_lockout_thresholds": [
    0,
    0,
    30
  ]
}
```

```
],
"password_lifetime": 30,
"policy_name": "custom_policy",
"password_change_min_days": 1
}
```

Changing custom password policies

To change a custom password policy, run:

Example Request

```
ksctl users pwdpolicy update --policy-name "custom_policy" --minupper 2 --minlower 2 --mindig 2 --minother 2 --minlength 20 --maxlength 20 --lifetime 0 --failed-logins-lockout-thresholds "[0,5,30]" --history 10 --pwdchngdays 30
```

Example Response

```
{
  "inclusive_min_upper_case": 2,
  "inclusive_min_lower_case": 2,
  "inclusive_min_digits": 2,
  "inclusive_min_other": 2,
  "inclusive_min_total_length": 20,
  "inclusive_max_total_length": 20,
  "password_history_threshold": 10,
  "failed_logins_lockout_thresholds": [
    0,
    5,
    30
  ],
  "password_lifetime": 0,
  "policy_name": "custom_policy",
  "password_change_min_days": 30
}
```

Deleting custom password policies

To delete a custom password policy, run:

Example Request

```
ksctl users pwdpolicy delete --policy-name custom_policy
```

There will be no response if policy is deleted successfully.

Getting list of all password policies

To list all the password policies, run:

Example Request


```
ksctl users pwdpolicy list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "inclusive_min_upper_case": 1,
      "inclusive_min_lower_case": 1,
      "inclusive_min_digits": 1,
      "inclusive_min_other": 0,
      "inclusive_min_total_length": 8,
      "inclusive_max_total_length": 30,
      "password_history_threshold": 0,
      "failed_logins_lockout_thresholds": [
        0,
        0,
        30
      ],
      "password_lifetime": 30,
      "policy_name": "custom_policy",
      "password_change_min_days": 1
    },
    {
      "inclusive_min_upper_case": 1,
      "inclusive_min_lower_case": 1,
      "inclusive_min_digits": 1,
      "inclusive_min_other": 1,
      "inclusive_min_total_length": 8,
      "inclusive_max_total_length": 30,
      "password_history_threshold": 5,
      "failed_logins_lockout_thresholds": [
        0,
        0,
        0,
        0,
        1
      ],
      "password_lifetime": 0,
      "policy_name": "global",
      "password_change_min_days": 0
    }
  ]
}
```

Key Policies

A key policy consists of all the permissions that are applied to a given label. It is a mapping of labels to their corresponding permissions. Clients, users, and groups can have access to these permissions. It outlines which clients, users, or organizations are permitted to use, read, sign, or interact with the key. Therefore, all labels associated with a key are retrieved anytime an action on that key is performed. The client, user, or group is then determined to be permitted to operate by computing all important policies connected to these labels.

This feature allows the **Key Admins** and **Admins** to perform the following operations on Key Policies:

- Create
- Update
- Delete
- List

Note

The examples are provided for the Windows platform. To run these examples on Linux, see [CLI usage with JSON parameters](#).

Permission Set

A permission set is a collection of permissions that grants you access to different operations that can be performed on the keys. For example, 'UseKey', 'ReadKey', 'SignWithKey', and so on.

The **permission_set** specifies which clients, users, or groups have permission to use a key, read a key, sign a key, and other operations.

```
{
  "clients": {
    "UseKey": [],
    "ReadKey": ["ClientID"],
    "SignWithKey": [],
    "DecryptWithKey": [],
    "EncryptWithKey": [],
    "SignVerifyWithKey": [],
    "ExportKey" :[],
    "UploadKey" :[]
  },
  "groups": {
    "UseKey": [],
    "ReadKey": ["Application Data Protection Admins"],
    "SignWithKey": [],
    "DecryptWithKey": [],
    "EncryptWithKey": [],
    "SignVerifyWithKey": [],
    "ExportKey" :[],
    "UploadKey" :[]
  },
  "users": {
    "UseKey": [],
    "ReadKey": ["UserID"],
    "SignWithKey": [],
    "DecryptWithKey": [],
    "EncryptWithKey": [],
    "SignVerifyWithKey": [],
    "ExportKey" :[],
    "UploadKey" :[]
  }
}
```

}

Managing Key Policies using ksctl

The following operations can be performed on Key Policies using ksctl:

- Creating a Key Policy
- Updating a Key Policy
- Getting details of a Key Policy
- Deleting a Key Policy
- Getting a list of Key Policies

Note

- The mandatory parameters such as name, label-selector, and permissions are required to be provided while creating a Key policy.
- The format of **label-selector** must be correct - label-selector env=test.
- Only the operators '=' and 'in' are supported. Also, multiple conditions can be combined using ','. For example, env in (test1,test2), test3=test4.

Creating a Key Policy

To create a Key Policy, run the following command:

Syntax

```
./ksctl key-policy create --name <name> --description <description> --label-selector <label_selector> --permissions <permission_set>
```

Example Request

```
./ksctl key-policy create --name testPolicy --description testing --label-selector env=test1 --permissions "{\"groups\":{\"ReadKey\":[\"group1\"]}}"
```

Example Response

```
{
  "id": "3222a138-28b2-4e4d-9269-d8bf01fe7cfe",
  "uri": "kylo:kylo:vault:key_policy:testpolicy-3222a138-28b2-4e4d-9269-d8bf01fe7cfe",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-08-25T12:24:59.52885Z",
  "name": "testPolicy",
  "description": "testing",
  "label_selector": "env=test1",
  "permissions": {
    "clients": {},
    "users": {},
    "groups": {
      "ReadKey": [
        "group1"
      ]
    }
  }
}
```

Updating a Key Policy

Syntax

```
./ksctl key-policy update --id <id> --description <new_description> --permissions <new_permission_set>
```

Example Request

```
./ksctl key-policy update --id 3222a138-28b2-4e4d-9269-d8bf01fe7cfe --permissions "{\\"groups\\":{\\"ReadKey\\":[\\"test2\\"]}}"
```

Example Response

```
{
  "id": "3222a138-28b2-4e4d-9269-d8bf01fe7cfe",
  "uri": "kylo:kylo:vault:key_policy:testpolicy-3222a138-28b2-4e4d-9269-d8bf01fe7cfe",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-08-25T12:24:59.52885Z",
  "name": "testPolicy",
  "description": "testing",
  "label_selector": "env=test1",
  "permissions": {
    "clients": {},
    "users": {},
    "groups": {
      "ReadKey": [
```

```

    "test2"
  ]
}
}
}

```

Getting details of a Key Policy

Syntax

```
./ksctl key-policy get --id <id>
```

Example Request

```
./ksctl key-policy get --id 3222a138-28b2-4e4d-9269-d8bf01fe7cfe
```

Example Response

```

{
  "id": "3222a138-28b2-4e4d-9269-d8bf01fe7cfe",
  "uri": "kylo:kylo:vault:key_policy:testpolicy-3222a138-28b2-4e4d-9269-d8bf01fe7cfe",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-08-25T12:24:59.52885Z",
  "name": "testPolicy",
  "description": "testing",
  "label_selector": "env=test1",
  "permissions": {
    "clients": {},
    "users": {},
    "groups": {
      "ReadKey": [
        "test2"
      ]
    }
  }
}

```

Deleting a Key Policy

Syntax

```
./ksctl key-policy delete --id <id>
```

Example Request

```
./ksctl key-policy delete --id 3222a138-28b2-4e4d-9269-d8bf01fe7cfe
```

Getting a list of Key Policies

Syntax

```
./ksctl key-policy list
```

Example Request

```
./ksctl key-policy list --labels "[{\"env\":\"test1\"}]"
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "3222a138-28b2-4e4d-9269-d8bf01fe7cfe",
      "uri": "kylo:kylo:vault:key_policy:testpolicy-3222a138-28b2-4e4d-9269-d8bf01fe7cfe",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2022-08-25T12:24:59.52885Z",
      "name": "testPolicy",
      "description": "testing",
      "label_selector": "env=test1",
      "permissions": {
        "clients": {},
        "users": {},
        "groups": {
          "ReadKey": [
            "test2"
          ]
        }
      }
    }
  ]
}
```

Changing Passwords

This section describes how to change the Application Administrator's initial password, if needed, and the more general case of changing the password for a user.

Note

Passwords must meet the current Password Policy settings. For more information on Password Policy and on changing the settings, see [Password Policy](#).

Changing the Application Administrator's initial password using the GUI

At first boot, the default Application Administrator is "admin" with the password "admin", which must be changed before using the CipherTrust Manager.

To change the Application Administrators initial password using the GUI

Open the URL: `https://<ciphertrust_ip_address>`

1. Login as user "admin" using the default initial password "admin". If the password is not the hard-coded "admin", contact your System Administrator for an auto-generated password.
2. After successfully logging in, the user is prompted to change the initial password immediately. Failure to do so could allow an attacker to compromise the system.

For more information changing the password for the Initial User, see [Changing the Initial Password](#)

Changing user passwords

To change a user password using the CLI:

```
$ ksctl changepw --user admin
```

User is prompted to enter and confirm the new password:

```
Please provide the new password of the user:
```

```
Please confirm the new password:
```

Domain Management

A domain is a logical division that isolates users and cryptographic keys. Resources like keys are isolated to a single domain and cannot be accessed across domains.

CipherTrust Manager's domains form its multitenancy model. Multitenancy enables the creation of multiple restricted or local domains within a single CipherTrust Manager appliance. Enterprises can use multitenancy to segregate key access (or other resource access) to better control usage of product by line of business or use case.

Every CipherTrust Manager appliance has a root domain by default. Additional child domains can be created, and child domains can have their own children domains. We have validated 100 domains for Virtual CipherTrust Manager k170v, and 1000 domains for every other model, and can guarantee that capacity. A domain administrator role controls the creation, deletion, update, and user assignment of domains.

Note

Child domains are not available in Community Edition, and require a valid Virtual CipherTrust Manager license. The license can be the included 90-day trial license, or a purchased license. Without a license, only the root domain is available. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

You can optionally anchor a child domain to a Luna Network HSM partition during creation. All keys and secrets within an HSM-anchored domain are wrapped and unwrapped by the HSM itself. This strengthens the security and isolation of the multitenancy model. There are back up and performance considerations for this option, which should be **planned** for before enabling the feature.

Not all resources are segregated. Certain settings and configurations are only configurable in the root domain. This includes:

- System settings such as NTP servers.
- Authentication settings such as LDAP servers.

CipherTrust Manager allows you to partition a single instance into multiple domains, and allows separate administrators with separation of duties to manage encryption keys, security controls, and auditing in each domain.

Domain Planning

The domain administrator should plan requirements for the domains before creating them.

Note

Domains do not support patch operations. Therefore, domain-related properties except the Syslog message redirection cannot be modified.

Administration

Plan the following:

- User Management
 - Number and type of administrators required. Administrators can assign user groups to the domain and manage child domains. If administrators have the correct permissions, they can also perform operations on domain resources directly.

- Number of users requiring access to the domain's resources.
- Administrative operations/practices
 - Backup and key rotation schedule.
 - Access to and storage of [audit records](#) and [activity logs](#) on an external [log forwarder](#) for audit purposes. By default, audit records and activity logs from a child domain are redirected to the log forwarder configured for the parent domain. The redirection setting is described [below](#).

Certificate Protection

Each domain has a default Certificate Authority (CA) automatically created. This field defines the parent CA which signs the Default CA for the new domain. If no parent CA is provided, the oldest CA in the root domain is used.

Consider the following characteristics of the parent CA:

- The default CA of a domain is signed by the parent CA selected while creating the domain.
- The duration of domain's default CA cannot be more than the parent CA duration.
- The maximum duration of a domain's default CA can be 10 years.

Key Hierarchy

Decide whether to anchor the domain to an HSM.

There are additional dependencies and limitations to consider for HSM-anchored domains:

- Configuration of the HSM partition(s) is required. Consult [Luna Network HSM Partition Compatibility for HSM-Anchored Domains](#) for details.
- Configuration of the HSM connection(s) is required.
- To successfully backup and restore HSM-anchored domain contents, the CipherTrust Manager needs an active connection to the HSM partition. As well, the destination domain must be created with a connection to the same HSM partition; an HSM connection cannot be added after creation.
- There might be a performance impact to applications that frequently interact with the domain data keys for cryptographic operations and key exports. The key caching [system property](#) can improve performance for such applications. This system property caches data key material in CipherTrust Manager memory, so that subsequent requests to use a data key don't require the HSM to unwrap the key.

Note

Consult documentation for CipherTrust connectors to see if data key caching would be beneficial to performance in your deployment. The CipherTrust Manager key cache might not noticeably improve

performance for connectors which also cache keys client-side.

Luna Network HSM Partition Compatibility for HSM-Anchored Domains

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.

User Management

When a domain is created, one or more initial administrators must be defined. When administrators subsequently log into the domain, they can use assign additional users to the domain and also use groups to assign user permissions.

Note

The CipherTrust Manager allows you to create users in a non-root domain if you set the `allow-user-mgmt` property to `true` while creating the domain. To create a user within a non-root domain, set the `is_domain_user` property to `true`.

The defined Admins should, at minimum, have [group permissions](#) to create domains, users, and groups. You can assign the admins to the 'Domain Admins' and 'User Admins' system-defined groups to have the correct permissions.

Creating Domains

A new domain is automatically a child domain of the logged-in domain at the time of creation. So, if you are logged into the root domain, and create a new domain, the new domain is a child of root.

By default, a parent domain and its descendants are part of the same certificate chain. You can create domains in a parent-child hierarchy down to ten levels before having to manually replace their associated certificates for client certificate authentication.

The workflow to create a domain varies depending on whether you wish to have the domain anchored in CipherTrust Manager software or in an HSM partition.

Note

A domain name can contain all characters and there is no restriction on the domain name length.

To create a domain anchored in CipherTrust Manager software

1. Review the [domain planning considerations](#).
2. Log into the CipherTrust Manager Management Console as an Admin with domain creation permissions.
3. Navigate to **Admin Settings > Domains**.
4. Select **Add Domain**.
5. Enter the following information in the **Add Domain** dialog:
 - **Name:** Name of the domain
 - **Admins:** Define the initial set of Admins who can log into the domain, assign users to groups to the domain, and create child domains. These Admins should have the [correct permissions](#).

The following information is optional:

- **Parent CA:** Each domain has a default Certificate Authority (CA) automatically created. This field defines

the parent CA which signs the Default CA for the new domain. If no parent CA is provided, the oldest CA in the root domain is used.

- **Allow Subdomain User Management:** Select the check box to allow the creation and management of users in non-root domains.
- **Meta:** Additional reference information regarding the domain. Must be JSON formatted. This can be used to attach information – such as the department or contact details – to the domain. It can then be queried and viewed later. This field is not used internally by CipherTrust Manager.

To create an HSM-anchored Domain

1. Review the [domain planning considerations](#), especially the [key hierarchy considerations](#).
2. Configure a Luna Network HSM version partition following [Luna Network documentation](#). Make sure the Luna Network HSM partition is compatible with the [key hierarchy considerations](#).
3. As an Admin with Connection Manager permissions, configure a connection to the HSM partition in [Connection Manager](#).

Warning

Thales strongly discourages using a Luna HSM partition containing the CipherTrust Manager [root of trust keys](#) for HSM-anchored domains. Do not add a connection to a Luna HSM root-of-trust partition with Connection Manager.

4. As an Admin with domain creation permissions, navigate to **Admin Settings > Domains**.
5. Select **Add Domain**.
6. Enter the following information in the **Add Domain** dialog:
 - **Name:** Name of the domain
 - **Admins:** Define the initial set of Admins who can log into the domain, assign users to groups to the domain, and create child domains. These Admins should have the [correct permissions](#).
 - **HSM-anchored domain:** Select the check box to anchor the domain to an HSM.
 - **HSM connection:** Use the drop-down to select the partition connection you created in step 3.

The following information is optional:

- **Parent CA:** Each domain has a default Certificate Authority (CA) automatically created. This field defines the parent CA which signs the Default CA for the new domain. If no parent CA is provided, the oldest CA in the root domain is used.

- **HSM KEK label:** A name for the HSM Key Encryption Key (KEK) stored in the HSM partition. If not provided, a random UUID is assigned for KEK label. You can [rotate the initial HSM KEK](#) after creation.
- **Meta:** Additional reference information regarding the domain. Must be JSON formatted. This can be used to attach information – such as the department or contact details – to the domain. It can then be queried and viewed later. This field is not used internally by CipherTrust Manager.

Deleting Domains

Caution

Deleting domains does not delete domain data from the database, but permanently makes this data inaccessible to end users. To clear storage space on the CipherTrust Manager or reduce license usage counts, you must directly delete objects like keys, clients, users, and configurations before deleting the domain.

For HSM-anchored domains, the HSM KEK remains in the HSM partition. If desired, you can clean up this key by interfacing to the HSM partition directly, outside of CipherTrust Manager.

To delete a domain:

1. Log into the CipherTrust Manager Management Console as an Admin with domain creation permissions.
2. Navigate to **Admin Settings > Domains**.
3. Click the overflow icon () corresponding to the desired domain and click **Delete**.

A confirmation pop-up appears.

4. Read the confirmation text and type the name of the domain to confirm.
5. Click **Delete** to finalize your action.

Certificate-based Login for Domain Users

Add New Domain

1. [Deploy](#) the CipherTrust Manager.
2. Log on to the CipherTrust Manager Management Console as an administrator with domain creation permissions.
3. In the left pane, click **Admin Settings > Domains**.
4. Select **Add Domain**.
5. Enter the required details. Refer to the [To create a domain anchored in CipherTrust Manager software](#) section

for the description of fields.

6. Select **Allow Subdomain User Management**.

7. Click **Save**.

Add New Domain User

1. [Switch](#) to the newly created domain.

2. Navigate to **Access Management > Users**.

3. Click **Add User** and enter the required details.

4. Select **Allow user to login using certificate**. Refer to [Enable the "Certificate based Login" Option for a User](#) for details.

5. Click **Add User** to save the details.

Download Local CA Certificate

1. Navigate to **CA > Local**.

2. Click **Add Local CA**.

3. Specify the required details. Refer to [Create and Download the Web Certificate](#) for details.

4. Under Name, click the link of the newly generated local CA.

5. Click **Issue Certificate**.

a. Enter the **Common Name** for this certificate.

Note

This common name should be the same common name that you specified while creating the user.

b. Select the desired algorithm (RSA or ECDSA).

c. In the **Name** field, specify the same details that you specified in the `certificate_subject_dn` property of the user.

d. Click **Issue Certificate**.

e. Click **save private key** to download the *key.pem* file.

f. Click **Issue Certificate**. The newly created certificate is displayed in the certificates list.

-
- g. Download the certificate issued by the local CA and save it at the same location where the private key is saved.

Download External CA and Update Certificate

1. Navigate to **CA > External**.
2. Download the external CA.
3. Append the local CA and external CA to the [certificate downloaded above](#).

Create and Import PKCS#12 (PFX) File

1. Run `openssl pkcs12 -export -out user.pfx -inkey key.pem -in certificate.pem`. Refer to [Create and Install pkcs12 Formatted Certificate](#) for details.
2. Import `user.pfx` to truststore.

Log on as Domain User

1. Clear the browser cookies.
2. Select the web certificate and [log on to the CipherTrust Manager as domain user](#).

Domain Authentication

Once the domain(s) have been created, the specified Administrator can log in to the domain. Users that are assigned to the domain can also login to the domain to perform key operations.

CipherTrust Web UI and Embedded API Guide

The embedded API Guide and CipherTrust Web UI share authentication, so login through the CipherTrust Web UI also allows REST API access.

Login Directly to a Child Domain

1. Select the **I am a domain user** check box.
2. Specify your auth domain in the **Home Domain** field.
3. Provide your **Username** and **Password**, if applicable.
4. Click **Log In** or **Log In with Certificate Only**.

The domain users are always redirected to their auth domain. However, when assigned users log on to the CipherTrust Web UI for the first time, they are redirected to the root domain. Next time onward, after login, these users are redirected to their last logged-in domain on the GUI.

Note

- While creating a domain, users created in the non-root domain cannot be added to the domains admin's list.
- Users created in the non-root domain cannot be assigned to any other domain.
- Certificate-based login does not work for the users created in the non-root domain.
- OIDC and LDAP authentication are only supported for users created in the root domain. If a user is created in a root domain and later assigned to an additional domain, LDAP and OIDC authentication is available for login to the root domain, and after login the user can switch to a non-root domain. For OIDC authentication, the user manually logs into the root domain and then manually switches to a non-root domain in two steps. For LDAP authentication, the user can login to the root domain and switch into a non-root domain in a [single step](#).

Switch Domains After Login

In the CipherTrust Web UI, in the upper right hand corner, click the name of the current domain, click **Switch Domains**, and click the name of the desired domain. You are now taken to the specified domain. From this point forward, all key management and admin activity is constrained to the selected domain.



ksctl CLI

In the CLI, you operate in the root domain by default. There are two ways to perform actions inside a child domain:

- Log into the domain with the `ksctl login --user <user_with_domain_permission> [--password <user_password>] --domain <domain_name_or_id>` command. If you don't specify the password, you are prompted for it. All subsequent commands in the session are within the domain.
- Use the `--domain <domain_name_or_id>` option with `--user <user_with_domain_permission>` for the particular key management or administrative command you wish to perform. You can specify the password with `--password` or enter it when prompted

Note

You can use `ksctl tokens self-domains --user <user_name> [--password <user_password>]` to view the domains that a user can access.

Generic REST API Client

In the [REST API](#), the `POST /v1/auth/tokens` command to authenticate goes to the root domain by default. You can include a child domain name in the schema with `domain`, as well as the domain administrator username and password. After this initial authentication, subsequent API requests apply to the authenticated domain.

You can also use the `GET /v1/auth/self/domains/` endpoint to return a list of domains that the current user is assigned to.

Domain Scoped Backups

You can [scope a backup to a single domain](#).

Rotating the HSM KEK

You can rotate the HSM KEK for HSM-anchored domains, which re-wraps all the keys and secrets in the domain with a different key held in an HSM partition. This is a different option than [editing the HSM KEK label](#), which just restores the association between a domain and its active HSM KEK.

The following rotation operations are available:

- [Rotate the HSM KEK](#)
- [Check the status of a KEK rotation](#)
- [Retry a failed rotation](#)

Rotate the HSM KEK

1. Log into the CipherTrust Manager Management Console as an Admin with domain permissions.
2. Navigate to **Admin Settings > Domains**.
3. Click the overflow icon () corresponding to the desired domain and click **View/Edit**.

On the details page, the KEK list is visible, with the current KEK at the top of the list, and all previous KEK listed below.

4. Click the **Rotate KEK** button.
5. Change the default rotation settings, if desired. By default, a new HSM KEK is created with a randomly generated UUID label, and stored in the existing HSM partition.

- a. To store the HSM KEK in a different HSM partition, select a different **HSM Connection**.
- b. To rotate to an existing KEK, enter its label as the **HSM KEK Label**.
- c. To create a new KEK with a unique custom label, enter the label as the **HSM KEK Label**.

Note

If there is an existing KEK present in the partition with the label you enter, CipherTrust Manager rotates to that existing KEK.

6. Click **Rotate KEK** to confirm.

Check the Status of a KEK Rotation

While a KEK rotation is ongoing, a notice above the **KEK List** shows progress for re-wrapping all the keys present on the domain, and the status of the KEK is reported as **Started** in the **KEK List**.

If a rotation times out or otherwise fails, the partially rotated KEK displays a **Failed** status in the **KEK List**. Click on the row to view details of the failed rotation job, including start and finish times, and **Meta** details of total domain keys, number of domain keys rotated, and number of domain keys that failed to be rotated.

Past successful rotations display a **Success** status. Click on the row to view details of a successful rotation job, including start and finish times, and **Meta** details of total domain keys and number of domain keys rotated.

Retry a Failed Rotation

If an HSM KEK rotation times out or otherwise fails, you can retry the rotation.

1. Log into the CipherTrust Manager Management Console as an Admin with domain permissions.
2. Navigate to **Admin Settings > Domains**.
3. Click the overflow icon () corresponding to the desired domain and click **View/Edit**.

On the details page, the KEK list is visible, with the current KEK at the top of the list, and all previous KEK listed below.

4. In the KEK list, find and the desired HSM KEK with **Failed** status. Click the **Retry** button at the far right of the row.

A confirmation dialog pops up.

5. Click **Retry KEK Rotation** to confirm.

The rotation continues from its previous point. Keys that that were successfully re-wrapped in previous failed rotation attempts are not wrapped again.

Editing HSM-anchored Domains to Recover Lost HSM KEK

You can edit HSM-anchored domains to recover a lost association to the active HSM KEK stored in an HSM partition. This is helpful if CipherTrust Manager's connection to the HSM partition was deleted, if the HSM KEK was moved to a different HSM partition, or if the HSM KEK's label was changed directly on the HSM.

To re-associate a domain with its original HSM KEK:

1. Log into the CipherTrust Manager Management Console as an Admin with domain permissions.
2. Navigate to **Admin Settings > Domains**.
3. Click the overflow icon () corresponding to the desired domain and click **View/Edit**.
4. Under the **HSM Domain Connection** section, select another **HSM connection** from the dropdown if desired.
5. Edit the **HSM KEK Label**, if desired. A key with this label must already exist on the HSM partition.

This operation does not re-wrap domain keys and secrets. [Rotate the HSM Kek](#) to re-wrap those objects.

6. Click **Update** to finalize your changes.

Log Forwarder Redirection

Log forwarder redirection is a setting to send audit records, KMIP activity logs, and NAE activity logs from a child domain to the [log forwarder](#) configured for a higher-level domain.

When an audit record, NAE activity log, or KMIP activity log is created in a child domain with log forwarder redirection, CipherTrust Manager checks the domain's immediate parent for a configured log forwarder. If the immediate parent also has log redirection enabled, CipherTrust Manager also checks its parent, the child domain's grandparent, for a configured log forwarder. CipherTrust Manager continues checking for log forwarders further up the domain hierarchy every time it finds the log redirection setting. Every time CipherTrust Manager finds a configured log forwarder in a domain, it forwards log messages to that log forwarder.

Note

- Log forwarder redirection can be enabled even if higher level domains do not have a log forwarder configured. In that situation, logs from the child domain are not forwarded.
- If you have log redirection enabled and log forwarders configured at multiple consecutive levels in a domain hierarchy, the same log message can be forwarded to multiple log forwarders. To consolidate messages, we recommend configuring only one log forwarder in a domain hierarchy, at the highest domain level possible.
- We recommend that administrators of parent domains are aware of log forwarding settings for child domains. If there is a need to track where logs from a particular domain are forwarded, employees from your organization might need to login to several domains to view log forwarder and log forwarder redirection settings.

Users in the Domain Admins or Application Administrators [group](#), such as `admin` can enable or disable log forwarder redirection. When a new child domain is created, log forwarder redirection is enabled by default. Existing child domains which were created in a version lower than 2.16.0 have log forwarder redirection disabled after upgrade.

Enabling or disabling log forwarder redirection in web console UI:

1. Login to the child domain as a user in the Domain Admins or Application Administrators group, such as `admin`.
2. Navigate to **Admin Settings > Logs**.
3. Click the **Redirect Log Forwarder** toggle to enable or disable log forwarder redirection.

Enabling or disabling log forwarder redirection in ksctl CLI:

[Login to the child domain](#) as a user in the Domain Admins or Application Administrators group, such as `admin`. For example:

```
ksctl login --user <user_with_domain_admin_permission> --password <user_password>
--domain <domain_name_or_id>
```

- The following command disables log forwarder redirection.

```
ksctl domains log-forwarders-redirection disable
```

- The following command enables log forwarder redirection.

```
ksctl domains log-forwarders-redirection enable
```

- The following command displays log forwarder redirection status

```
ksctl domains log-forwarders-redirection status
```

Managing Syslog Messages Redirection to Parent Domain (Legacy Syslog) using ksctl

Syslog messages redirection allows you to send the [legacy syslog](#) messages of the current domain to the legacy syslog server configured in its parent domain.

If the current domain is receiving the syslog messages from its child domain, those syslog messages will also be sent to the legacy syslog server configured in the parent domain of the current domain.

It allows you to perform the following operations:

- Enabling/Disabling syslog messages redirection
- Status of the syslog messages redirection

Enabling/Disabling Syslog Messages Redirection

To enable syslog messages redirection to the parent of the current domain, run:

```
ksctl domains syslog-redirection enable
```

To disable syslog messages redirection to the parent of the current domain, run:

```
ksctl domains syslog-redirection disable
```

Note

To enable/disable the redirection of syslog messages to the parent domain is not applicable for the root domain.

Getting Status of Syslog Messages Redirection

It returns the syslog messages redirection status of the current domain. This status shows whether the redirection to parent domain is enabled or disabled. To get the status, run:

```
ksctl domains syslog-redirection status
```

Note

Status will always be false for the root domain.

Groups

A group carries with it permissions for performing specific tasks. A group also consists of a set of users and/or clients that have been authorized to perform these tasks.

The CipherTrust Manager defines Special System Users, System Defined Groups, and User Defined Groups. In addition, the CipherTrust Manager provides an option to create [customized groups for CipherTrust Transparent Encryption \(CTE\)](#) to control permissions on CTE resources.

Caution

It is critical that credentials of these users are kept in a secure location. If a credential is compromised, an attacker could gain access to sensitive data.

Assign a User to a Group

1. Log in to CipherTrust Manager as a user in the `admin` or `User Admins` group.
2. Navigate to **Access Management > Users**.
3. Click the desired **Username** in the table to view the user details.
4. Expand the **Group Memberships** section.
5. Click **Add Group**.
6. Enable the checkbox for any desired group(s). You can search for a group by name.
7. Click **Add Group** to confirm.

Special System Users

There are a few special system users. These are described here:

'ksadmin' user

The "ksadmin" user, is a specialized operating system administrator that can access the CipherTrust Manager via SSH or via password authentication on a physical server console port.

```
$ ssh ksadmin@<ip or hostname>
```

For public cloud providers, the SSH key used for authentication is the key used to launch the instance. For Private Cloud Images (e.g. VMware, Hyper-V), the SSH key must be replaced before the system will fully boot, which can be done via the CLI, API, or through a web browser.

The **ksadmin** user has permission to run a specific set of commands using **sudo**. These commands allow the user to troubleshoot problems on the CipherTrust Manager server, and perform upgrades. To see the list of commands that can be run with **sudo**, type the command **sudo -l**.

The duties of the System Administrator ("ksadmin" user) are:

- Deploying and configuring the CipherTrust Manager:
 - Run cloud-init commands, refer to [Plan Configuration Settings for Cloud-Init](#).
 - Run **kscfg** commands, refer to [System Configuration Utility](#).

- Retrieving the initial application admin user password, if configured to generate a unique password on first boot.
- Troubleshooting CipherTrust Manager issues in conjunction with Thales Technical Support.

When making a support call, the System Administrator may be asked to log in to the CipherTrust Manager using ssh to retrieve CipherTrust Manager Logs.

- Applying system upgrades (see [System Upgrade/Downgrade](#)).
- [Upgrading PCIe HSM firmware](#) for Thales CipherTrust Manager k570 appliances.

'admin' user

Initially, there is only one Application Administrator and the name of this user is 'admin'.

Note

You can only delete the 'admin' user if another Application Administrator is present. The policy engine enforces that at least one Application Administrator exists at all times to prevent lockouts.

An 'admin' user, and all other Application Administrators if created, are responsible for:

- Creating and managing Users and Groups
- Configuring the CipherTrust Manager ports and licenses
- Viewing audit logs
- Managing backups
- Administrating clusters
- Adding other users to the 'admin' group

Note

A user that is not in the "admin" group is known as an Application User. An Application User must also be part of the System Defined 'Key Users Group' to create and manage their keys. Refer to the ['Key Users' group](#).

The Application Administrator can also perform all duties of the Application User.

'Global' user

The 'Global' user exists to support specific NAE-XML functionality for compatibility with SafeNet KeySecure

Classic and should not be deleted or modified.

System Defined Groups

System Defined Groups exist on CipherTrust Manager at launch time. Each System Defined Group carries with it permissions to perform specific tasks.

Note

For NAE requests, the System Defined Groups can be masked by modifying the NAE interface using the `ksctl` utility. For details, refer to [To create/modify the NAE interface to mask system groups from NAE requests](#).

System Defined Groups are:

'admin' group

There is a System Defined Group named "admin". Users within the "admin" group are referred to as Application Administrators. Application Administrators have full privileges and are able to perform any operation via the REST API, CLI, NAE-XML or GUI interface.

Note

Initially, there is only one Application Administrator and the name of this user is "admin". If there is only one Application Administrator, this user cannot be deleted. The policy engine enforces that at least one Application Administrator exists at all times to prevent lockouts.

An Application Administrator is responsible for:

- Creating and managing Users and Groups
- Configuring the CipherTrust Manager ports and licenses
- Viewing audit logs
- Managing backups
- Administrating clusters
- Adding other users to the 'admin' group

The Application Administrator can also perform all duties of the Application User.

'All Clients' group

A client, upon successful registration with CipherTrust Manager, is made a member of All Clients group. These clients have permissions to:

-
- enroll with their respective CipherTrust Manager services.

'Application Data Protection Admins' group

There is a group named "Application Data Protection Admins*". Users within this group are Application Data Protection Administrators.

The Application Data Protection Administrator is responsible for creating and managing resources in the Application Data Protection tile:

- Defining application, including configuring connector settings.
- Configuring protection policy, including
 - creating user sets
 - configuring access policies
 - creating character set
 - creating masking formats

'Application Data Protection Clients' group

There is a group named "Application Data Protection Clients". The users who are part of this group only have read access to the Application Data Protection tile resources.

'Audit Admins' group

Users who belong to "Audit Admins" group are audit records administrators. These users have permissions to:

- View audit records

'Backup Admins' group

Backup Administrators have permissions to:

- create backups
- create backup keys

'CA Admins' group

CA Administrators have permissions to:

- create Certificate Authorities on the CipherTrust Manager
- manage Certificate Authorities on the CipherTrust Manager
- generate Certificate Signing Requests (CSRs)

'Domain Admins' group

Domain Administrators have permissions to:

- list the domains for a specific account
- create a domain
- access information about a domain
- delete a domain

'CCKM Admins' group

There is a System Defined Group named "CCKM Admins". Users within the "CCKM Admins" group are CCKM Administrators. Additionally, the "CCKM Admins" 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' group

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.

Client Admins

There is a System Defined Group named "Client Admins". Users within the "Client Admins" group can perform some administrative tasks on the CipherTrust Manager Clients.

Client Administrators have permissions to:

- Read a client
- Delete a client
- Renew a client
- Manage KMIP client administration
- Manage client profiles
- Manage registration tokens

Cluster Info Viewers

There is a System Defined Group named "Cluster Info Viewers". Users and clients within the "Cluster Info Viewers" group can view cluster information, but not edit it.

Cluster Info Viewers have permissions to:

- View detailed cluster replication information in the ksctl CLI or REST API.
- View general cluster node status in the CipherTrust Web UI, ksctl CLI, or REST API.

Connection Admins

There is a System Defined Group named "Connection Admins". Users within the "Connection Admins" group are Connection Manager Administrators.

Connection Manager Administrators have permissions to:

- Create connections with third party servers and services such as AWS, Azure, DSM, Google Cloud Platform (GCP) Cloud, Hadoop, Luna SA HSM, SCP, CIFS/SMB, or Salesforce.
- Read, delete, or update the connections.
- Test an already created connection.
- Test a new connection with the connection parameters.

'CSM Admins' Group

There is a System Defined Group named "CSM Admins". The CSM Admins are responsible for managing Akeyless configuration and Customer fragments.

Note

The user must belong to the Connection Admins group to create Akeyless connections.

'CTE Admins' group

There is a System Defined Group named "CTE Admins". Users within the "CTE Admins" group are CTE Administrators.

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

- Clients and client groups
- Profiles, policy elements, and policies
- GuardPoints
- Client registration tokens (with additional rights of System Defined Group named "CA Admins" and "Client Admins")

Note

Only users of the "CTE Admins" group can delete CTE keys.

'CTE Clients' group

There is a System Defined Group named "CTE Clients". CTE clients registered with the CipherTrust Manager are part of this group.

'DDC Admins' group

Data Discovery and Classification (DDC) Administrators can create and manage all DDC resources. For example, they can:

- Create and manage locations, classification profiles, data stores, and scans
- Configure, run, and view reports
- View sensitivity levels
- Manage Hadoop configuration
- Decrypt scan packages coming from databases

'DDC Full Reports Admins' group

DDC Full Report Administrators can:

- Create, view, and run reports
- View available data stores

-
- View available scans
 - View available classification profiles
 - View available sensitivity levels
 - View locations

'DDC Full Similarity Searches Admins' group

This group has two permissions under custom policies for getting and exporting similarity search results.

- Read similarity search results
- Read similarity search data store permission

'DDC Infotype Admins' group

DDC Infotype Admins can view and edit custom infotypes.

'DDC Infotype Viewers' group

DDC Infotype viewers can view custom infotypes.

'DDC L3 Support' group

DDC L3 Support Administrators can help identify and troubleshoot issues you may encounter when using DDC. They can decrypt scan packages coming from databases.

'DDC Profiles Admins' group

DDC Profile Administrators can:

- Create and manage classification profiles
- View available scans

'DDC Profiles Viewer' group

DDC Profile Viewers can only view available classification profiles.

'DDC Reports Admins' group

DDC Reports Administrators can:

- Create, view, and run reports
 - View available data stores
 - View available scans
-

-
- View available sensitivity levels

'DDC Scans Admins' group

DDC Scan Administrators can:

- Create and manage scans
- View available classification profiles
- View available data stores

'DDC Scans Viewer' group

DDC Scan Viewers can only view available scans.

'DDC Similarity Searches Admins' group

This group has the following permissions under a policy of type CTE:

- Read data store
- Read Infotype
- Create similarity search
- Update similarity search
- Read similarity search
- Launch similarity search

'DDC Similarity Searches Viewer' group

This group has the following permissions under a policy of type CTE.

- Read data store
- Read infotype
- Read similarity search

'DDC Stores Admins' group

DDC Store Administrators can create and manage:

- Data stores
- Locations

'DDC Stores Viewers' group

DDC Store Viewers can only view available:

- Data stores
- Sensitivity levels
- Locations

'DDC V2 Admins' group

DDC V2 Admins have CRUD (Create, Read, Update, and Delete) permissions both for the DDC ML agent connection and the Config Profile required for DDC ML agent.

'DDC V2 Clients' group

DDC V2 Clients have CRUD (Create, Read, Update and Delete) permissions for the ML agent connection and only read permission for the Config Profile required for the DDC ML agent.

'Domain Backup Admins' group

Domain Backup Admins have permissions to:

- create domain-scoped backups
- create domain backup keys

'Domain Restore Admins' group

Domain Restore Admins have permissions to:

- restore domain-scoped backups
- read and restore domain backup keys

'DSF Hubs'

This system group is generally intended to grant Data Security Fabric (DSF) clients required permissions for File Activity Monitoring (FAM) as integrated with CTE and DDC. At least one DSF Hubs user is required to register a DSF Hubs client. We recommend that you only use the DSF Hubs user for that purpose.

DSF Hubs users and clients have permissions to:

- Create, view, modify, and delete **FAM policies**.
- Create FAM security rules. View, modify, and delete FAM security rules created by the same client/user.
- Create FAM policy associations. View, modify, and delete the FAM policy associations created by the same client/user.

-
- Create [FAM destinations](#). View, modify, and delete the FAM destinations created by the same client/user.
 - View and update FAM attributes.
 - Create, view, modify, and delete CTE policy elements with protected label, created by the same client/user.
 - View all other CTE resources.

'HSM Admins' group

HSM Administrators have permissions to:

- configure an HSM for the CipherTrust Manager
- manage an HSM for the CipherTrust Manager

'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

'Key Users' group

Users that are not in the "admin" group are Application Users. An Application User must also be part of the System Defined 'Key Users' group for permission to do the following:

- create keys
- perform operations with any key they own or to which they have been granted access
- manage KMIP client administration

'Restricted Key Users' group

Restricted Key Users have permissions to:

- create keys using templates. The templates can be created by Admins, Key Admins groups.
- read the template that matches the ownerId of the user.
- read the templates with no ownerId defined.
- perform key lifecycle management operations such as Revoke and Reactivate.

The users part of the Restricted Key Users cannot update the created key.

'License Admins' group

'License Admins' can manage product licensing. They have permissions to:

- View licenses
- Add new licenses
- Delete licenses
- Enable/disable the trial evaluation

'Migration Split Key Admins' group

Users who belong to this group manage the migration split keys required for [Data Security Manager migration](#). These users can:

- create or delete migration split keys
- create, delete, or modify migration split key shares

'Read-Only Admins' group

This group's purpose is to allow members to access and monitor all CipherTrust Manager systems without the ability to change them. A Read-Only Admin can list all objects of a given resource type, retrieve details about a particular resource, view statuses, and download logs.

'Restore Admins' group

Restore Administrators have permissions to:

- restore backups
- read and restore backup keys

'Restricted Key Users' group

Restricted Key Users have permissions to:

- create keys using templates. The templates can be created by Admins, Key Admins groups.
- read the template that matches the ownerId of the user.
- read the templates with no ownerId defined.
- perform key lifecycle management operations such as Revoke and Reactivate.

The users part of the Restricted Key Users cannot update the created key.

'System Admins' group

Members of the 'System Admins' group have permissions to configure the following:

- Interfaces
- LDAP and OIDC connections
- Logging
- NTP
- Instance
- Cluster
- Licenses

Do not confuse members of this group with the 'ksadmin', the System Administrator who is responsible for deploying the CipherTrust Manager server using an SSH connection or the console port on a physical appliance. For more information on the 'ksadmin' refer to ['ksadmin' user](#).

'User Admins' group

User Administrators have permissions to create users and groups. They can:

- Create other sub-administrator users, for example, policy administrator, key administrator, etc., and regular users
- Assign users to most groups.

Note

Only an [existing member of the 'admin' group](#) can assign another user to this group.

'ProtectAPP Users' Group

The 'ProtectAPP Users' group allow CipherTrust Manager users to list the **registration token** needed to register ProtectApp clients. These tokens enable users to successfully register ProtectAPP clients.

'ProtectDB Users' group

There is a System Defined Group named "ProtectDB Users". Users within this group can perform the following ProtectDB operations:

- Configuring databases
- Managing database connections

-
- Managing database tables
 - Managing user mappings

'ProtectFile Administrator' group

There is a System Defined Group named "ProtectFile Admins". Users within the "ProtectFile Admins" group are ProtectFile Administrators.

Note

CTE UserSpace also uses the 'ProtectFile Administrator' group.

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

- Client profiles and clients
- Network shares, and share-clients and share-rules associations
- Clusters, and cluster-clients and cluster-rules associations
- Access policies, access policy groups, and their associations
- Rules and client-rule associations
- Client Registration Tokens (with additional rights of System Defined Group named "CA Admins" and "Client Admins")

'ProtectFile User' group

There is a System Defined Group named "ProtectFile Users". CipherTrust Manager clients enrolled for ProtectFile are part of this group.

Note

CTE UserSpace also uses the 'ProtectFile User' group.

'CSM Admins' Group

There is a System Defined Group named "CSM Admins". The CSM Admins are responsible for managing Akeyless configuration and Customer fragments.

Note

The user must belong to the Connection Admins group to create Akeyless connections.

User Defined Groups

User Defined Groups are created by Application Administrators. Application Administrators can:

- create and delete User Defined Groups
- add users to a User Defined Group
- remove users from a User Defined Group

Administrators may use groups solely for organizing users, or may create policies that use group membership to assign other permissions.

Adding group permissions to keys grants users in a User Defined Group the privileges to perform operations with those keys. The semantics of the NAE-XML requests and the permissions they grant to keys are identical to SafeNet KeySecure Classic.

Groups are stored in CipherTrust Manager's internal database.

Managing User Defined Groups

Using the GUI, REST API or the CLI, an Application Administrator can create a User Defined Group and add users/clients to this group. The following are examples using the CLI:

To create a new User Defined Group called "eng" :

```
$ ksctl groups create -n eng
```

Note

The % character is disallowed for a group name. All other characters are permitted.

The response looks like:

```
{
  "name": "eng",
  "created_at": "2018-04-27T21:15:36.644959Z",
  "updated_at": "2018-04-27T21:15:36.644959Z"
}
```

To add a user to the new User Defined Group "eng":

You specify the group name and the ID of a user that you previously created.

```
$ ksctl groups adduser -n eng -u "<id of user>"
```

The response looks like:

```
{
  "name": "eng",
  "created_at": "2018-05-02T16:47:51.248735Z",
  "updated_at": "2018-05-02T17:24:20.015915Z"
}
```

Customized Groups for CTE

Create CTE groups to provide granular permissions to users on specific CTE resources or all permissions on all resources. A CipherTrust Manager administrator can create, modify, and delete custom CTE groups on the CipherTrust Manager GUI.

For example, the CipherTrust Manager administrator can create a group of users who have only read permissions on the CTE resources. Similarly, the administrator can create another group of users who can perform all operations except the delete operations on the CTE resources.

Refer to [Permissions](#) for the complete list of permissions required to perform operations on CTE resources.

Creating Customized Groups for CTE

Create custom CTE groups to provide granular permissions to users on specific CTE resources or all permissions on all resources. The CipherTrust Manager administrator can create custom CTE groups on the CipherTrust Manager GUI.

To add a custom CTE group:

1. Log on to the CipherTrust Manager GUI.
2. In the left pane, click **Access Management > Groups**.
3. Click **Create New Group**. The General Info screen of the Create New Group wizard is displayed.

General Info

1. Enter a **Name** for the group.
2. Enable **CTE Resource Permissions**.
3. Click **Next**. The CTE Permissions screen is displayed.

CTE Permissions

Add granular permissions to users on specific CTE resources or all permissions on all resources. Refer to [Permissions](#) for the complete list of permissions required to perform operations on CTE resources.

1. Add the permissions:

To grant *all* permissions on *all* resources:

- a. Enable **Select All Resources & Permissions**.
- b. Click **Next**.

To grant *granular* permissions on *specific* resources:

- a. From the **Resource Type** drop-down list, select the resource you want to grant permission on. The field displays the available CTE resources.
- b. From the **Permissions** drop-down list, select single, multiple, or all permissions. The field displays the available permissions for the selected resource. To grant all the available permissions, select **Select All**.

To add permissions on more resources, click **Add More Permissions**. Add permissions for as many resources as required.

To remove a permission, click .

2. Click **Next**. The Assign Members screen is displayed.

Assign Members

This screen displays the available members with their user IDs. Select the members you want to add to the custom CTE group.

1. Select the members. To select all displayed members, select the check box under the Search by Name search field.
2. Click **Next**. The Review screen is displayed.

Review

This screen shows the group details that you have provided. These details are divided into GENERAL INFO, CTE PERMISSIONS, and ASSIGN MEMBERS sections.

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

1. Review the group details displayed on the screen.

If details are incorrect or you want to make any changes, click **Back** and make changes, as appropriate.

Alternatively, you can click the **Edit** links next to the GENERAL INFO, CTE PERMISSIONS, and ASSIGN MEMBERS sections to make changes.

2. Click **Add Group**. The group is successfully created.
 3. Click **Close**.
-

The Groups list shows the newly created custom CTE group.

Modifying Custom CTE Groups

After a custom CTE group is created, the CipherTrust Manager administrator can modify it to add new members or remove existing members.

To modify a custom CTE group:

1. Log on to the CipherTrust Manager GUI.
2. In the left pane, click **Access Management > Groups**. The list of available groups is displayed.
3. Under **Name**, click the group you want to modify. The edit view of the group is displayed.
4. Under **Members of the <group-name> group**:
 - Click **Remove** next to the members you want to remove from the group.
 - Click **Add** next to the members you want to add to the group. The **Member** check box is selected for the member.

Deleting Custom CTE Groups

The CipherTrust Manager administrator can delete custom CTE groups.

To delete a custom CTE group:

1. Log on to the CipherTrust Manager GUI.
2. In the left pane, click **Access Management > Groups**. The list of available groups is displayed.
3. Click the ellipsis icon (**⋮**) corresponding to the group you want to delete.
4. Click **Delete**. A message appear stating that deleting a group may effect the permissions of users within the group.
5. Click **Delete** to confirm the action.

The group is deleted and removed from the Groups list.

Group Mapping

Group maps extend group-based configuration of CipherTrust Manager users to Lightweight Directory Access Protocol (LDAP) or OpenID Connect (OIDC) groups, associated with the Access Management LDAP or OIDC connection. A group map associates an LDAP or OIDC group belonging to a specific connection to a CipherTrust Manager group.

For example, an LDAP group can be mapped to the system defined `Key Users` group in order to allow the LDAP group's members to encrypt data. Alternatively, an LDAP or OIDC group can be mapped to a newly created CipherTrust Manager group where group-based key permissions can be configured.

Group maps are specific to a CipherTrust Manager domain.

Connection Requirements

To set up group maps, the LDAP or OIDC connection must be configured with information to find a given user's group membership on the external server. LDAP connections require [six fields](#) to allow group maps. [OIDC connections](#) can rely on default or explicit configuration, as described in [Determination of OIDC Group Membership](#).

Note

Group maps are not applicable to the LDAP or OIDC connection available through connection manager.

Determination of OIDC Group Membership

By default, the OIDC connection is set up to allow, but not require, [group mapping](#). CipherTrust Manager checks for group membership in the ID token or returned from the UserInfo Endpoint. If no group membership can be obtained from either source, no CipherTrust Manager group is assigned.

By default, CipherTrust Manager checks the `groups` claim field in the ID token. When you create the OIDC connection, you can specify another claim field in the ID token which contains the group membership.

CipherTrust Manager first checks the claim field parameter in the ID token. If no group membership is found in the claim, and the OIDC connection is configured for authorization code flow, CipherTrust Manager then attempts to get group membership from the identity provider's UserInfo Endpoint. This attempt to get group membership from the UserInfo Endpoint is not supported for implicit flow.

CipherTrust Manager obtains the UserInfo endpoint from the configured Discovery URI, so there is no need to provide the UserInfo endpoint value. There is an optional `--userinfo-endpoint` parameter in the ksctl CLI to override this behavior, which is intended for test deployments only, and not for use in production.

Create a Group Map

1. Login to CipherTrust Manager as `admin` or another user in the **User Admins** group to the desired domain.
2. Navigate to **Access Management>Groups**.
3. Click the desired group name.
4. Click **+ Add Group Map**.
5. Provide the following configuration details:
 - **Strategy** - select `ldap` or `oidc` from the drop-down menu.
 - **Connection Name** - select the desired OIDC or LDAP connection from the drop-down menu.
 - **Connection Group Name** - type in the name for the OIDC or LDAP group.

-
6. Click **Add Group Map** to confirm.

View and Delete Group Maps for a Group

1. Login to CipherTrust Manager as `admin` or another user in the **User Admins** group to the desired domain.
2. Navigate to **Access Management>Groups**.
3. Click the desired group name.
4. View the currently mapped groups in the **Connection groups mapped to the <CipherTrust Manager group name> group**.
5. If desired, delete a groupmap by clicking the trash can icon to the right of the groupmap row.

Example Use Cases

The utility of group mappings is illustrated by the following examples. LDAP groups are shown in the examples, but OIDC groups can also apply to these scenarios.

Making All Users in a Specific LDAP Group Members of Key Users Group

Assume that there is a LDAP connection named `bababini` containing a group named `IT`. All users in the `IT` group should have the ability to create keys. This can be achieved by creating a group map that maps the LDAP `IT` group into the built-in CipherTrust Manager 'Key Users' group:

1. Login to CipherTrust Manager as `admin` or another user in the **User Admins** group.
2. Navigate to **Access Management>Groups**.
3. Click the 'Key Users' group name.
4. Click **+ Add Group Map**.
5. Provide the following configuration details:
 - **Strategy** - select `ldap` from the drop-down menu.
 - **Connection Name** - select the `bababini` connection from the drop-down menu.
 - **Connection Group Name** - type in `IT`.
6. Click **Add Group Map** to confirm.

Users in the LDAP group `IT` can now create keys.

Two LDAP Groups Share Keys

Assume that there is a LDAP connection named `bababini` containing two groups: `IT` and `Engineering`. It is desired to share cryptographic keys between the two LDAP groups. This can be achieved by the following steps:

1. Create a **user-defined group** on CipherTrust Manager called `it-engg-shared-keys`.
2. Create cryptographic keys and allow all users in the `it-engg-shared-keys` group **access to those keys**.
3. Create a group mapping between the IT and `it-engg-shared-keys`.
 - a. Click the `it-engg-shared-keys` group name.
 - b. Click **+ Add Group Map**.
 - c. Provide the following configuration details:
 - **Strategy** - select `ldap` from the drop-down menu.
 - **Connection Name** - select the `bababini` connection from the drop-down menu.
 - **Connection Group Name** - type in `IT`.
 - d. Click **Add Group Map** to confirm.
4. Create a group mapping between the Engineering and `it-engg-shared-keys` via the command
 - a. Click the `it-engg-shared-keys` group name.
 - b. Click **+ Add Group Map**.
 - c. Provide the following configuration details:
 - **Strategy** - select `ldap` from the drop-down menu.
 - **Connection Name** - select the `bababini` connection from the drop-down menu.
 - **Connection Group Name** - type in `Engineering`.
 - d. Click **Add Group Map** to confirm.

Users in both LDAP groups can now share all the keys granting access permissions to the `it-engg-shared-keys` group.

Services

The CipherTrust Manager contains operations for the management of CipherTrust Manager services. It supports operations such as a reset of the service (erases all data) and restart of a service (without erasing all data).

Note

The following operations can have a major impact on the availability of CipherTrust Manager and the data inside of it. We recommend taking a backup of your devices and information before using these operations.

The following are typical operations initiated using the **ksctl**. If you are using the API, see the "API playground".

Services Reset

This operation can be used to return a CipherTrust Manager to its initial state. The **Reset** operation will wipe all data from the CipherTrust Manager. This includes keys, users, and audit logs.

Warning

This is a destructive operation and will wipe all data in the CipherTrust Manager.

This operation requires authentication and an optional parameter called `--delay int` that defines the delay, in seconds, before the CipherTrust Manager initiates the operation.

To initiate a services reset

```
ksctl services reset
```

This returns the following response:

```
Please note that this operation could take some time.
```

```
WARNING - Reset is a destructive operation and will wipe all data in the CipherTrust Manager!  
Are you really sure you want to continue? [y/N] y  
Resetting CipherTrust Manager services...
```

Services Restart

This operation restarts the CipherTrust Manager without causing a hard reboot of the appliance. It requires authentication and an optional parameter called **'delay'** that defines the delay, in seconds, before the CipherTrust Manager initiates the operation.

This operation can be used in the following scenarios:

- Configuration changes
- General troubleshooting

To initiate a services restart

CLI

```
ksctl services restart
```

This returns the following response:

```
Please note that this operation could take some time and temporarily disrupt the
service.
Are you really sure you want to continue? [y/N] y
Restarting CipherTrust Manager services...
```

UI

1. Log on to the CipherTrust Manager GUI as administrator.
2. Navigate to **Admin Settings > Services**.
3. Click the **System Restart** button.

A message box appears prompt to confirm restart.

4. Click **Restart Services**.

Practice caution, this operation could take some time and temporarily disrupt the services.

To initiate a services restart by service-name

```
ksctl services restart --service-names <List-of-service-names>
```

Example (restarting `nae-kmip` service)

```
ksctl services restart --service-names nae-kmip
```

Example (restarting `web` service)

```
ksctl services restart --service-names web
```

Services Status

This operation reports the status of a CipherTrust Manager services.

To check on the status of each service

```
ksctl services status
```

This returns the following response:

```
{
  "status": "started",
  "services": [
    {
      "name": "nae-kmip",
```

```

    "status": "started"
  },
  {
    "name": "web",
    "status": "started"
  }
]
}

```

To check on the overall status of the services:

```
ksctl services status -o
```

This returns the following response:

```
"started"
```

Root of Trust Hardware Security Module

Note

This feature is not included in Community Edition and requires a valid Virtual CipherTrust Manager license. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

A Hardware Security Module (HSM) is a physical device that provides more secure management of sensitive data, such as keys, inside CipherTrust Manager.

CipherTrust Manager internally uses a chain of key encryption keys (KEKs) to securely store and protect sensitive data such as user keys. When an HSM is setup, the CipherTrust Manager uses a set of keys on the HSM partition that protect the KEKs chain and become the root of trust. In most cases, setting up an HSM generates new root of trust keys. If there are [keys already present on the HSM](#) with the correct attributes, those keys become root-of-trust keys.

Warning

As a security best practice, we strongly recommend dedicating CipherTrust Manager root of trust keys to that purpose, and not granting access to these keys to any other application or for any other purpose.

- **Luna Network HSM:** Do not use a Luna HSM partition containing the CipherTrust Manager root of trust key with CipherTrust Cloud Key Management (CCKM) or HSM-anchored domains. Do not create a connection to a Luna HSM root of trust partition with Connection Manager.
- **Google Cloud HSM:** Do not manage the Google key ring containing the root of trust key with CCKM. Do

not create a Google connection on Connection Manager with a service account key file that grants permission to root of trust keys.

If multiple CipherTrust Manager instances are configured to use the same HSM partition they end up using shared 'root of trust' keys.

After setting up an HSM to allow CipherTrust Manager to connect as a client, you can configure an HSM as the root-of-trust using the CipherTrust Manager GUI. This can also be done using the API or the [CLI toolkit **ksctl**](#), as shown below. It is assumed that you have added a CipherTrust Manager user and URL to the **ksctl** configuration file. Root-of-Trust HSM configuration requires you to login to the root **domain** as **admin** or another [Application Administrator](#).

You must set up the HSM to allow CipherTrust Manager to connect as a client, including creating necessary passwords, certificates, and string values. For more detailed information on creating these HSM objects, refer to [HSM product documentation](#).

Note

Using the cloud-init method to set up a root-of-trust is not recommended because it exposes sensitive information, such as the partition password, that can be viewed by anyone with access to the associated cloud console.

Note

HSM setup might fail if time of the system is out of sync. It is recommended to set up time sync using [NTP](#) before continuing.

With the CipherTrust Manager API or CLI toolkit **ksctl**, it is possible to:

- Setup HSMs
- Get and List detailed information on existing HSMs
- Add HSMs
- Delete HSMs

Refer to the CipherTrust Manager API and CLI online documentation for more information.

Supported HSMs

The CipherTrust Manager supports the following HSMs:

- Luna Network HSM version 7. See [Luna Network HSM supported versions and configurations](#) for more details.

-
- TCT Luna T-Series Network HSM
 - TCT Luna PCIe HSM
 - Luna PCIe HSM
 - AWS CloudHSM (Cavium)
 - Thales Data Protection On Demand (DPoD) Luna Cloud HSM Service
 - IBM Hyper Protect Cloud Services (HPCS)
 - TCT k160 High Assurance HSM
 - TCT k160 Standard HSM
 - Azure Dedicated HSM with a Virtual CipherTrust Manager which is also hosted on Microsoft Azure. This setup requires a [specific deployment architecture](#).
 - nShield Connect HSM
 - The supported firmware version is 13.4.5.
 - Only Operator Card Sets (OCS) are accepted for FIPS 140-2 level 3 compliant Security World. Therefore, this configuration is required for using nShield Connect HSM as a root of trust. Administrator Card Sets (ACS) are not accepted.
 - ProtectServer 3 PCIe HSM (Network Mode Only)

The supported firmware version is 7.01.00 and supported software version is 7.2.0.

- Google Cloud HSM

Hybrid HSM

Since release 1.3.0, any CipherTrust Manager can be clustered with another CipherTrust Manager, regardless of the HSM partition it is using, or even if it is not using an HSM partition at all. This is different from prior releases that required all CipherTrust Manager's in a cluster to be connected to the same HSM partition. When cluster nodes have different HSMs, it is called a "Hybrid HSM" configuration. When cluster nodes share the same HSM partition, it is called a "Shared HSM" configuration. There are several implications to a "Hybrid HSM" configuration that should be taken into consideration when deploying a cluster.

1. **System Security:** A system is only as secure as its weakest link. In a Hybrid HSM configuration, a CipherTrust Manager instance that is rooted in an HSM can be clustered with an instance that is not - this reduces the overall security of the system. The most secure configuration is to have all CipherTrust Manager instances use an HSM.
2. **Cluster Join Security:** When a system is joined to a cluster, it must receive certain keys that allow it to join the cluster. A shared HSM partition can protect these keys from ever being exposed. In this scenario, the join operation should use the **ksctl** utility's `--shared-hsm-partition` flag for the best security.

3. Backup Security: When a backup is created, there is a **ksctl** option called `--tied-to-hsm` that can tie the backup to the HSM of the instance. This is the most secure backup, but also limits it to being restored on a system with access to that specific HSM partition. By default, a backup is not tied to a specific HSM partition and can be restored to any compatible CipherTrust Manager system.

Supplying Existing Root of Trust Keys

By default, setting up an HSM automatically generates new root of trust keys. You can optionally create these keys directly on the HSM *before setting up the HSM with CipherTrust Manager*. When the HSM is later setup with CipherTrust Manager, the keys are automatically recognized and used to protect the KEK chain.

Caution

Do not manually create root of trust keys directly on the HSM **after** the HSM is set up on CipherTrust Manager. Instead, use the [CipherTrust Manager functions to rotate root of trust key](#) to generate new root-of-trust keys.

Manually creating root of trust keys directly on the HSM after CipherTrust Manager setup can result in the CipherTrust Manager application becoming unavailable after reboot, requiring customer support to recover.

You can supply existing root of trust key for any generally supported HSM type. For the embedded Luna PCIe HSM in the Thales CipherTrust Manager k570 appliance, existing keys can come from [restoring from a Luna Backup HSM](#).

Required Root of Trust Key Configuration

There are two keys required, an encryption key and an HMAC key for integrity validation.

Caution

In addition to the required characteristics, we strongly recommend to generate the keys with the non-exportable flag to ensure they never leave the boundary of the HSM.

The encryption key must have the following characteristics:

- AES-256 key algorithm.
- Permissions to encrypt and decrypt.
- The key name must be `DARKSTAR-encryption-key` or `<some_other_prefix>-encryption-key`.

Note

An encryption key named `DARKSTAR-encryption-key` immediately becomes the active root of trust key for CipherTrust Manager on HSM setup. An encryption key named with a different prefix must be **rotated** to become active.

The HMAC key must have the following characteristics:

- HMAC-256 key algorithm.
- Permissions to sign and verify.
- The key name must be `DARKSTAR-mac-key` or `<some_other_prefix>-mac-key`.

Note

The HMAC key's prefix must match the encryption key's prefix.

Example Root of Trust Key PKCS#11 Configuration

This is an example of the recommended PKCS#11 attributes and values for the encryption key:

PKCS#11 Attribute	Value
CKA_CLASS	CKO_SECRET_KEY
CKA_KEY_TYPE	CKK_AES
CKA_LABEL	<prefix>-encryption-key
CKA_VALUE_LEN	32
CKA_PRIVATE	CK_TRUE
CKA_TOKEN	CK_TRUE
CKA_SENSITIVE	CK_TRUE
CKA_ENCRYPT	CK_TRUE
CKA_DECRYPT	CK_TRUE
CKA_EXTRACTABLE	CK_FALSE

This is an example of the recommended PCKS#11 attributes and values for the HMAC key:

PKCS#11 Attribute	Value
CKA_CLASS	CKO_SECRET_KEY
CKA_KEY_TYPE	CKK_GENERIC_SECRET
CKA_LABEL	<prefix>-mac-key
CKA_VALUE_LEN	32
CKA_PRIVATE	CK_TRUE
CKA_TOKEN	CK_TRUE
CKA_SENSITIVE	CK_TRUE
CKA_SIGN	CK_TRUE
CKA_VERIFY	CK_TRUE
CKA_EXTRACTABLE	CK_FALSE

Setting up an HSM

Creating the first HSM requires that you supply the hsm type and connection information (connInfo).

The general **ksctl** hsm setup command is:

```
$ ksctl hsm setup <hsm type> <config parameters>
```

The valid options for hsm type are:

- `luna` for Luna Network HSM,
- `lunapci` for the embedded Luna PCIe HSM and TCT Luna PCIe HSM in CipherTrust Manager k570 models,
- `lunatct` for TCT Luna T-Series Network HSM,
- `aws` for AWS CloudHSM,
- `dpod` for DPoD Luna Cloud HSM Service,
- `ibm-hpcs` for IBM Hyper Protect Cloud Services (HPCS),
- `k160` for TCT k160 HSM, both the High Assurance and Standard models
- `nshield` for nShield Connect HSM
- `protectserver` for ProtectServer 3 PCIe HSM
- `gcp-cloud-hsm` for Google Cloud HSM

The configuration info (config parameter) depends on the hsm type and is described below. You must set up the HSM to allow CipherTrust Manager to connect as a client, including creating necessary passwords, certificates, and string values. For more detailed information on creating these HSM objects, refer to HSM product

documentation.

Note

If *Luna HSM policy 21 (Force user PIN change after set/reset)* is enabled, you must first login to the HSM and change the password before connecting to CipherTrust Manager. The resulting error message does not indicate that a password change is required.

Note

CipherTrust Manager is reset after adding the HSM, with a delay in seconds (default: 5) specified before the reset is performed.

Caution

Be aware that performing a reset is destructive; it deletes all data in the CipherTrust Manager and so must be used with great care. You must [backup](#) all information you wish to retain.

This backup is particularly important if you created users and keys before setting up an HSM. For example, you might create users and keys without an HSM as part of the Community Edition before applying a Virtual CipherTrust Manager license or starting a trial evaluation. If you wish to retain those objects, you must back them up before setting up an HSM root of trust.

Luna Network HSM and TCT Luna T-Series Network HSM (hsm type: luna, lunatct)

For the Luna Network HSM or TCT Luna T-Series Network HSM, the required parameters for initial configuration are:

- `hsm-host`: IP or hostname of the HSM
- `partition-name`: The name of the HSM partition to use
- `partition-password`: The password of the initial partition to use. For Luna 7 and most Luna 6, this is the Crypto Officer role password or challenge secret. For TCT Luna T-Series Network HSM, this is the partition password or challenge secret. This password will also be used if more servers and high-availability (HA) mode is used, in which case all HSM must have the same password.
- `serial`: Serial number of the partition to use
- `delay`: Delay in seconds before reset, defaults to 5 seconds
- `reset`: Reset the CipherTrust Manager instance as part of operation

As well, you are required to provide some certificates and other parameters to establish trust with the connection. The parameters depend on whether you are setting up a Network Trust Link (NTLS) or a Secure Trusted Channel (STC) between CipherTrust Manager and the HSM partition.

Note

You provide certificates generated on an external Luna HSM Client. To ensure compatibility with this CipherTrust Manager version, we strongly recommend generating certificates on Luna HSM Client version 10.7 only.

Parameters for Luna Network HSM NTLS Connection or TCT Luna T-Series Network HSM Connection

- `- server-cert-file`: File containing server certificate in PEM format.

Note

For Luna Network HSM, you can supply a server certificate signed by a trusted certificate authority (CA) external to the Luna Network HSM. For this type of server certificate, append all the certificates in the signing chain to the certificate file.

Externally signed server certificates are not supported for TCT Luna T-Series Network HSMs.

- `- client-cert-file`: File containing client certificate in PEM format. This is the client certificate that was used to register the client on the Luna Network HSM with the `lunash:> client register` command.

Note

Both self-signed and externally signed certificates are accepted. In either case, you need only provide the client certificate and client private key to CipherTrust Manager. For externally signed client certificates, the CA certificates must have been provided to the Luna Network HSM with the `lunash:> client addCA externalCA` command.

- `- client-cert-key-file`: File containing client private key in PEM format

Parameters for Luna Network STC Connection

- `- server-cert-file`: File containing server certificate in PEM format

Note

You must provide a self-signed certificate.

- `-partition-identity-file`: File containing the partition identity. This file is obtained as part of a sequence on the Luna Network HSM to prepare the STC connection.

Note

The [Luna Network HSM Documentation](#) provides detailed information to configure values for `hsm-host`, `partition-name`, `partition-password`, `serial`, `server-cert-file`, `client-cert-file`, and `client-cert-key-file`.

Following is a full example for a Luna Network HSM setup command using the NTLS connection:

```
$ ksctl hsm setup luna --reset --partition-name "partition name" --partition-pass
word "sOmeP@ssword" --hsm-host 192.168.0.1 --serial 1234 --server-cert-file serve
r_cert.pem --client-cert-file client_cert.pem --client-cert-key-file client_cer
t_key.pem
```

Luna Network HSM Supported Versions and Configurations

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.

Caution

This list of Luna Network HSM supported versions is specific to this CipherTrust Manager version. If you are at a lower version of CipherTrust Manager, some of these HSM versions might be incompatible. For example, CipherTrust Manager versions 2.16 or lower are incompatible with Luna firmware 7.8.4 or higher. Be sure to consult the appropriate CipherTrust Manager version of this page to check supported HSM versions before configuring root of trust. If you attempt to configure incompatible Luna Network HSM firmware and CipherTrust Manager software versions, the root of trust configuration is lost and the CipherTrust Manager becomes unavailable.

Using Externally Signed Certificates for Luna Network HSM NTLS Connection

Note

Externally signed certificates are not supported for TCT Luna T-Series Network HSMs or for the STC connection type.

You can choose to provide both server and client certificates which are signed by a trusted CA, external to the Luna Network HSM and CipherTrust Manager, instead of self-signed certificates.

- **Server certificate:** Provide the full certificate chain in a file with the `server-cert-file` parameter when setting up the first Luna Network HSM with CipherTrust Manager.
- **Client certificate:** As part of the client registration on the Luna Network HSM, the CA certificates must have been provided to the Luna Network HSM with the `lunash:> client addCA externalCA` command. You do not provide the CA certificates to CipherTrust Manager, only the client certificate itself and the client private key.

Set up a Secure Trusted Channel Connection

1. **Create the STC connection to the Luna Network HSM.** As part of that process, you create a Luna partition with the LunaSH CLI, and initialize a Luna partition for STC with a Luna Client. Be sure to retain the Partition

SO password.

Note

The partition size must be at least 1536 bytes to accommodate two client registrations and the CipherTrust Manager root keys. If you plan to rotate the root keys, make the partition large enough to accommodate 512 more bytes storage per rotation.

You can alternatively convert an existing NTLS partition to use STC.

2. On CipherTrust Manager, obtain the client identity file needed to register CipherTrust Manager as an additional client.

```
ksctl hsm clients stcidentity download [--outputfile <download-filename>]
```

Alternatively, use `GET /v1/system/hsm/clients/stcidentity/download/` in the REST API.

3. Use the client identity file to connect CipherTrust Manager to the STC partition as an additional client. More details are available in Luna documentation about the Partition SO's actions to [allow an additional client access to the STC partition](#).
 - a. Login to the initial client as the Partition SO.
 - b. Supply the client identity file as part of the LunaCM command `stcconfig clientregister -label <client_label> -file <path/client_ID_file>`.

Note

Transfer the downloaded client identity file through SCP or other means. Do not copy and paste its contents to another file, and use this second file to execute `stcconfig clientregister -label <client_label> -file <path/client_ID_file>`. Client registration fails in this case.

- c. Export the partition identity public key using the LunaCM command `stcconfig partitionidexport`.
 - d. Obtain the HSM server certificate, located in `<Lunaclient_install_directory>/cert/server/<hostname/IP>Cert.pem`
4. On CipherTrust Manager, register the partition.

```
ksctl hsm setup luna --reset --partition-name <partition-name> --partition-password <Crypto_Officer_role_password_or_challenge_secret> --hsm-host <HSM_host_IP> --serial <partition_serial_number> --server-cert-file <HSM_server_certificate> --partition-identity-file <partition_identity_filepath>
```

For example:

```
$ ksctl hsm setup luna --reset --partition-name "partition name" --partition-password "sOmEP@ssword" --hsm-host 192.168.0.1 --serial 1234 --server-cert-file server_cert.pem --partition-identity-file 1234.pid
```

Embedded Luna PCIe HSM or TCT Luna PCIe HSM (hsm type: lunapci)

Note

While the HSM type for both the Luna PCIe HSM or TCT Luna PCIe HSM is the same, the value required for the `partition-name` parameter is different. The Thales CipherTrust Manager k570 contains a Luna PCIe HSM, and the Trusted Cyber Technologies CipherTrust Manager k570 contains a TCT Luna PCIe HSM.

Caution

If you intend to [restore master keys from a Backup HSM](#), restore the keys **before** performing root of trust configuration.

Configuring Luna PCIe HSM as the root of trust and then restoring master keys can result in the CipherTrust Manager application becoming unavailable after reboot, requiring customer support to recover.

For the embedded Luna PCIe HSM in CipherTrust Manager k570 models, the required parameters for initial configuration are:

- `partition-name`: The name of the HSM partition to use. For TCT Luna PCIe HSM, this is the HSM label.
- `partition-password`: The password of the initial partition to use. For PED-authenticated HSMs, this is the challenge password.
- `delay`: Delay in seconds before reset, defaults to 5 seconds
- `reset`: Reset CipherTrust Manager instance as part of operation

Following is a full example for a Luna PCIe HSM setup command:


```
$ ksctl hsm setup lunapci --reset --partition-name "partition name" --partition-password "s0meP@ssword"
```

The `ksadmin` system administrator can [upgrade firmware](#) for Luna PCIe HSM to take advantage of performance and security fixes for the root of trust keys.

AWS CloudHSM (Cavium) (hsm type: aws)

Prerequisite - Prepare an instance of CloudHSM in AWS:

- Prior to configuring CipherTrust Manager instance to use AWS CloudHSM service, an instance of CloudHSM must be prepared in AWS.

Refer to AWS CloudHSM documentation to create, initialize and activate a CloudHSM cluster. This step involves running a separate AWS client instance to connect and setup HSM users as well as sign the cluster CSR of your cloudhsm cluster instance. Upon completion of this step you will have a cloudhsm cluster in "Active" state. See [AWS documentation](#)

Warning

If you plan to have more than one AWS CloudHSM in the AWS Cluster, you must [add each additional CloudHSM to CipherTrust Manager](#) as soon as possible.

Otherwise, if the first CloudHSM is not available on root-of-trust service restart, the service does not recognize the additional CloudHSMs, and CipherTrust Manager users are locked out of all interfaces with no root-of-trust available.

- Create a new cryptouser on your new cloudhsm instance. Similar to the cluster activation step when you reset the CO password, use the `cloudhsm_mgmt_util` utility on the AWS client instance to create a new user with cryptouser role (CU). CipherTrust Manager will communicate to the HSM instance on behalf of this CU user you setup.
- Be sure that the AWS CloudHSM is accessible to the CipherTrust Manager through the network. To achieve this for a [Virtual CipherTrust Manager deployed on an AWS EC2 instance](#), attach the cluster security group to the EC2 instance as described in [AWS documentation](#). Check the [hardening guidelines](#) for more deployments and details.

For AWS CloudHSM (Cavium), the required parameters for initial configuration are:

- `- partition-name`: (Optional) The name of the HSM partition to use for CloudHSM. Defaults to "cavium"
- `- cryptouser-password`: Specify the credentials of a cryptouser in the form of "username:password"
- `- hsm-host`: IP Address or host name of cloudhsm cluster
- `- server-cert-file`: This is the CA public certificate that was used to sign the CloudHSM Cluster CSR

during the cluster initialization. This is the certificate that signed the Cluster certificate. Refer to: <https://docs.aws.amazon.com/cloudhsm/latest/userguide/initialize-cluster.html>

- `delay`: Delay in seconds before reset, defaults to 5 seconds
- `reset`: Reset CipherTrust Manager instance as part of operation

Following is a full example of an AWS CloudHSM setup command:

```
$ ksctl hsm setup aws-cloud-hsm --reset --cryptouser_password "sOmeP@ssword" --hsm-host 192.168.0.1 --server-cert-file server_cert.pem
```

Thales Data Protection On Demand (DPoD) Luna Cloud HSM service (hsm type: dpod)

The CipherTrust Manager integration with Luna Cloud HSM service uses JWT-based authentication, which has been required for all clients to this service since December 31, 2021.

Note

The older authentication based on the CVAppSpecificData is no longer available on CipherTrust Manager.

You must provide the AuthTokenConfigURI, AuthTokenClientId, and the AuthTokenClientSecret values from the service client package's Chrystoki.conf file.

Prerequisites:

- You have installed a service client and provisioned the Luna Cloud HSM Service.
- The connection information (partition-name and co-password) is obtained while provisioning your Luna Cloud HSM service.
- The configuration information is obtained from the HSM client package. This includes certificate files and authentication values in the Chrystoki.conf file, and the cv-partition-data value from crystoki-template.ini.

For DPoD, the required parameters for initial configuration are:

- `--partition-name`: The name of the partition configured during partition initialization
- `--co-password`: The password of the Crypto Officer
- `--server-cert-file`: Path to server-certificate.pem file
- `--partition-cert-file`: Path to partition-certificate.pem file
- `--partition-ca-cert-file`: Path to partition-ca-certificate.pem file
- `--cv-partition-data`: PartitionData00 value from crystoki-template.ini

- `- auth_token_config_uri`: AuthTokenConfigURI value from Chrystoki.conf
- `--auth-token-client-id`: AuthTokenClientId value from Chrystoki.conf
- `--auth-token-client-secret`: AuthTokenClientSecret value from Chrystoki.conf
- `--delay`: Delay in seconds before reset, defaults to 5 seconds
- `--reset`: Reset CipherTrust Manager instance as part of operation

The following is a full example for a DPoD Luna Cloud HSM setup command:

```
$ ksctl hsm setup dpod --reset --partition-name "partition name" --co-password "s
OmeP@ssword" --server-cert-file "server-certificate.pem" --partition-cert-file "p
artition-certificate.pem" --partition-ca-cert-file "partition-ca-certificate.pem"
--cv-partition-data "crystoki-template-PartitionData00" --auth-token-config-uri
"https://111111.uaa.system.snakefly.dpsas.io/.well-known/openid-configuration" --
auth-token-client-id "111111-1111-1111-1111-111111111111" --auth-token-client-se
cret "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
```

IBM Cloud Hyper Protect Crypto Services

CipherTrust Manager is able to use IBM Cloud Hyper Protect Crypto Services (HPCS) as a root of trust through IBM Cloud HPCS's PKCS#11 API. To establish a connection between CipherTrust Manager and an HPCS PKCS#11 endpoint, you must perform the following configuration on IBM:

1. **Create PKCS#11 user roles.** Following IBM best practices, CipherTrust Manager requires a **Normal User** API Key with **Key operator** privileges.
2. **Download the IBM PKCS#11 module and configure it.** As part of this configuration, you create a private keystore, also called a `tokenspace` for CipherTrust Manager to access. Note the following values, as they need to be provided to CipherTrust Manager to access the HPCS instance:
 - `instance_id`
 - `EP11_endpoint_URL` CipherTrust Manager requires the public URL.
 - `EP11_endpoint_port_number`
 - `private_key_store_spaceid`
3. Initialize the private keystore with the **Security Officer(SO)** user. The SO performs a `C_InitToken` operation. For example, to do this using the `opensc pks11-tool`:

```
sudo apt install opensc
pks11-tool --module=/usr/local/lib/pkcs11-grep11-amd64.so --init-token --labe
l <you-can-choose> --so-pin=<your-so-api-key>
```

On CipherTrust Manager, you must provide the following configuration parameters:

- **host**: The hostname part of the public URL for the Enterprise PKCS#11 endpoint, called `EP11_endpoint_URL` in HPCS.
- **port**: The port for the Enterprise PKCS#11 endpoint, called `EP11_endpoint_port_number` in HPCS.
- **instance-id**: The id for the HPCS instance. There multiple ways to retrieve this value in [IBM Cloud](#).
- **api-key**: The api key for the Normal User with Key operator privileges.
- **token-space-id**: The 128-bit Universally Unique IDentifier (UUID) of the private key store accessible by the **Normal User**. This value is called `private_key_store_spaceid` in IBM HPCS.

The following is a full example for an IBM HPCS setup command:

```
$ ksctl hsm setup hsm setup ibm-hpcs --reset -m <EP11_endpoint_public_host> --port <EP11_endpoint_port> --instance-id <instance_id> --api-key <normal_user_api_key> --token-space-id <private_key_store_spaceid>
```

TCT k160 HSM (hsm type: k160)

For the CipherTrust Manager TCT k160 High Assurance HSM and CipherTrust Manager TCT k160 Standard HSM models, the required parameters for initial configuration are:

- **delay**: Delay in seconds before reset, defaults to 5 seconds
- **reset**: Reset CipherTrust Manager instance as part of operation

You must also specify whether the model is High Assurance or Standard.

- **ha**: Specifies the High Assurance version of the k160 model.
- **std**: Specifies the Standard version of the k160 model.

Following is a full example for a k160 HA HSM setup command:

```
ksctl --configfile <k160>.yaml hsm setup k160 --reset --ha
```

Following is a full example for a k160 Standard HSM setup command:

```
ksctl --configfile <k160>.yaml hsm setup k160 --reset --std
```

Note

The process may take longer than the default 30 seconds timeout, so extending the timeout value in the yaml config file will help with the command response. Setting the "KSCTL_TIMEOUT:" value to 180 seconds is recommended.

Note

If you attempt to log in on the GUI, you will see various messages about services starting up and it will take over 10 minutes to complete the start-up process.

Once the CipherTrust Manager TCT k160 HSM is added, you can [register additional Token HSMs](#).

nShield Connect HSM

Before setting up an nShield Connect HSM as a root of trust to CipherTrust Manager, you must perform the following configuration in nShield utilities, as described in nShield documentation:

1. On the nShield Connect HSM, add the CipherTrust Manager IP address as a client. Apply the following settings:
 - Set up an unprivileged connection, as is default
 - Enroll the client without secure authentication
2. Configure the Remote File System (RFS) to accept access by the CipherTrust Manager as a cooperating client.

Note

You do not need to install or use nShield client software on CipherTrust Manager.

3. Create a new Security World if it doesn't exist yet.
4. Copy or note down the values for the following nShield HSM attributes:
 - Electronic Serial Number (ESN)
 - HKNETI (Hash of KNETI key)
 - nShield HSM IP address
 - RFS IP address
 - nShield HSM port, if different than the default 9004 value.
 - RFS port, if different than the default 9004 value.

If you want to use an existing softcard for CipherTrust Manager, copy or note down its name and pass phrase.

To set up an nShield Connect HSM as root of trust on the CipherTrust Manager web console:

-
1. Login to the root domain as `admin`.
 2. Navigate to **Admin Settings>Root of Trust**.
 3. Click **Yes, configure an HSM**.
 4. Select **nShield HSM** and click **Next**.
 5. Provide a **Softcard Name** and **Softcard Password**.

You can either provide these values for an existing softcard in the Security World, or you can enter new values to create a new softcard in the Security World.

6. Provide values obtained from the nShield HSM:

- ESN (Electronic Serial Number)
- IP of nShield Connect
- HKNETI
- RFS Host, meaning the IP address.

If the nShield port or RFS port are different than the default value of 9004, you must provide the port number(s) as well.

7. Click **Next**.
8. Confirm loss of key material and system restart to set up the nShield HSM.

ProtectServer 3 PCIe HSM

The following prerequisite configuration must have taken place on the ProtectServer HSM:

- [Initial Configuration](#) including creation and initialization of at least one user slot and token.
- A Certificate trust chain must be established for the ProtectServer HSM.
- [Compatible security flags](#) must be set for the HSM. CipherTrust Manager requires the following security flags:
 - Auth Protection
 - FIPS Algorithms only
 - Mode Locked
 - No Clear PINs
 - No Public Crypto
 - Tamper Before Upgrade

You can set all these flags with the command `ctconf -fualnct`.

To obtain values needed to set up ProtectServer 3 PCIe HSM as a root of trust to CipherTrust Manager:

1. Install the ProtectToolkit-C 7 software, including the C SDK.

Note

As part of this software installation, configure the ProtectServer 3 PCIe HSM for **network mode**. This is the only operating mode supported with CipherTrust Manager.

2. Import the ProtectServer server certificate to the client. The directory location of the ProtectServer Identity Cert (PIC) is displayed.
3. Copy the certificate from the directory, or note the file location, as you need to provide the server certificate file for CipherTrust Manager configuration.
4. Obtain the token serial number and User PIN values for the user slot, with the label listed as HSMTOKEN. The User PIN was set at token initialization. The token serial number can be obtained with the following command:

```
ctstat -s <user_slot_number>
```

To set up a ProtectServer 3 PCIe HSM as a root of trust on CipherTrust Manager, the required parameters are:

- `- hsm-host`: The IP address or hostname of the server containing the ProtectServer PCIe HSM.
- `- port`: The port of the ProtectServer PCIe HSM
- `- token-serial`: The user token serial number
- `- token-password`: The user PIN for the token.
- `- server-cert-file`: Path to ProtectServer HSM server certificate file in PEM format.
- `- delay`: Delay in seconds before reset, defaults to 5 seconds
- `- reset`: Reset CipherTrust Manager instance as part of operation

The following is a full example for a ProtectServer setup command:

```
$ ksctl hsm setup hsm setup protectserver --reset --hsm-host <protectserver_ip_address_or_hostname> --port <protectserver_port> --token-serial <token_serial_number> --token-password <user_pin> --server-cert-file <path_to_server_certificate_file>
```

Google Cloud HSM

CipherTrust Manager can use the Cloud HSM available in Google Cloud Platform (GCP) as a root of trust.

Note

You can only [rotate the root of trust key](#) on CipherTrust Manager. You cannot rotate the root of trust key on Google Cloud Platform, as CipherTrust Manager only recognizes the first version of a Google Cloud key.

Prerequisites

On Google Cloud Platform, you require the following Google Cloud objects:

- a service account
- a private key file for the service account. This file contains the service account private key in JSON format, along with other values associated with the service account. Consult Google Cloud documentation to [generate this file](#).
- a key ring in a [Google location that supports Google Cloud HSM](#).

The key ring and service account can be in the same or in separate projects.

The service account requires the following [Identity and Access Management \(IAM\)](#) permissions:

- `cloudkms.keyRings.get` for the location of the key ring.
- `cloudkms.locations.generateRandomBytes` for the location of the key ring.
- `cloudkms.cryptoKeys.list` for the key ring.
- `cloudkms.cryptoKeyVersions.list` for the key ring.
- `cloudkms.cryptoKeyVersions.manageRawAesCbcKeys` for the key ring.
- `cloudkms.cryptoKeyVersions.useToDecrypt` for the key ring.
- `cloudkms.cryptoKeyVersions.useToEncrypt` for the key ring.
- `cloudkms.cryptoKeyVersions.useToSign` for the key ring.
- `cloudkms.cryptoKeyVersions.useToVerify` for the key ring.
- `cloudkms.cryptoKeys.create` for the key ring.
- `cloudkms.cryptoKeyVersions.destroy` for the key ring.

Setup on CipherTrust Manager Web Console UI

1. Login to the root domain as `admin`.
2. Navigate to **Admin Settings>Root of Trust**.
3. Click **Yes, configure an HSM**.
4. Select **Google Cloud HSM** and click **Next**.
5. Provide the following required values:
 - **Location:** The Google Cloud location associated with the key ring. The location must support Cloud HSM.
 - **Key Ring:** The name of the key ring associated with the Google Cloud HSM.
6. Provide the private key file values associated with the service account. Consult Google Cloud documentation to [generate this file](#). You can either upload the file itself or paste the file contents into the text field.
 - Select **File Upload** and click **Upload Key File**.

or

 - Select **Text** and paste in the content of the private key file.
7. If your key ring is in a different project than your service account, specify the **Project ID**. By default, CipherTrust Manager uses the project ID associated with the service account and included in the service account key file.
8. Click **Next**.
9. Confirm loss of key material and system restart to set up the HSM.

Parameters for ksctl CLI setup

To setup a Google Cloud HSM in the CLI, the required parameters are:

- `location`: The Google Cloud location associated with the key ring. The location must support Cloud HSM.
- `keyring`: The name of the key ring associated with the Google Cloud HSM.
- `service-account-key-file`: A file containing the service account private key in JSON format, along with other values associated with the service account. Consult Google Cloud documentation to [generate this file](#).
- `reset`: Reset CipherTrust Manager instance as part of operation

Optional parameters are:

- `delay`: Delay in seconds before reset. Defaults to 5 seconds

- `project-id`: ID of the Google Project which contains the key ring. Specify this value if your key ring is in a different project than your service account. Defaults to the project ID associated with the service account and included in the service account key file.

The following is a full example for a Google Cloud HSM setup command:

```
$ ksctl hsm setup hsm setup gcp-cloud-hsm --reset --location <key_ring_location>
--keyring <key_ring_name> --service-account-key-file <path_to_service_account_key_file>
```

Getting and Listing HSMs

`Get` and `List` return detail information about the instance(s) added previously, `Get` returns a single instance and `List` returns an array of data.

Following is a full example of a `HSM Servers List` command:

```
$ ksctl hsm servers list

{
  "total": 2,
  "resources": [
    {
      "id": "2f81163b-f53d-45c1-b151-9bc333ce3a04",
      "type": "luna",
      "config": {
        "host": "192.168.0.1",
        "serial": "1234",
        "server-cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICAT
E-----"
      }
    },
    {
      "id": "be942b82-b00e-4f73-9081-28186e1580e6",
      "type": "luna",
      "config": {
        "host": "192.168.0.2",
        "serial": "4321",
        "server-cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICAT
E-----"
      }
    }
  ]
}
```

Adding an HSM

For supported HSMs, you can add additional HSMs of the same type as the initial HSM to provide redundancy for the root of trust keys.

This operation is only supported by the Luna Network HSM, Luna T-Series HSM, nShield Connect HSM, and AWS CloudHSM.

Note

To add extra AWS CloudHSM instances to the existing HSM cluster, no change is required on CipherTrust Manager; consult AWS CloudHSM documentation.

Adding an Additional nShield Connect HSM

A member of the `admins` group, such as `admin`, can add a new nShield Connect HSM in the root domain.

nShield Connect HSMs must already share the same Security World and RFS before you add the HSM to CipherTrust Manager, so that the root of trust keys can successfully replicate between all the root of trust HSMs. You require the nShield HSM hostname or IP address, the Electronic Serial Number (ESN), and the HKNETI (Hash of KNETI key). If the nShield HSM port is different than the default 9004 value, you must provide this value.

CipherTrust WebUI

1. Navigate to **Admin Settings>Root of Trust**.
2. Click **+ Add**.
3. Read the Introduction text and click **Next**.
4. Enter the **ESN (Electronic Serial Number)**, **IP of nShield**, and **HKNETI** values, being careful to enter the values exactly.
5. If the port number is different than the default 9004 value, enter the **Port of nShield**.
6. Click **Next**.
7. Confirm the operation by clicking **Add HSM**.
8. Repeat steps 2-7 for each nShield HSM you wish to add.

ksctl CLI

The command syntax varies depending on whether you need to supply a port value, and which operating system ksctl is run from. If no port is specified, the default value of 9004 is set.

Command syntax for default port value on any operating system

```
ksctl hsm servers add --hsm-host <nShield_IP_or_hostname> --serial <ESN> --hkneti <hkneti_value>
```

Command syntax for nondefault port value on Linux

```
ksctl hsm servers add --config '{"host":"<hostname_or_ip_address>","port":"<por
```

```
t_value>","serial":"<ESN>","hkneti":"<hkneti_value>"}
```

Command syntax for nondefault port value on Windows

```
ksctl hsm servers add --config "{\"host\":\"<hostname_or_ip_address>\",\"port\":\"<port_value>\",\"serial\":\"<ESN>\",\"hkneti\":\"<hkneti_value>\"}"
```

Adding an Additional Luna Network HSM partition or Luna T-Series HSM partition

Adding subsequent HSM partitions to a CipherTrust Manager inserts them into the HA group. The first `hsm servers add` command that you do automatically creates an HA group.

If you are adding multiple HSM partitions from the same Luna Network HSM, they must all use the same type of network channel, NTLS or STC. That means if your initial root-of-trust connection uses NTLS, you can only add partitions that use NTLS from the same HSM. Similarly, if your initial root-of-trust connection uses STC, you can only add partitions that use STC from the same HSM.

STC Prerequisites

More prerequisites are required to add a Luna Network HSM partition with an STC connection:

- You must obtain the CipherTrust Manager client identity file

```
ksctl hsm clients stcidentity download [--outputfile <download-filename>]
```

Alternatively, use `GET /v1/system/hsm/clients/stcidentity/download/` in the REST API.

- As the Partition SO, supply the client identity file to the partition as part of the LunaCM command `stcconfig clientregister -label <client_label> -file <path/client_ID_file>`.

Note

Transfer the downloaded client identity file through SCP or other means. Do not copy and paste its contents to another file, and use this second file to execute `stcconfig clientregister -label <client_label> -file <path/client_ID_file>`. Client registration fails in this case.

- Obtain the partition identity file using the LunaCM command `stcconfig partitionidexport`.

More details are available in Luna documentation about the Partition SO's actions to [allow an additional client access to the STC partition](#).

Parameters to Add a Luna HSM

Adding a Luna HSM requires that you supply the configuration information.

The configuration information (config) is a JSON blob; it has the following keys:

- host
- serial
- server-cert

Note

For Luna Network HSM, you can provide a server certificate signed by a trusted CA external to the Luna Network HSM. For CipherTrust Manager to accept an externally signed server certificate, you must have supplied the full certificate chain when you [setup the first HSM](#) for CipherTrust Manager.

Externally signed server certificates are not supported for TCT Luna T-Series Network HSMs.

- forceClear (optional)

When set to "true", forcefully clears all existing data on the partition of the joining node before adding it to the HA group. Make sure the existing objects are not used elsewhere. They cannot be recovered after the deletion. Use with caution.

- forceCopy (optional)

When set to "true", all existing objects on the joining partition are retained and propagated within the HA group. This option can be useful if the joining partition is shared among CipherTrust Manager instances that use the partition only as a member of a current HA group. This option can also be useful if the joining partition is shared among different applications other than CipherTrust Manager, and the goal is to retain the existing security objects.

Do not use the forceCopy option if the joining partition was previously used as root of trust for other CipherTrust Manager instance(s) and that it was not part of the current HSM HA group. Using this option in such cases can break the existing CipherTrust Managers that use the HA group servers. Use with caution.

- partition-identity-file

For adding an additional STC connection only. See [STC prerequisites](#) for additional required steps to add an HSM partition with an STC connection.

Following is a full command example for adding a Luna Network HSM:

```
$ ksctl hsm servers add --config '{ "host": "192.168.0.2", "serial": "4321", "ser
```

```
ver-cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----" }'
```

If the joining HSM partition contains data, the operation will fail unless `forceClear` or `forceCopy` flags are specified. The following example command shows how to forcefully add a partition by first wiping all its data:

```
$ ksctl hsm servers add --config '{ "host": "192.168.0.2", "serial": "4321", "forceClear": "true", "server-cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----" }'
```

Warning

DO NOT use `forceClear` and `forceCopy` flags by default when adding a new HSM server. If the command fails indicating the joining node contains objects:

1. Manually log on to the partition.
2. Carefully investigate the use of existing objects.
3. Repeat the command with `forceClear` or `forceCopy` flag.

Adding an Additional AWS CloudHSM

You can add an additional AWS CloudHSM in the same AWS cluster as the existing root-of-trust AWS CloudHSM.

Adding every AWS CloudHSM in the AWS cluster to CipherTrust Manager guarantees that CipherTrust Manager can recognize and connect to any CloudHSM when the cluster fails over.

Warning

We recommend adding all cluster CloudHSMs to CipherTrust Manager as soon as possible.

Otherwise, when the root-of-trust service restarts, CipherTrust Manager might not recognize the available CloudHSMs and CipherTrust Manager users are locked out of all interfaces with no root-of-trust available.

CipherTrust WebUI

1. Prepare the CloudHSM(s) in the AWS cluster that contains the initial root-of-trust AWS CloudHSM. You require the HSM Elastic Network Interface (ENI) IP for the new AWS CloudHSM(s).
2. Login to the root `domain` as a member of the `admins` group, such as `admin`.
3. Navigate to **Admin Settings>Root of Trust**.
4. Add a new AWS CloudHSM.
 - a. Click **+ Add**.
 - b. Read the Introduction text and click **Next**.

- c. Enter the **HSM ENI IP Address**, being careful to enter the value exactly.
- d. Click **Next**.
- e. Confirm the HSM ENI IP address you are adding. Click **Add HSM**.
- f. Repeat steps a-e for each CloudHSM you wish to add.

ksctl CLI

First, prepare the CloudHSM(s) in the AWS cluster that contains the initial root-of-trust AWS CloudHSM. You require the HSM Elastic Network Interface (ENI) IP for the new AWS CloudHSM(s).

A member of the `admins` group, such as `admin`, can then add a new AWS CloudHSM in the root `domain`.

Command syntax for adding an AWS CloudHSM:

```
ksctl hsm servers add --hsm-host <HSM_ENI_IP>
```

Be careful to provide the HSM ENI IP value exactly.

Repeat this command for each AWS CloudHSM you are adding.

Deleting an HSM

`Delete` will remove the record for the `--id` provided, with optional parameters for `reset` and `delay`. Just as in the Setup function, deleting the last HSM requires a reset of the CipherTrust Manager. After the `reset` operation, newly encrypted data is longer rooted to an HSM.

Note

Just as in the Setup function (see [Setting up an HSM](#)) after deleting the last HSM, a reset of the CipherTrust Manager must be performed before the HSM is reused.

Warning

Performing a CipherTrust Manager reset is destructive; it deletes all information in the system and so must be used with great care. You must backup all information you wish to retain.

Following is a full example for the `delete` command:

```
$ ksctl hsm servers delete --id ce6da730-c266-44f8-a0f6-97e49c7596eb --reset
```

Register an Additional Token HSM for TCT k160

The CipherTrust TCT k160 allows for additional Token HSMs to be initialized.

Keeping the first token inserted into the k160, insert another token into an open USB slot on the back of the k160 device and issue the following command:

```
ksctl --configfile <k160>.yaml hsm tokens reg
```

Once the command completes, remove the additional token from the back of the k160.

Note

Removing the token from the front of the k160 will cause a system restart.

Rotating the Root of Trust Key

Application administrators logged into the [root domain](#) can rotate the active root of trust (RoT) key stored on the HSM. Periodic rotation is a best security practice. As well, rotation also allows your organization to recover in case the RoT key is compromised. You can also rotate the [Master Key Encryption Key \(MKEK\)](#) which is below the RoT key in the key hierarchy.

One RoT key can be active at a time, but CipherTrust Manager retains older RoT keys, which can be made active again. The CLI and API allow you to list all RoT keys, view details for a given RoT key, rotate the active RoT key, and delete a given RoT key.

System [backups](#) that are tied to an HSM use the HSM RoT key to encrypt the backup file. This means that to decrypt and restore a backup that has been tied to an HSM, the CipherTrust Manager must have the encrypting RoT key set to active.

View Root of Trust Keys

Use `ksctl rot-keys list` to return a list of all RoT keys, including IDs, creation dates, and a boolean `is_active_key` indicating which key is the active key.

Use `ksctl rot-keys get --id <key_id>` to return this information about a particular key.

Rotate Active Root of Trust Key

Use `ksctl rot-keys rotate` to rotate the active key to a new or existing key. Pass in an `--id` flag for an existing RoT key to make that key active. Omit the `--id` flag to create a new RoT key with a randomly generated name. If a key with the specified id does not exist, CipherTrust Manager creates a new RoT key.

If you are creating a [cluster](#) sharing the same HSM partition, we recommend setting the RoT key to be the same on every node before joining nodes together. This way, CipherTrust Manager secrets and master keys can be protected by an HSM-secured RoT key during the join process, when these objects are transferred between nodes for the first time.

Note

For the k160 model, rotating the RoT key is not supported.

Delete a Root of Trust Key

Use `ksctl rot-keys delete --id <key_id>` to delete a RoT key. If there is only one RoT key, you must also supply a `--force` flag to force the operation. You can also use the `--force` flag to clean up any leftover RoT keys in the HSM which are not visible with `ksctl rot-keys list`.

Warning

Deleting the last root of trust key makes the CipherTrust Manager unusable, and any keys on the device **irrecoverable**. Deleting the last RoT key is only for the purpose of decommissioning a CipherTrust Manager and freeing storage space on the HSM.

Warning

Do not delete an RoT key if there are any system backups associated with the RoT key. If a **system backup** was taken with the `--tied-to-hsm` option, the backup file was encrypted with the active HSM RoT key. If you delete the RoT key which encrypted the backup file, the backup becomes permanently unusable, and cannot be decrypted and restored.

Upgrade Luna PCIe HSM Firmware for CipherTrust Manager k570 Appliances

You can update firmware for the Luna PCIe HSM embedded in CipherTrust Manager k570 models. The update steps vary for [Thales CipherTrust Manager k570](#) and [Trusted Cyber Technologies\(TCT\) CipherTrust Manager k570 appliances](#).

Update PCIe Firmware for Thales CipherTrust Manager k570

Updating the firmware ensures that the HSM has more up-to-date performance and security fixes.

For more detailed information on firmware upgrade, consult [Luna PCIe Documentation](#).

We have tested HSM firmware upgrade to the following versions:

- 7.8.4, which is the FIPS-validated firmware recommended by Thales.
- 7.9.0

Caution

Make sure to upgrade to CipherTrust Manager 2.17 or higher before upgrading the PCIe HSM. 7.8.4 and 7.9.0 firmware is incompatible with lower CipherTrust Manager versions. The root of trust configuration is lost and the CipherTrust Manager becomes unavailable if you upgrade the PCIe HSM with incompatible CipherTrust Manager software.

Tested Upgrade Paths

We have tested the following upgrade paths:

- 7.0.1 to 7.8.4
- 7.0.1 to 7.9.0
- 7.7.0 to 7.8.4
- 7.7.1-20 to 7.8.4
- 7.8.4 to 7.9.0

7.0.1 is the version installed on the PCIe HSM at the factory. 7.7.0 and 7.7.1-20 are previously recommended upgrade versions.

Caution

Due to an Out of Memory error, we no longer recommend upgrading any CipherTrust Manager device's PCIe HSM to the 7.7.0 HSM firmware version.

Note

If you do not know the current version of the embedded PCIe HSM, as the System Administrator (ksadmin) SSH in to the appliance (or connect via serial port using your password) and execute `"/usr/safenet/lunaclient/bin/lunacm"` utility. A Firmware Version is listed.

Update the Firmware

1. If your appliance is PED-authenticated, ensure you have [compatible PED firmware](#) before upgrade.

Caution

Luna PED firmware versions are backwards-compatible with older Luna HSM firmware, but a Luna HSM with firmware 7.7.0 or newer will refuse connection to a Luna PED with older firmware, with the error (LUNA_RET_PED_UNSUPPORTED_PROTOCOL error).

2. Obtain the 7.8.4 or 7.9.0 firmware update file (<filename>.fuf) and authentication code file (<filename>.txt). As `ksadmin`, transfer these files into the `/home/ksadmin/` directory of the appliance.
3. Take a **system backup** of all the CipherTrust Manager objects.
4. As the System Administrator (`ksadmin`) SSH in to the appliance (or connect via serial port using your password) and execute `"/usr/safenet/lunaclient/bin/lunacm"` utility.
5. Make sure an HSM admin slot is selected.

To see the available slots, enter:

```
lunacm:> slot list
```

Look for a slot with description "Admin Token Slot".

To select the active slot, enter:

```
lunacm:> slot set -slot <number>
```

6. Login as the HSM Security Officer (SO). For PED-authenticated devices, you are prompted to insert the blue PED key. For password-authenticated devices, you are prompted for the HSM SO password.

```
lunacm:> role login -name so
```

7. Update the HSM with the firmware upgrade file and authcode. You are prompted to confirm.

```
lunacm:>hsm updatefw -fuf /home/ksadmin/<fuf_filename> -authcode /home/ksadmin/<authcode_filename>
```

```
You are about to update the firmware.
The HSM will be reset.
Are you sure you wish to continue?
```

```
Type 'proceed' to continue, or 'quit' to quit now ->proceed

Updating firmware. This may take several minutes.

Firmware update passed. Resetting HSM.
```

The update is applied and the HSM is restarted.

8. If you plan to use a Luna Backup HSM to backup root of trust keys, make sure the Backup HSM has a version compatible with HSM firmware 7.7.0 or higher.

Update PCIe Firmware for TCT CipherTrust Manager k570

To update the firmware:

1. If your appliance is PED-authenticated, ensure you have compatible PED firmware.
2. Obtain the firmware update file (<filename>.fuf) and authentication code file (<filename>.txt). As ksadmin, transfer these files into the /home/ksadmin/ directory of the appliance.
3. Take a [system backup](#) of all the CipherTrust Manager objects.
4. As the System Administrator (ksadmin) SSH in to the appliance (or connect via serial port using your password) and execute "/usr/safenet/lunaclient/bin/lunacm" utility.
5. Make sure an HSM admin slot is selected.

To see the available slots, enter:

```
lunacm:> slot list
```

Look for a slot with description "Admin Token Slot".

To select the active slot, enter:

```
lunacm:> slot set -slot <number>
```

6. Login as the HSM Security Officer (SO). For PED-authenticated devices, you are prompted to insert the blue PED key. For password-authenticated devices, you are prompted for the HSM SO password.

```
lunacm:> hsm login
```

7. Update the HSM with the firmware upgrade file and authcode. You are prompted to confirm.

```
lunacm:>hsm updatefw -fuf /home/ksadmin/<firmware_update_file>.fuf -authcode
/home/ksadmin/<authcode_file>
```

```
You are about to update the firmware.
```

```
The HSM will be reset.
```

```
Are you sure you wish to continue?
```

```
Type 'proceed' to continue, or 'quit' to quit now ->proceed
```

```
Updating firmware. This may take several minutes.
```

```
Firmware update passed. Resetting HSM.
```

The update is applied and the HSM is restarted.

Note

Firmware upgrade can take a long time, depending on the number of objects stored on the HSM (a few minutes to several hours). Ensure that you can leave the update operation uninterrupted for this amount of time. Do not interrupt the procedure even if the operation appears to have stalled.

Clusters and Nodes

Note

This feature is not included in Community Edition and requires a valid Virtual CipherTrust Manager license. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

Overview

Clusters and nodes are the resources used to create and manage CipherTrust Manager clustering.

A cluster is a group of connected CipherTrust Manager appliances that share data. A cluster node is an individual appliance within the cluster. A cluster configuration is needed to start grouping CipherTrust Manager appliances together. [Application Administrators](#), such as `admin`, can perform cluster operations in the `root domain`.

The main purpose of clustering is to support High Availability. When clustered, all cluster nodes continuously synchronize their databases with each other. Any node in the cluster may be contacted for any operation. A cluster supports any combination of appliance models; both virtual and physical appliances can join the same cluster. However, all cluster nodes must be the same firmware version during normal operation, and this is enforced during the join operation. [Upgrading CipherTrust Manager firmware version for nodes in a cluster](#) requires a special sequence.

Nodes in a cluster communicate over port 5432 and 2380. Members of the cluster must have bi-directional access to each other on port 5432 and 2380.

Note

For proper cluster functionality, every node must be synchronized to the same time. Otherwise, communication between nodes fails. You must manually configure the same [NTP server](#) for each node. The NTP server configuration is not replicated, so you must configure each node independently.

Cluster Disconnection

If a cluster node is disconnected or powers down, an alarm is set and syslog messages are sent.

When this happens, you should first [monitor the node](#) and perform basic troubleshooting to determine if the disconnection is temporary, and the node will eventually resynchronize to the cluster.

If these actions don't resolve the connection, you should consider the node [permanently disconnected](#) and remove the node from the cluster.

Initial Monitoring and Troubleshooting

First determine if the disconnection is temporary and recoverable. When the node comes back it synchronizes with other members of the cluster. It can take a few minutes to a few hours for data to fully synchronize.

To monitor cluster node status during this time, there are multiple options.

- As an Application Administrator such as `admin` [view the cluster summary](#) using `ksctl`.
- As an Application Administrator [view the status of all nodes](#) using `ksctl`.
- As an Application Administrator [view cluster error information](#) using `ksctl`.
- As an Application Administrator view the status of an individual cluster node using `ksctl` or in the [GUI](#).
- As an Application Administrator, [view the Raft Status](#) of the cluster using `ksctl`.
- As an Application Administrator, view the [Raft connection status of the cluster](#) using `ksctl`.
- If the above statuses display errors, obtain more detailed information to troubleshoot. As `ksadmin` in an SSH session, use `kscfg cluster-report` to generate a detailed information on each node. Move the resulting `tar.bz2` file to a different machine and unpack it. View the following data files in `cm-report-
<timestamp>/postgres/spock/views`:
 - `node_summary.data`
 - `node_slots.data`
 - `node_replication_rates.data`

If a node lists its state as "down" persistently, and the `dataLag` value in the node status results keeps increasing over time, those are indications that the connection will not repair itself, and you should consider the node **permanently disconnected**.

Recover from Permanent Disconnection

In this scenario, we recommend that you remove the disconnected node from the cluster. This prevents the remaining nodes from continually storing catch-up changes for the missing node, which would eventually fill up their local volumes.

1. Issue the `ksctl cluster nodes delete` command to one remaining node.

That node then sends the same command to the rest of the cluster.

2. Reset the disconnected node.

```
kscfg system reset -y
```

Warning

Refer to section [Services Reset](#) for important information on using this command.

3. Join the disconnected node back to the cluster.

Cluster Size and Performance

The maximum number of cluster nodes is 20. Joining a 21st node will fail.

For optimal performance and stability, the recommended cluster size is 4-6 nodes. If you desire a larger cluster size, be sure to plan carefully for these scalability considerations:

- Good connectivity is required between every node. As each node is continually connected to every other node, network issues can cause synchronization problems between nodes, creating maintenance and troubleshooting overhead.
- Upgrading firmware for a larger cluster can be complex and time-consuming.
 - Each node must be upgraded individually.
 - For online in-place upgrade, running different CipherTrust Manager firmware versions within a cluster is only supported for one version back, and only during the upgrade window.

What is not clustered?

Most things replicate across the cluster. However, the following items are specific to an individual node, and are not clustered:

- Backup files
- Backup keys
- Debug, KMIP Activity, and NAE Activity logs
- NTP configuration
- HSM configuration
- Instance name
- Virtual CipherTrust Manager license

Note

Licenses for connector applications are replicated. This is because the 'Connector Lock Code', unlike the 'Key Manager Lock Code', is cluster-wide.

- Interface certificates
- Proxy settings

Sharing HSM Root of Trust Configuration

CipherTrust Manager internally uses a flexible security architecture to allow clustering instances with both shared and distinct 'root of trust' configurations.

If the member node and the new node share the same HSM partition, it is recommended to pass the flag `--shared-hsm-partition` during nodes **create** or **fulljoin**, and to **set each node to use the same Root of Trust (RoT) key** using `ksctl rot-keys rotate --id <root-of-trust-id>`. This configuration further increases the security of cluster join procedure by ensuring that CipherTrust Manager secrets and master keys are protected by an HSM-secured RoT key during the join process, when these objects are transferred between nodes for the first time.

When the new node uses a distinct HSM partition or does not use any HSM, it is still possible to join it to an existing cluster by not specifying the `--shared-hsm-partition` flag.

Caution

Each CipherTrust Manager node in the cluster uses its own root of trust configuration to protect the Key Encryption Key (KEK) chain that secures the sensitive data of the cluster. When a new node, not connected to an HSM, joins a CipherTrust Manager cluster with all HSM connected nodes, it becomes the weakest point in the key hierarchy and potentially weakens the overall security of the cluster.

Managing Clusters and Nodes through UI

Cluster management operations are available on the **Admin Settings>Cluster** page of the UI in the root domain, to Application Administrators such as `admin`.

Creating a New Cluster Configuration

A cluster configuration is necessary before you can join nodes to the cluster.

1. Login to the root domain as an Application Administrator, such as `admin`. The CipherTrust Manager you log into becomes the first node of the new cluster.
2. Navigate to **Admin Settings > Cluster**.
3. Click **Manage Cluster** and select **Add cluster** from the dropdown options.
4. Review the **Introduction** describing the effects of creating a cluster configuration.
5. Click **Next**.
6. Provide the **Privately accessible IP Address or Hostname**. This is the IP address or hostname used for data replication between cluster nodes.
7. Review the **Public address connectors will use to access this node** value and change it if necessary. In most cases, this is the same value as the **Privately accessible IP Address or Hostname**. This is the Fully Qualified Domain Name or public IP address of the CipherTrust Manager you are logged into. This is the value which external clients use to communicate with the CipherTrust Manager.
8. Click **Add Cluster**.

The current CipherTrust Manager is listed as a **Host** in the table, with a **Status** of ready.

Joining a Node to a Cluster

Joining involves operations on two CipherTrust Manager instances.

- A joining node - This is a node that you wish to add into the cluster. Some services restart on the joining node during the join, making the node inaccessible. As well, its data is reset and synchronized with the cluster.
- A member node - This is a node that is already in the cluster, and which can exchange certificates with the joining node.

You can only join one node to the cluster at a time. Wait for the join operation to complete before starting another join operation.

Starting in version 2.12, CipherTrust Manager enforces that the joining node must be the same firmware version as the nodes already in the cluster. For example, if you attempt to join a node running 2.11 firmware to a cluster of nodes with 2.12 firmware, the join operation fails.

Caution

Restoring a backup to the node immediately before the join is unnecessary and strongly discouraged, as joining a cluster resets and synchronizes the node's data. If you have recently restored a backup on the CipherTrust Manager, make sure the restore operation is complete, all services are up and running, and that at least 10 minutes have passed. Then, you can join the CipherTrust Manager to a cluster. If you join a cluster before or soon after a restore is complete, this can result in other nodes in the cluster failing to reboot.

To join a node to a cluster

1. Login to the root domain as an Application Administrator, such as `admin` to a member node.
2. Navigate to **Admin Settings > Cluster**.
3. Click **Manage Cluster** and select **Add node** from the dropdown options.
4. Provide connection information for the joining node.
 - **Public URL of the new node** - The general public hostname, IP address, or FQDN of the joining node.
 - **Connector Public address of the new node** - The hostname, IP address, or FQDN of the joining node accessible by connector clients.
5. If you are using a separate network for cluster nodes to communicate with each other:
 - a. Uncheck the **Cluster address is the same as the Public URL hostname** option
 - b. Provide the **Cluster network address of the new node**.

Note

The IP address or FQDN you provide here is used by the nodes to communicate with each other. For example, in AWS deployments, if the nodes are on the same VPC, the internal IP address of the member and joining node should be used, not a public IP.

Additionally, to successfully join a cluster, a CipherTrust Manager's hostname can consist only of lowercase letters, numbers, and hyphens. The default hostname of `ciphertrust` meets these criteria. The `ksadmin` user can view and change the hostname with the `kscfg` utility.

To successfully join a cluster using an FQDN, uppercase letters are not allowed.

6. If you wish to use a shared HSM [root of trust configuration](#), enable the **Shared HSM Partition** checkbox.
7. Click **Add node**.

The current node now attempts to connect to the joining node. 60 connection attempts are allowed. Another tab opens in the browser for the joining node's **Cluster** page.

8. Switch to the other window or tab to the joining node to accept the request. Login as an Application Administrator, such as `admin`.

A popup **Join Node to Cluster** is displayed.

9. Review the displayed information.

- **The origin of the request** indicates the IP address or hostname of the cluster node making the request.
- The IP address/hostname and port for cluster operations on the joining node are displayed.

10. If the information is correct, click **Join** to confirm.

Screens on both CipherTrust Manager demonstrate the joining node passing a certificate signing request to the member node to sign, the member node passing the signed certificate back, and then the join operation succeeding.

The joining node restarts, and the cluster replicates data to the node.

When the operation is complete, the joining node is listed as a **Host** in the table, with a **Status** of "Ready", indicating that the node is synchronized with the cluster and ready to receive and transmit cluster data.

Caution

The join operation might take some time to complete, depending on network speed and database size. Completion times of 30 minutes are not unusual.

While a join is underway, do not restart the system, as this can cause the join to fail, and the node to disappear from the cluster. There is no need to manually restart the system after join is completed as well.

As well, make sure that the join operation is complete, with the node displaying a **Status** of "Ready" before attempting to join another node to the cluster.

You can always [view the node status](#) to check progress while a join is underway. If there is a failure during joining, a persistent down status is displayed. The down status can appear briefly during a normal join.

Removing a Node from a Cluster

Warning

When the cluster is in an unhealthy state, it must be restored to a healthy state before any nodes are deleted.

To remove a node from a cluster, first remove that node from the cluster then delete the cluster configuration on the removed node. Deleting the cluster configuration allows a node to rejoin the original cluster, join another cluster, or create a new cluster.

1. Login to the root domain as an Application Administrator, such as `admin`, to a node in the cluster which will remain in the cluster.

Note

You cannot remove the node you are logged into from the cluster.

2. Navigate to **Admin Settings > Cluster**.
3. In the table, find the node you want to remove. Click the overflow icon () corresponding to the desired node and click **Remove Node**.

A confirmation pop-up displays.

4. Click **Remove** to confirm.
5. Login to the root domain as an Application Administrator, such as `admin`, to the newly removed node.
6. Navigate to **Admin Settings > Cluster**.
7. Click **Manage Cluster** and select **Delete cluster configuration** from the dropdown options.

A confirmation pop-up displays.

8. Click **Delete** to confirm. This operation restarts the CipherTrust Manager and can take up to 5 minutes.

Removing a node from a cluster using force flag

If node removal fails for any reason, you can use the `--force` flag to remove the node.

```
ksctl cluster nodes delete --id=ebab0738-6e09-4b0d-8c99-850d7f24dfac --force
```

Note

- By default, node deletion is replicated across the cluster.
- With the `--force` flag set to `true`, the given node is removed only from the local node. You must call `delete` with `--force` set to `true` on remaining cluster nodes to remove them.

- Use the `--force` flag only when a majority of nodes are unavailable at the time of removing a node from the cluster.

Re-joining a Node to a Cluster

Before rejoining it is recommended to reset the node.

To reset the node enter the command in an SSH session as the `ksadmin` user:

```
$ kscfg system reset -y
```

Warning

Refer to section [Services Reset](#) for important information on using this command.

Join the node as described in [Joining a Node to a Cluster](#).

Deleting a Cluster

1. [Remove](#) all but one node from the cluster, taking care to delete the cluster configuration on each removed node.
2. Login to the root domain as an Application Administrator, such as `admin`, to the last remaining node in the cluster.
3. Navigate to **Admin Settings > Cluster**.
4. Click **Manage Cluster** and select **Delete cluster configuration** from the dropdown options.

A confirmation pop-up displays.

5. Click **Delete** to confirm. This operation can take up to 5 minutes.

Viewing Cluster Node Status

You can view information on the general health of cluster nodes through the web console UI. More detailed replication health information is available through the `"ksctl cluster summary"` CLI command.

1. Login to the root domain as an Application Administrator, such as `admin`, to the last remaining node in the cluster.
2. Navigate to **Admin Settings > Cluster**.

3. View the cluster nodes listed in the table. Each **Host** lists a **Status**.

The possible statuses for a node in the table are:

- **Ready**: indicates that the node is synchronized with the cluster and able to receive and transmit data.
- **Both Ports are Down**: indicates that both the cluster replication port and the raft consensus port are unavailable.
- **Replication is Down**: indicates that the cluster port is unavailable.
- **Consensus Port is Down**: indicates that the raft consensus port is unavailable.

4. View the diagram of individual nodes and below the table, with hexagons representing individual nodes.

The possible statuses for a node displayed in the hexagon diagram are:

- **Not Clustered**
- **Ready**
- **Joining**
- **Down**

Viewing Overall Cluster Status

1. Login to the root domain as an Application Administrator, such as `admin`, to the last remaining node in the cluster.
2. Navigate to **Admin Settings > Cluster**.
3. View the cluster statuses directly below the **Cluster** page title.

From left to right, the statuses show the total number of nodes, the number of nodes which are down, the number of nodes which are ready, and the cluster consensus status.

Cluster consensus operates by electing a single leader among a group of nodes in a cluster. This leader is responsible for managing the system's state and coordinating all updates. A status of **Ready** indicates that the cluster consensus is functioning properly, a leader has been successfully selected for the entire cluster, and a majority of cluster ports are available.

Note

In a cluster, the Cluster Consensus Status may be shown as up even when more than 50% of the nodes' consensus ports are down. This indicates that the Cluster Consensus Status and the Consensus port up/down are not directly correlated. Therefore, if more than half of the nodes' consensus ports are down, it

should not be automatically assumed that the Cluster Consensus Status will also be down. The cluster is fault-tolerant, and the consensus mechanism can continue to function as long as there are active connections from other nodes.

Managing Clusters and Nodes through ksctl CLI

Use the **ksctl** command line interface for managing clusters and nodes. The relevant commands are:

```
ksctl cluster csr
ksctl cluster delete
ksctl cluster error
ksctl cluster fulljoin
ksctl cluster info
ksctl cluster join
ksctl cluster new
ksctl cluster nodes list
ksctl cluster nodes get
ksctl cluster nodes create
ksctl cluster nodes delete
```

Note

All global flags can be configured using a configuration file. Otherwise, you will need to specify the URL, username and password for each call as demonstrated in a few examples below.

Check the Status of a Cluster

To check the status of a cluster enter the command:

```
ksctl cluster info
```

This returns the following response:

```
{
  "nodeID": "371adaf",
  "status": {
    "code": "r",
    "description": "ready"
  },
  "nodeCount": 1,
  "raftStatus": "up",
  "dataReplicationStatus": {
    "code": "r",
    "description": "ready"
  }
}
```

```
}
```

`status` indicates overall cluster health for the node. A `status` of `ready` indicates that the node is synchronized with the cluster, and able to receive and transmit data to other cluster nodes.

`raftStatus` indicates Raft Consensus. Raft operates by electing a single leader among a group of nodes (servers) in a cluster. This leader is responsible for managing the system's state and coordinating all updates. In the above response, the `raftStatus` is `up`, which indicates that the Raft Consensus is functioning properly, and a leader has been successfully selected for the entire cluster.

Create a New Cluster Configuration

Make the following call on a running appliance (you will need to insert the hostname/IP of the appliance). When running `cluster new`, `--host` and `--url` are most likely going to be identical except that `--host` will not have the protocol. To create a new cluster enter the command:

```
ksctl cluster new --host=localhostName --url=urlOfCurrentNode --user=username --password=Password_1
```

This returns the following response:

```
{
  "nodeID": "371adaf",
  "status": {
    "code": "r",
    "description": "ready"
  },
  "nodeCount": 1,
  "raftStatus": "up",
  "dataReplicationStatus": {
    "code": "r",
    "description": "ready"
  }
}
```

Show the Status of All Nodes in the Cluster

This command shows the status of the nodes in the cluster from the current node's point of view. This means that you do not see detailed replication health information about the cluster node the command is run on. To return full information for every node, including the current node, use the [cluster summary](#) command.

Enter the following command on any node:

```
ksctl cluster nodes list
```

This returns the following response:

```
{
  "skip": 0,
  "limit": 256,
  "total": 3,
```



```

"resources": [
  {
    "nodeID": "5337792",
    "status": {
      "code": "r",
      "description": "ready"
    },
    "host": "13.220.57.46",
    "port": 5432,
    "raftPort": 2380,
    "state": null,
    "isThisNode": true,
    "publicAddress": "13.220.57.46",
    "dataReplicationStatus": {
      "code": "r",
      "description": "ready"
    }
  },
  {
    "nodeID": "8313821",
    "status": {
      "code": "d",
      "description": "down"
    },
    "host": "54.242.79.18",
    "port": 5432,
    "raftPort": 2380,
    "state": "streaming",
    "isThisNode": false,
    "sentLSN": "0/59F86E0",
    "writeLSN": "0/59F86E0",
    "flushLSN": "0/59F86E0",
    "replayLSN": "0/59F86E0",
    "writeLag": "00:00:00",
    "flushLag": "00:00:01.002402",
    "replayLag": "00:00:00",
    "sentLagBytes": 0,
    "writeLagBytes": 0,
    "flushLagBytes": 0,
    "replayLagBytes": 0,
    "sentLagSize": "0 bytes",
    "writeLagSize": "0 bytes",
    "flushLagSize": "0 bytes",
    "replayLagSize": "0 bytes",
    "connected": true,
    "replicationBlocked": false,
    "connectTime": "2025-07-22T18:15:41.541186Z",
    "uptime": "1 day 14:50:52.315629",
    "dataReplicationStatus": {
      "code": "r",
      "description": "ready"
    },
    "raftConnectionStatus": "disconnected"
  },
  {
    "nodeID": "ec0163f",
    "status": {
      "code": "r",
      "description": "ready"
    }
  }
]

```

```

    },
    "host": "3.92.69.63",
    "port": 5432,
    "raftPort": 2380,
    "state": "streaming",
    "isThisNode": false,
    "sentLSN": "0/59F86E0",
    "writeLSN": "0/59F86E0",
    "flushLSN": "0/59F86E0",
    "replayLSN": "0/59F86E0",
    "writeLag": "00:00:00",
    "flushLag": "00:00:01.002576",
    "replayLag": "00:00:00",
    "sentLagBytes": 0,
    "writeLagBytes": 0,
    "flushLagBytes": 0,
    "replayLagBytes": 0,
    "sentLagSize": "0 bytes",
    "writeLagSize": "0 bytes",
    "flushLagSize": "0 bytes",
    "replayLagSize": "0 bytes",
    "connected": true,
    "replicationBlocked": false,
    "connectTime": "2025-07-22T18:13:44.980966Z",
    "uptime": "1 day 14:52:48.875849",
    "dataReplicationStatus": {
      "code": "r",
      "description": "ready"
    },
    "raftConnectionStatus": "connected"
  }
]
}

```

In the above response, The `raftConnectionStatus` shows whether current node is connected to the Raft service of the other node.

`status` indicates overall cluster health for indicated node. A `status` of `ready` indicates that the node is synchronized with the cluster, and able to receive and transmit data to other cluster nodes.

The possible statuses are:

- `not clustered`
- `ready`
- `joining`
- `down`

Show the Status of a Single Node in the Cluster

To show the status of a single node in a cluster enter the command:

```
ksctl cluster nodes get --id 371adaf
```

This returns the following response:

```
{
  "nodeID": "371adaf",
  "status": {
    "code": "r",
    "description": "ready"
  },
  "host": "18.234.52.146",
  "port": 5432,
  "raftPort": 2380,
  "state": null,
  "isThisNode": true,
  "publicAddress": "18.234.52.146",
  "dataReplicationStatus": {
    "code": "r",
    "description": "ready"
  }
}
```

`status` indicates overall cluster health for the node. A `status` of `ready` indicates that the node is synchronized with the cluster, and able to receive and transmit data to other cluster nodes.

Show a summary of all nodes in the cluster

To show a summary of all nodes in a cluster, run:

```
ksctl cluster summary
```

This returns the following response:

```
{
  "skip": 0,
  "limit": 0,
  "total": 3,
  "resources": {
    "03b578c": {
      "clusterSummary": {
        "03b578c": {
          "summary": "GOOD",
          "nodeInfo": {
            "host": "18.205.235.25",
            "port": 5432,
            "nodeID": "03b578c",
            "status": {
              "code": "r",
              "description": "ready"
            },
            "nodeType": "rw",
            "raftPort": 2380,
            "isThisNode": true,
            "dataReplicationStatus": {
              "code": "r",
              "description": "ready"
            }
          }
        }
      }
    }
  }
}
```

```

    },
    "clusterErrors": null
  },
  "371adaf": {
    "summary": "GOOD",
    "nodeInfo": {
      "host": "18.234.52.146",
      "port": 5432,
      "state": "streaming",
      "nodeID": "371adaf",
      "status": {
        "code": "r",
        "description": "ready"
      },
    },
    "uptime": "00:19:17.191817",
    "sentLSN": "0/52F5660",
    "flushLSN": "0/52F5660",
    "flushLag": "00:00:00",
    "nodeType": "rw",
    "raftPort": 2380,
    "writeLSN": "0/52F5660",
    "writeLag": "00:00:00",
    "connected": true,
    "replayLSN": "0/52F5660",
    "replayLag": "00:00:00",
    "isThisNode": false,
    "connectTime": "2025-07-21T09:21:16.286806Z",
    "sentLagSize": "0 bytes",
    "flushLagSize": "0 bytes",
    "sentLagBytes": 0,
    "writeLagSize": "0 bytes",
    "flushLagBytes": 0,
    "publicAddress": "18.234.52.146",
    "replayLagSize": "0 bytes",
    "writeLagBytes": 0,
    "replayLagBytes": 0,
    "replicationBlocked": false,
    "raftConnectionStatus": "connected",
    "dataReplicationStatus": {
      "code": "r",
      "description": "ready"
    }
  },
  "clusterErrors": null
},
"b9e46cc": {
  "summary": "BAD: node is inactive",
  "nodeInfo": {
    "host": "3.80.116.119",
    "port": 5432,
    "state": "streaming",
    "nodeID": "b9e46cc",
    "status": {
      "code": "d",
      "description": "down"
    },
  },
  "uptime": "00:04:46.472322",
  "sentLSN": "0/52F5660",
  "flushLSN": "0/52F5660",

```

```

        "flushLag": "00:00:00",
        "nodeType": "rw",
        "raftPort": 2380,
        "writeLSN": "0/52F5660",
        "writeLag": "00:00:00",
        "connected": true,
        "replayLSN": "0/52F5660",
        "replayLag": "00:00:00",
        "isThisNode": false,
        "connectTime": "2025-07-21T09:35:47.006301Z",
        "sentLagSize": "0 bytes",
        "flushLagSize": "0 bytes",
        "sentLagBytes": 0,
        "writeLagSize": "0 bytes",
        "flushLagBytes": 0,
        "replayLagSize": "0 bytes",
        "writeLagBytes": 0,
        "replayLagBytes": 0,
        "replicationBlocked": false,
        "raftConnectionStatus": "disconnected",
        "dataReplicationStatus": {
            "code": "r",
            "description": "ready"
        }
    },
    "clusterErrors": [
        {
            "errorTime": "2025-07-21T09:39:57Z",
            "errorMessage": "RAFT: dial tcp 3.80.116.119:2380: co
connect: connection timed out"
        }
    ]
},
"lastUpdated": "2025-07-21T09:40:33.597766Z",
"summary": "bad: node - b9e46cc down",
"clusterInfo": {
    "nodeID": "03b578c",
    "status": {
        "code": "r",
        "description": "ready"
    },
    "nodeCount": 3,
    "raftStatus": "up",
    "dataReplicationStatus": {
        "code": "r",
        "description": "ready"
    }
},
"371adaf": {
    "clusterSummary": {
        "03b578c": {
            "summary": "GOOD",
            "nodeInfo": {
                "host": "18.205.235.25",
                "port": 5432,
                "state": "streaming",
                "nodeID": "03b578c",

```

```

        "status": {
            "code": "r",
            "description": "ready"
        },
        "uptime": "00:19:20.523477",
        "sentLSN": "0/39CFB20",
        "flushLSN": "0/39CFB20",
        "flushLag": "00:00:00",
        "nodeType": "rw",
        "raftPort": 2380,
        "writeLSN": "0/39CFB20",
        "writeLag": "00:00:00",
        "connected": true,
        "replayLSN": "0/39CFB20",
        "replayLag": "00:00:00",
        "isThisNode": false,
        "connectTime": "2025-07-21T09:21:15.967216Z",
        "sentLagSize": "0 bytes",
        "flushLagSize": "0 bytes",
        "sentLagBytes": 0,
        "writeLagSize": "0 bytes",
        "flushLagBytes": 0,
        "replayLagSize": "0 bytes",
        "writeLagBytes": 0,
        "replayLagBytes": 0,
        "replicationBlocked": false,
        "raftConnectionStatus": "connected",
        "dataReplicationStatus": {
            "code": "r",
            "description": "ready"
        }
    },
    "clusterErrors": null
},
"371adaf": {
    "summary": "GOOD",
    "nodeInfo": {
        "host": "18.234.52.146",
        "port": 5432,
        "nodeID": "371adaf",
        "status": {
            "code": "r",
            "description": "ready"
        },
        "nodeType": "rw",
        "raftPort": 2380,
        "isThisNode": true,
        "publicAddress": "18.234.52.146",
        "dataReplicationStatus": {
            "code": "r",
            "description": "ready"
        }
    },
    "clusterErrors": null
},
"b9e46cc": {
    "summary": "BAD: node is inactive",
    "nodeInfo": {
        "host": "3.80.116.119",

```

```

        "port": 5432,
        "state": "streaming",
        "nodeID": "b9e46cc",
        "status": {
            "code": "d",
            "description": "down"
        },
        "uptime": "00:04:41.176151",
        "sentLSN": "0/39CFB20",
        "flushLSN": "0/39CFB20",
        "flushLag": "00:00:00",
        "nodeType": "rw",
        "raftPort": 2380,
        "writeLSN": "0/39CFB20",
        "writeLag": "00:00:00",
        "connected": true,
        "replayLSN": "0/39CFB20",
        "replayLag": "00:00:00",
        "isThisNode": false,
        "connectTime": "2025-07-21T09:35:55.314542Z",
        "sentLagSize": "0 bytes",
        "flushLagSize": "0 bytes",
        "sentLagBytes": 0,
        "writeLagSize": "0 bytes",
        "flushLagBytes": 0,
        "replayLagSize": "0 bytes",
        "writeLagBytes": 0,
        "replayLagBytes": 0,
        "replicationBlocked": false,
        "raftConnectionStatus": "disconnected",
        "dataReplicationStatus": {
            "code": "r",
            "description": "ready"
        }
    },
    "clusterErrors": [
        {
            "errorTime": "2025-07-21T09:40:05Z",
            "errorMessage": "RAFT: dial tcp 3.80.116.119:2380: co
nnect: connection timed out"
        }
    ]
},
"lastUpdated": "2025-07-21T09:40:36.602899Z",
"summary": "bad: node - b9e46cc down",
"clusterInfo": {
    "nodeID": "371adaf",
    "status": {
        "code": "r",
        "description": "ready"
    },
    "nodeCount": 3,
    "raftStatus": "up",
    "dataReplicationStatus": {
        "code": "r",
        "description": "ready"
    }
}
}

```

```

},
"b9e46cc": {
  "clusterSummary": {
    "03b578c": {
      "summary": "GOOD",
      "nodeInfo": {
        "host": "18.205.235.25",
        "port": 5432,
        "state": "streaming",
        "nodeID": "03b578c",
        "status": {
          "code": "r",
          "description": "ready"
        },
        "uptime": "00:04:38.184095",
        "sentLSN": "0/529BB88",
        "flushLSN": "0/529BB88",
        "flushLag": "00:00:00",
        "nodeType": "rw",
        "raftPort": 2380,
        "writeLSN": "0/529BB88",
        "writeLag": "00:00:00",
        "connected": true,
        "replayLSN": "0/529BB88",
        "replayLag": "00:00:00",
        "isThisNode": false,
        "connectTime": "2025-07-21T09:35:56.47439Z",
        "sentLagSize": "1144 bytes",
        "flushLagSize": "1144 bytes",
        "sentLagBytes": 1144,
        "writeLagSize": "1144 bytes",
        "flushLagBytes": 1144,
        "replayLagSize": "1144 bytes",
        "writeLagBytes": 1144,
        "replayLagBytes": 1144,
        "replicationBlocked": false,
        "raftConnectionStatus": "connected",
        "dataReplicationStatus": {
          "code": "r",
          "description": "ready"
        }
      },
      "clusterErrors": null
    },
    "371adaf": {
      "summary": "GOOD",
      "nodeInfo": {
        "host": "18.234.52.146",
        "port": 5432,
        "state": "streaming",
        "nodeID": "371adaf",
        "status": {
          "code": "r",
          "description": "ready"
        },
        "uptime": "00:04:38.267751",
        "sentLSN": "0/529BB88",
        "flushLSN": "0/529BB88",
        "flushLag": "00:00:00",

```



```

        "nodeType": "rw",
        "raftPort": 2380,
        "writeLSN": "0/529BB88",
        "writeLag": "00:00:00",
        "connected": true,
        "replayLSN": "0/529BB88",
        "replayLag": "00:00:00",
        "isThisNode": false,
        "connectTime": "2025-07-21T09:35:56.390734Z",
        "sentLagSize": "1144 bytes",
        "flushLagSize": "1144 bytes",
        "sentLagBytes": 1144,
        "writeLagSize": "1144 bytes",
        "flushLagBytes": 1144,
        "publicAddress": "18.234.52.146",
        "replayLagSize": "1144 bytes",
        "writeLagBytes": 1144,
        "replayLagBytes": 1144,
        "replicationBlocked": false,
        "raftConnectionStatus": "connected",
        "dataReplicationStatus": {
            "code": "r",
            "description": "ready"
        }
    },
    "clusterErrors": null
},
"b9e46cc": {
    "summary": "GOOD",
    "nodeInfo": {
        "host": "3.80.116.119",
        "port": 5432,
        "nodeID": "b9e46cc",
        "status": {
            "code": "r",
            "description": "ready"
        }
    },
    "nodeType": "rw",
    "raftPort": 2380,
    "isThisNode": true,
    "dataReplicationStatus": {
        "code": "r",
        "description": "ready"
    }
},
"clusterErrors": [
    {
        "errorTime": "2025-07-21T09:39:06Z",
        "errorMessage": "RAFT: failed to get leased session f
or etcd client: context deadline exceeded"
    },
    {
        "errorTime": "2025-07-21T09:39:11Z",
        "errorMessage": "RAFT: context deadline exceeded"
    }
]
},
"lastUpdated": "2025-07-21T09:40:34.762306Z",

```

```

    "summary": "bad: raft down",
    "clusterInfo": {
      "nodeID": "b9e46cc",
      "status": {
        "code": "d",
        "description": "down"
      },
      "nodeCount": 3,
      "raftStatus": "down",
      "dataReplicationStatus": {
        "code": "r",
        "description": "ready"
      }
    }
  }
}

```

The response displays:

- summary of each node
- cluster error information, if any
- human readable consolidated summary
- last updated information

Show all cluster errors

To show a summary of cluster error information, if any, run:

```
ksctl cluster errors
```

Join a Node to a Cluster

You can join a node to a cluster in [one command](#) or [three commands](#).

During cluster join, the node's data is reset and synchronized with the cluster. This means that restoring a backup to the node is unnecessary.

You can only join one node to the cluster at a time. Wait for the join operation to complete before starting another join operation.

Starting in version 2.12, CipherTrust Manager enforces that the joining node must be the same firmware version as the nodes already in the cluster. For example, if you attempt to join a node running 2.11 firmware to a cluster of nodes with 2.12 firmware, the join operation fails.

Note

The `ksctl` commands to join a node require an IP address, hostname, or FQDN of both the cluster member and of the joining node. These IP addresses and hostnames are used by the nodes to talk to each other, and not by the CLI to talk to the nodes. For example, in AWS deployments, if the nodes are on the same VPC, the internal IP address of the member and joining node should be used, not a public IP.

Additionally, to successfully join a cluster, a CipherTrust Manager's hostname can consist only of lowercase letters, numbers, and hyphens. The default hostname of `ciphertrust` meets these criteria. The `ksadmin` user can view and change the hostname with the `kscfg` utility.

To successfully join a cluster using an FQDN, uppercase letters are not allowed.

Caution

If `allowlist` is enabled on any node in the cluster, before adding the new node to the cluster, ensure that the new node's IP address is added to the allowlist of nodes where `allowlist` is active. Refer to [cluster allowlist](#) for details.

One Step Cluster Join Using `fulljoin`

It is simplest to use the **`fulljoin`** command to join a node. For the cluster **`fulljoin`** command, you are required to input a member's IP address, hostname, or FQDN, the joining node's IP address, hostname, or FQDN, and either the joining node's configuration file, or its username, password, and URL.

To complete the process using the **`fulljoin`** command with a configuration file enter the command:

```
ksctl cluster fulljoin --member=<member_IP_or_hostname_or_FQDN> --newnodehost=<joining_IP_or_hostname_or_FQDN> --newnodeconfig=<config_File_of_Joining_Node>.
```

As with the individual commands, you must wait until the join operation has completed and the node status is ready before joining another node.

Response:

```
When you add a node to a cluster, the existing data of the node is deleted. Are you sure want to join? [y/N]
```

```
Attempting to get the CSR from the joining node...
Finished getting the CSR from the joining node...
Attempting to get the Certificate and CA Chain from the member node...
Finished getting the Certificate and CA Chain from the member node...
Attempting to join the new node to the member's cluster...
```

```
{
  "nodeID": "",
  "status": {
    "code": "creating",
```

```

    "description": "joining: creating (1/5)"
  }
}

```

When a node joins a cluster, the node adopts the credentials of the cluster. Would you like to write the new cluster credentials

to the provided configuration file? [y/N]

y

An example of the **cluster fulljoin** command without using a configuration file:

```

ksctl cluster fulljoin --member=<member_IP_or_hostname_or_FQDN> --newnodehost=<joining_IP_or_hostname_or_FQDN> --newnodepass=<joining_node_password> --newnodeuser=<joining_node_user> --newnodeurl=<joining_node_URL>

```

Response:

When you add a node to a cluster, the existing data of the node is deleted. Are you sure want to join? [y/N]

y

Attempting to get the CSR from the joining node...

Finished getting the CSR from the joining node...

Attempting to get the Certificate and CA Chain from the member node...

Finished getting the Certificate and CA Chain from the member node...

Attempting to join the new node to the member's cluster...

```

{
  "nodeID": "",
  "status": {
    "code": "creating",
    "description": "joining: creating (1/5)"
  }
}

```

Three Step Cluster Join

1. Get the csr from the joining node by entering the command **ksctl cluster csr**
2. Get the certificate and CA chain from a member node by entering the command **ksctl cluster nodes create**
3. Run the join command from the joining node by entering the command **ksctl cluster join**

Caution

If you have recently restored a backup on the CipherTrust Manager, make sure the restore operation is

complete and all services are up and running. Then, you can join the CipherTrust Manager to a cluster. If you join a cluster before restore is complete, this can result in other nodes in the cluster failing to reboot.

Caution

The join operation might take some time to complete, depending on network speed and database size. Completion times of 30 minutes are not unusual.

While a join is underway, do not restart the system, as this can cause the join to fail, and the node to disappear from the cluster. There is no need to manually restart the system after join is completed as well.

You can always [view the node status](#) to check progress while a join is underway. If there is a failure during joining, a persistent `down` status is displayed. The `down` status can appear briefly during a normal join.

4. Check that the join operation has completed and that the node status is `ready` before joining another node.

Remove a Node from a Cluster

To remove a node from a cluster, first remove that node from the cluster then delete the cluster configuration on the removed node. Deleting the cluster configuration allows a node to rejoin the original cluster, join another cluster, or create a new cluster.

1. Remove the node from the cluster. A node cannot remove itself, so you must call this on some other node in the cluster:

```
ksctl cluster nodes delete --id=acblefg
```

2. Delete the cluster configuration on the **removed node**:

```
ksctl cluster delete
```

Note

If `ksctl cluster delete` doesn't work for any reason it is always possible to perform a full system

reset to ensure any left over data is removed from the node. A node can be reset using the `ksctl services reset` command.

Warning

Refer to section [Services Reset](#) for important information on using this command.

Rejoin a Node to a Cluster

Before rejoining it is recommended to reset the node.

To reset the node enter the command:

```
$ ksctl services reset
```

Warning

Refer to section [Services Reset](#) for important information on using this command.

Join the node as described in [join a node to a cluster](#).

Create a New Cluster from a Removed Node

Ensure the node has been successfully removed from the cluster and that `ksctl cluster delete` has been performed, as described in [remove a node from a cluster](#).

Create a new cluster as described in [create a new cluster](#).

Cluster Allowlist

The CipherTrust Manager introduces a feature to configure allowlist for the cluster. The allowlist is a list of IPs that are allowed to connect to the cluster port.

To enable this feature, you need to add at least one IP to the allowlist. By default, this feature is disabled.

Note

- The allowlist is configured on a per-node basis. Prior to adding a new node to the cluster, the new node's

IP address must be included in the allowlist of every node in the cluster where the allowlist is active.

- Only IPs (IPv4, IPv6) are allowed in the allowlist.

Caution

When a **public IP** is used in the "--host" field while creating a cluster, that IP must be added to the allowlist of the current node.

Configure Cluster Allowlist

Note

- The cluster allowlist can only be configured through **kscfg**.
- The CipherTrust Manager GUI only allows to **view** allowlist for each node, if enabled. To do so:
 - a. Log on to CipherTrust Manager GUI.
 - b. Go to **Admin Settings > Cluster**.
 - c. In the Cluster window, click the ellipsis on the right side of a node and select **View Node Allowlist**.
It displays the allowed IP list for that specific node.

To configure, SSH to CipherTrust Manager with ksadmin user.

The kscfg commands are as follows:

- `add`: adds IPv4 and IPv6 address to cluster allowlist
- `list`: lists all IPs in the cluster allowlist
- `remove`: removes IPv4 and IPv6 address from the cluster allowlist

Add IPs to allowlist

To add an IP to allowlist, run:

Syntax

```
kscfg cluster allowlist add --ip <node-ip>
```

Example Request

```
kscfg cluster allowlist add --ip 3.94.160.118
```

There will be no response if node is added successfully in a cluster.

Note

You can add only one node IP at a time in the cluster allowlist.

Fetch list of IPs added to allowlist

To get the list of IPs added to allowlist, run:

Syntax

```
kscfg cluster allowlist list
```

Example Request

```
kscfg cluster allowlist list
```

Example Response

```
[
  "3.94.160.118"
]
```

Remove IPs from allowlist

To remove an IP from allowlist, run:

Syntax

```
kscfg cluster allowlist remove --ip <node-ip>
```

Example Request

```
kscfg cluster allowlist remove --ip 3.94.160.118
```

There will be no response if node is removed successfully from the cluster.

CipherTrust Manager System Monitoring

CipherTrust Manager has six mechanisms to monitor its system activity.

- [Server audit and client audit records](#) include general auditable events available in the GUI, CLI, and REST API in a user-readable JSON format, which can be used to create [alarms](#). By default, these are stored using a Loki Grafana microservice, as we recommend. Storage on local database instead of Loki Grafana is available, but deprecated.

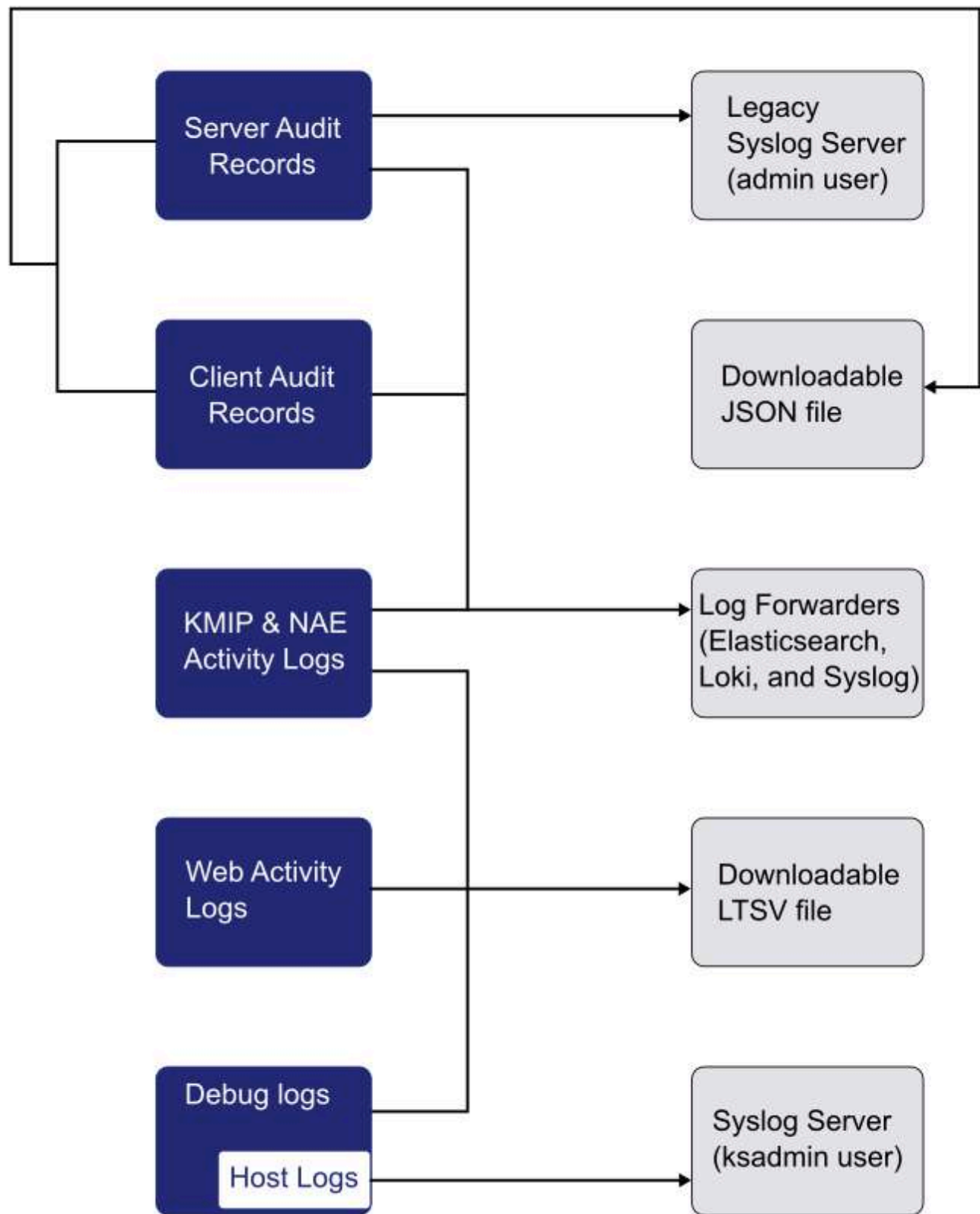
Note

Consult documentation for a specific CipherTrust connector for information on configuring and interpreting client records.

- [Host logs](#) are very detailed messages at the base operating system level.
- [Debug logs](#) are detailed downloadable LTSV files intended for troubleshooting CipherTrust Manager operations with Thales customer support. These include host logs as well as other logs.
- [KMIP and NAE activity logs](#) are messages about operations performed through the [KMIP](#) and [NAE](#) interfaces.
- [Web activity logs](#) are downloadable LTSV format files that include information on operations performed through the web interface.
- [Prometheus Metrics Endpoint](#) allows the [Prometheus monitoring system](#) to scrape the CipherTrust Manager continuously, providing metrics over time to help monitor overall system health, performance, and cryptographic activity. As well, there is a sample configuration available on request, including Prometheus and Grafana data visualization application Docker containers.

The system records and logs can be exported to various external locations.

- Server audit records are [downloadable](#) in JSON format. They can be sent to an legacy external [syslog server](#), and through a log forwarder to [Elasticsearch](#), [Loki](#), or a [syslog server](#).
- Client audit records are [downloadable](#) in JSON format, and can be sent through a log forwarder to [Elasticsearch](#), [Loki](#), or [syslog server](#).
- KMIP and NAE activity logs can be sent through a log forwarder to [Elasticsearch](#), [Loki](#), or a [syslog server](#) and are [downloadable](#) to an LTSV file.
- Web activity logs are [downloadable](#) to an LTSV file.
- Host logs can be sent to an additional [syslog server](#) by the `ksadmin` user.
- Debug logs, including host logs, are [downloadable](#) to an LTSV file as part of the Debug Logs package.



Records

Audit records are logs of auditable events viewable in CipherTrust Manager in a user-friendly JSON format.

Server records contain operations performed by the CipherTrust Manager. You can also view the audit records that are uploaded from the CipherTrust Manager Clients (for example, CTE). Consult documentation for a specific connector for information on configuring and interpreting client records. The *Logs* chapter of *CTE Agent for Linux Advanced Configuration and Integration Guide* contains details on CTE client record format and effects codes.

Audit records can be locally fetched and searched using the CipherTrust Manager REST API, CLI, or the GUI. The records available through these queries are scoped to the [domain permissions](#) of the logged in user.

Both server and client records can be configured to raise [alarms](#).

There are three options to store and display records:

- [Loki Grafana Microservice](#)
- [Log Forwarding](#)
- [Local Database Logging](#)

Loki Grafana Microservice Option

Records for the local node are automatically logged to an internal [Loki Grafana log aggregation microservice](#), where they can be viewed and queried. This option has the best performance locally. In the GUI, Loki records are visible under **Records > Loki Audit Records**. The [retention period for records is configurable](#), with a default value of 240 hours. These records are not replicated in cluster.

Log Forwarding Options

Records can also be sent to an external [logging system through a log forwarder](#). Supported log forwarder types are Elasticsearch, Loki, and Syslog server. These options allow you to view records for a cluster in aggregate.

Local Database Logging Option

Caution

To reduce cluster traffic and disk usage, local database audit logging is disabled by default. Local database logging is deprecated, and we recommend using Loki Grafana microservice and/or enabling forwarding to a remote Syslog.

If local database logging is enabled by restoring a backup, system operation performance can degrade.

To view database audit records for CipherTrust Manager activity, you must first [enable local database logging](#). Then, in the GUI, go to **Records > Server Records**. To view client records in the GUI, go to **Records > Client Records**. The Server Audit Records database table is automatically trimmed when it reaches 10 million records in size.

Server Records Parameters

Parameters (Columns)	Description
Status	Indicates whether an operation was successful or failed.
Event Created	Date and time when the event was created. Time is in UTC time zone.
Severity	Severity of the record. Severity types are: • INFO • CRITICAL • WARNING • ERROR
Message	A short description of the operation performed.
Service	Indicates which internal microservice performed the operation. This is mostly valuable for troubleshooting purposes in conjunction with customer support.
Details	Details associated with the operation performed. These include <code>user_id</code> and <code>client_id</code> to show which entity performed the operation.

Loki Grafana Microservice

You can [list and query records](#) on the Loki Grafana Microservice, and [download the records to a file](#). As well, you can configure the [retention period](#), after which records are deleted.

Note

All the CCKM-managed AWS/Azure keys that are going to expire in the next 10 days are displayed under the Loki Audit Records. After the CipherTrust Manager starts, a back-end job runs every 24 hours to:

1. Fetch the list of AWS/Azure keys going to expire in the next 10 days.
2. Log records for a batch of every 20 keys.

These records are available under **Records > Loki Audit Records > Server Records**.

Listing and Querying Records With Loki Grafana Microservice

You can query both server audit records and client audit records from the local CipherTrust Manager node to return records. The available records depend on the configured [retention period](#) after which records are deleted. The default retention period is 240 hours.

Note

Only data for the local node is logged to the Loki Grafana microservice, not any other cluster nodes.

View and Query Records in the GUI

[Application administrators](#), such as the `admin` user can view and query records. There is a basic query and an advanced query to limit which Loki audit records are displayed.

To view records, navigate to **Records>Loki Audit Records**. Select **Server Records** or **Client Records**.

Basic Query

Select the filter icon () to filter on particular fields or within a specified time range. The following parameters are available to filter on:

- Severity
- Status
- By
- Source
- Client IP
- Message
- Max Records
- Date and Time
- Start Date
- End Date

Select **Apply Query** to view which records are returned.

Advanced Query

Select **Advanced Query** to input a query following [Loki Grafana's LogQL syntax](#). If you have applied a basic query, it is already inputted in LogQL syntax. There are options to [save and load queries](#) from the **Advanced Query** menu.

Select **Apply Query** to view which records are returned.

List and Query Records with ksctl

[Application administrators](#), such as the `admin` user can list records in the Loki Grafana microservice with `ksctl loki query_range --server-audit-records` and `ksctl loki query_range --client-audit-records`.

The query parameters follow [Loki Grafana's LogQL syntax](#) for doing a query over a specified time range. Use `ksctl loki query_range` commands to input parameters for `limit`, `start`, `end`, `step`, `interval`, and `direction`.

Save Queries With Loki Grafana Microservice

The CipherTrust Manager web console UI allows you to save and load queries for convenience.

We also provide two public server records queries available to load for **Google CSE Records** and **Google Cloud EKM Records**, which are [CipherTrust Cloud Key Manager \(CCKM\) integrations](#).

To save a query:

1. Login as an [Application administrator](#), such as the `admin` user.
2. Navigate to **Records>Loki Audit Records**. Select **Server Records** or **Client Records**.
3. Select **Advanced Query**.

A text box appears.

4. Input a query following [Loki Grafana's LogQL syntax](#).
5. Optionally, select **Apply Query** to view which records are returned.
6. Select **Save Query**.

The **Save Query** popup window appears.

7. Provide a **Name** and review the **Query Preview**.

Note

You can only save the query as **Private**.

8. Select **Save** to finalize.

To load a query:

1. Login as an [Application administrator](#), such as the `admin` user.
2. Navigate to **Records>Loki Audit Records**. Select **Server Records** or **Client Records**.

3. Select **Advanced Query**.

A text box appears for query input.

4. Select **Load Saved Query**.

The **Load Saved Query** popup appears.

5. Select one of the saved queries on the left.

Two public queries for CCKM integrations, **Google CSE Records** and **Google Cloud EKM Records**, are available, as well as any previously saved private queries.

6. Select **Load**.

The query is loaded and applied.

Download Loki Audit Records to a File

You can download the Loki Audit Records displayed in the CipherTrust Manager web console UI to a file.

To download the records to a file:

1. Login as an [Application administrator](#), such as the `admin` user.
2. Navigate to **Records>Loki Audit Records**. Select **Server Records** or **Client Records**.
3. [Adjust the query](#) as desired, to limit which Loki audit records will be downloaded.
4. Click **Download Logs** in the upper right corner.

The logs are downloaded to a file `loki_audit_log.txt`.

Configuring Retention Period for Loki Grafana Microservice

By default, all records generated since the CipherTrust Manager is deployed are retained in Loki. To preserve disk space, users in the 'admin' group can set a retention period for which Loki logs are retained.

The retention period has the following characteristics:

- The minimum value is 24 hours.
- A valid value must be a multiple of 24 hours.
- The retention period must be a string that can be parsed using Go library's `time.Duration` type, such as `24h`.

To set the Loki configuration retention period:

```
ksctl loki config modify --retention-time "<desired_retention_time>"
```

To view the Loki configuration retention period:

```
ksctl loki config get
```

Configuring remote Syslog server

Records can optionally be sent to one or more external Syslog server(s). By default, audit records are stored in the Loki Grafana Microservice, and continue to do so even if Syslog connections are configured. Each server audit record will be sent to each configured Syslog connection.

Refer to [Log Forwarding](#) for instructions on configuring a new connection to a Syslog server on the CipherTrust Manager console. Refer to the API guide, CLI Guide, or the GUI for more details.

Note

Upgraded CipherTrust Manager instances can have [existing syslog connections through Admin Settings](#), which continue to be supported. Syslog servers configured as log forwarders can forward client audit records, while syslog servers configured through **Admin Settings** cannot.

A Syslog connection can either use UDP or TCP + TLS as the transport protocol. When TCP + TLS is used a trusted CA certificate in PEM format must also be provided.

All Syslog messages are generated with severity notice and facility local0.

An example entry looks as follows:

```
2019-08-12 06:22:16 keysecure1 KeySecure: <134>1 2019-08-12 06:22:16 keysecure KeySecure - b7999852-dd64-46e9-b924-c3999eca9fad [msg="Update Syslog Connection" sev="6" details="'Update Syslog Connection' succeeded" ({ "createdAt": "2019-08-12T06:22:16.113081Z", "details": { "id": "e847e529-d331-45f9-a494-83ee7ce6ab69" }, "message": "Update Syslog Connection", "service": "kylo", "success": true, "username": "admin", "severity": "info", "lineage": "", "source": "" } ) ]
```

The first time stamp is generated and added by Syslog and the second time stamp is the time of the actual audit record. The time stamps use the UTC time zone.

In a multi-node clustered environment, the Syslog connection configurations are automatically synchronized and each node is aware of all Syslog servers. The Syslog message is sent from the currently active node. This means that if an event that results in a server audit record is performed on node 1, the Syslog message originates from node 1. Similarly, if a server audit event is performed on node 2, the Syslog message originates from node 2.

Note

The Syslog connection configuration can take up to five minutes to apply to all nodes in the cluster.

Local Database Audit Logging

Local database logging of server and client audit records is disabled by default, to reduce cluster traffic and disk usage. Local database logging is deprecated; for most deployments, we recommend to view records on [Loki Grafana microservice](#), and to enable logging using a remote [Syslog server](#).

Caution

If local database logging is enabled by restoring a backup, system operation performance can degrade.

If local database audit logging is [enabled](#), you can [list](#) and [query records](#) on the local database.

Enabling and disabling Local Database Audit Logging

If you have a production system and cluster with a limited load, you might like to enable local database logging, which is disabled by default.

Enabling Example

```
ksctl properties modify -n ENABLE_RECORDS_DB_STORE -p true
```

Note

This is a cluster wide setting and only needs to be run against one node in the cluster.

The Server Audit Records database table is automatically trimmed when it reaches 10 million records in size. If a permanent record is required, syslog should be used and the audit records stored elsewhere.

If you wish to later disable local database logging, run:

```
ksctl properties modify -n ENABLE_RECORDS_DB_STORE -p false
```

Listing and Querying Records With Local Database Logging

If your server and client audit records are logged to [the local database](#), you can list server audit records with `ksctl records list` and client audit records with `ksctl client-records list`.

The response looks like:

```
{
  "skip": 0,
  "limit": 10,
  "total": 40,
  "Resources": [
    {
```

```

    "id": "18ef98f1-0704-4dec-a6f0-e87a581ce52f",
    "uri": "kylo:kylo:audit:records:18ef98f1-0704-4dec-a6f0-e87a581ce52f",
    "account": "kylo:kylo:admin:accounts:kylo",
    "application": "system:system:admin:applications:system",
    "devAccount": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2018-05-22T18:34:05.239823Z",
    "message": "Create Token",
    "service": "kylo",
    "requestId": "45a8dcef-0b9e-44e2-b3e1-51b3d2ef99ca",
    "instanceId": "",
    "instanceTime": "0001-01-01T00:00:00Z",
    "instanceTz": "",
    "success": true,
    "username": ""
  }
]
}

```

There will be a number of records returned (10 shown in the example above). Most have been removed in the above example for space reasons. Note that the response indicates that there are 40 records available ("total": 40,). These records can be returned with options in the **ksctl records** and **ksctl client-records** commands to filter which records are returned. Type the following commands to get more information:

```

ksctl records list --help
ksctl client-records list --help

```

Configuring alarm triggers based on a record

CipherTrust Manager can be configured to trigger one or more alarms based on the content of a record.

Alarm configurations apply to both records stored in the Loki Grafana microservice, and records stored in the database, if [enabled](#).

To create an alarm configuration:

1. Log on to the CipherTrust Manager console as administrator.
2. In the left pane, click **Records**. Click **Loki Audit Records**.
3. Click the **Alarm Configurations** tab.
4. Click **Configure Alarm Configuration**. The **Alarm Config - Create** dialog box is displayed. Specify the following details:
 - a. **Severity**: Defines the nature of the alarm. It can be Info, Error, Critical, or Warning.

Note

This value is independent from the severity included with server records. If you want an alarm to

inherit the severity of a server record, you can create the alarm configuration using the REST API or CLI and omit the severity option.

- b. **Name:** Name of the alarm.
- c. **ErrorThreshold:** The maximum number of times the trigger condition is met in the specified time **Interval** before the alarm is raised.
- d. **Interval:** The time duration in seconds, in which if the trigger condition is met for a defined number of times (**Error Threshold** value) , the alarm is raised.
- e. **Description:** A description for the alarm.
- f. **Condition:** Conditional properties/events that triggers an alarm.

Note

Multiple alarms may be triggered using a single record.

Example (For Windows)

Note

In Windows environment, ensure to use backslash (\) as an escaped character with double quotes (\").

Request

```
ksctl records alarm-configs create --name "Weak RSA Key" --description "RSA key should be 2048 bits or higher" --severity "critical" --condition "input.success; input.message == \"Create Key\"; input.details.algorithm == \"RSA\"; input.details.size <= 1024"
```

Example (For Linux)

Request

```
ksctl records alarm-configs create --name "Weak RSA Key" --description "RSA key should be 2048 bits or higher" --severity critical --condition 'input.success; input.message == "Create Key"; input.details.algorithm == "RSA"; input.details.size <= 1024'
```

Response (For Windows and Linux):

```
{
  "id": "4bf23269-3985-42f3-995c-8f3c8f8b6543",
  "uri": "kylo:kylo:audit:alarm-configs:4bf23269-3985-42f3-995c-8f3c8f8b6543",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2019-08-27T14:37:19.145946719Z",
  "name": "Weak RSA Key",
  "condition": "input.success; input.message == \"Create Key\"; input.details.algorithm == \"RSA\"; input.details.size <= 1024",
  "description": "RSA key should be 2048 bits or higher",
  "severity": "critical"
}
```

To trigger this alarm, create a weak RSA key:

```
ksctl keys create --name mysmallkey --alg RSA --size 1024
```

The alarm should be triggered and this can be confirmed by listing the alarms:

```
ksctl alarms list --state on
```

Here is an abbreviated response for brevity:

```
{
  "resources": [
    {
      "name": "Weak RSA Key",
      "state": "on",
      "description": "RSA key should be 2048 bits or higher",
      "severity": "critical",
      "details": {
        "details": {
          "id": "f2ec44f6977edf2d6ff7fa327b5499301732dc6b395284a71e47ac0346cf72d6",
          "uri": "kylo:kylo:vault:keys:mysmallkey-v0",
          "name": "mysmallkey",
          "size": 1024,
          "ownerId": "local|fa0e4dfa-e6fd-413b-a4a0-48c356cf0557",
          "algorithm": "RSA",
          "usageMask": 3,
          "objectType": "Private Key"
        },
        "message": "Create Key",
        "success": true,
      }
    }
  ]
}
```

Creating Alarm Trigger for Certificate Expiration

To create an alarm trigger (and receive email notifications) for certificate expiration, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as `input.message == "Certificate Expiry Check"`.

Note

The action **Certificate Expiry Check** can be created in the root domain only.

After saving this alarm configuration, the [recipients in the email address](#) list will receive email notifications every time a certificate expiration related record is logged. The emails will contain the certificate IDs, expiration dates, and the domain name.

Creating Alarm Trigger for Client Certificate Expiration

To create an alarm trigger (and receive email notifications) for client certificate expiration, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as `input.message == "Client Certificate Expiry Check"`.

Note

The action **Client Certificate Expiry Check** can be created in the root domain only.

After saving this alarm configuration, the [recipients in the email address](#) list will receive email notifications every time a client certificate expiration related record is logged.

The emails will contain the following information:

- Client name
- Client type if available
- Whether certificate is expired
- Expiration duration in days (if `expired` flag is `false`) or expired duration in days (if `expired` flag is `true`)
- Domain name

Creating Alarm Trigger for Connection Certificate Expiration

To create an alarm trigger (and receive email notifications) for the connection certificate expiration, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as `input.message == "Connection Certificate Expiry Check"`.

Note

The action **Connection Certificate Expiry Check** can be created in the root domain only.

After saving this alarm configuration, the [recipients in the email address](#) list will receive email notifications every time a connection certificate expiration related record is logged.

The emails will contain the following information:

- Domain name
- Whether certificate is expired (expired flag set to `true/false`)
- Expiration duration in days (`days_since_expired`, if expired flag is `true`) and (`days_to_expire`, if expired flag is `false`)

Creating Alarm Trigger for Interface Certificate Expiration

To create an alarm trigger (and receive email notifications) for the interface certificate expiration, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as `input.message == "Interface Certificate Expiry Check"`.

Note

The action **Interface Certificate Expiry Check** can be created in the root domain only.

After saving this alarm configuration, the [recipients in the email address](#) list will receive email notifications every time a Interface certificate expiration related record is logged.

The emails will contain the following information:

- Interface name
- Whether certificate is expired (expired flag set to `true/false`)
- Expiration duration in days (`days_since_expired`, if expired flag is `true`) and (`days_to_expire`, if expired flag is `false`)

Creating Alarm Trigger for System Restart

To create an alarm trigger for a system restart, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
input.message == "Restart Services" input.service_name == "service-controller"
```

This alarm will trigger to inform the users about the reboot operation on the CipherTrust Manager.

Creating Alarm Trigger for Service Restart

To create an alarm trigger for a service restart, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
input.message == "Restart Services"
```

This alarm will trigger to inform the users about a restart performed on a service.

Creating Alarm Trigger to Enable Prometheus Metrics

To create an alarm trigger to enable Prometheus metrics on the CipherTrust Manager, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
input.message == "Enable Prometheus Metrics Collection" input.service_name == "platform"
```

This alarm will trigger to inform the users that Prometheus metrics are enabled on the CipherTrust Manager.

Creating Alarm Trigger to Disable Prometheus Metrics

To create an alarm trigger to disable Prometheus metrics on the CipherTrust Manager, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
input.message == "Disable Prometheus Metrics Collection" input.service_name == "platform"
```

This alarm will trigger to inform the users that Prometheus metrics are disabled on the CipherTrust Manager.

Creating Alarm Trigger for Certificate Expiration Warning

To create an alarm trigger for a certificate expiration warning, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
input.message == "Following certificates are expired or going to expire in a few days. To avoid a possible interruption of your service, please renew your certificates." input.message == "Certificate Expiry Check" input.service_name == "certificate-authority"
```

This alarm will trigger to inform the user about the certificates that are going to be expired in near future or have already expired.

Creating Alarm Trigger for Multiple Login Failure Occurrences

To create an alarm trigger for multiple login failures occurrences on the CipherTrust Manager, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
"input.message == "error getting client/user: Wrong username or password." input.message == "Create Token"
```

Creating Alarm Trigger for Failed System Backup using SCP/SFTP

To create an alarm trigger for failed system backup using SCP/SFTP, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
input.details.status == "failed" input.message == "Secure Copy Backup"
```

This alarm will trigger to inform the user about the failure of securely copying a backup to the set destination.

Creating Alarm Trigger for Unauthorized Key Usage

To create an alarm trigger for unauthorized key usage on the CipherTrust Manager, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
input.message == "read on target is not authorized: verdict included  
conditions, which are not supported: ReadKey" input.service_name == "key-vault"
```

This alarm will trigger to inform the user about an unauthorized attempt to access an existing key on the CipherTrust Manager.

Creating Alarm Trigger when Disk/Key Storage is Full

To create an alarm trigger when the disk/key storage on the CipherTrust Manager is full, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
input.message == "Disk Full" input.details.previous_state == "off"  
input.details.service == "Sheepdog" input.details.state == "on" input.service  
== "Sheepdog"
```

This alarm will trigger to inform the user that storage on the CipherTrust Manager has been consumed more than the configured threshold limit of 80%.

Creating Alarm Trigger for Unreachable HSM

To create an alarm trigger when HSM is offline, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
input.message == "HSM is offline" input.details.name == "HSM Offline"  
input.details.service == "Darkstar" input.details.source == "ciphertrust"  
input.details.source_type == "server_record" input.service == "Darkstar"  
input.source == "ciphertrust"
```

This alarm will trigger to inform the user that the CipherTrust Manager is unable to access the HSM.

Note

To trigger this alarm, the HSM should be configured on the CipherTrust Manager as a Root of Trust.

Creating Alarm Trigger when Cluster Node is Unavailable

To create an alarm trigger when a node in a cluster goes down, follow the steps mentioned in the [Configuring alarm triggers based on a record](#) section, and specify the **Condition** action as:

```
input.message == "Cluster Node Down" input.details.service == "dbmgr"  
input.details.source_type == "server_record" input.service == "dbmgr"
```


This alarm will trigger to inform the user about the unavailability of a node within the cluster.

Alarms

Alarms are CipherTrust Manager's mechanism for notifying administrators to problems with the state of the CipherTrust Manager, one of its nodes, or select clients such as CTE. Alarms are raised based on [Server or Client Records](#) indicating CipherTrust Manager or a client is not healthy or is not configured securely. Alarms can be fetched via the REST API, CLI, or the GUI.

Alarm states are listed in the following table:

State	Description
off	The alarm is inactive and does not need to be investigated.
on	The alarm is active and should be investigated. It remains active until the condition that triggered it is no longer valid.
unknown	The alarm's state could not be determined and should be investigated. Typically this occurs when the service responsible for triggering the alarm failed to communicate its state (e.g. the service is down).

The CipherTrust Manager has the following built-in alarms for server problems:

Name	Severity	Trigger	Remediation
Disk Full	Critical	The root file system's 'used space' percentage exceeds the configured threshold of 80%.	Increase the node's disk/partition/file-system, or replace the node with a new node that has sufficient storage.
NAE TLS Disabled	Critical	The system is started with NAE's interface mode configured to not use TLS link to interfaces .	Modify NAE's interface mode link to interfaces to one of the options that specifies TLS and restart link to services .
HSM Offline	Critical	The system cannot access the HSM after more than 15 seconds. NOTE: If connectivity to the HSM is not restored after 5 minutes, all services are shut down until the HSM becomes available.	Restore connectivity to the HSM.
Cluster Node Down	Critical	A node within a cluster is down.	Restore connection of the down node to the cluster.
Cluster Node Certificate Expiration	Critical	Automatic renewal failed for an internal CipherTrust Manager certificate that is	Re-join the node to the cluster to retrieve a current cluster certificate.

Name	Severity	Trigger	Remediation
		used for database access.	In the Loki audit records , look for the cluster node which logged the message "Cluster cert about to expire ..." followed by a date. This the node to re-join to the cluster.
Syslog Connection Offline The behavior of this alarm is different from other built-in alarms. It doesn't appear in the list of built-in alarms on the CipherTrust Manager. It is created at runtime only when the alarm is triggered and gets destroyed after the remediation is achieved.	Critical	A Syslog connection goes offline.	Restore connectivity to the specified Syslog connection.
Deprecated TLS version Enabled	Critical	When TLSv1.0 or/and TLSv1.1 is set as minimum TLS version on the CipherTrust Manager for NAE-KMIP interface.	Set minimum TLS version to TLSv1.2 or higher.
License Violation	Critical	A Virtual CipherTrust Manager k170v instance has more than four CPUs assigned to it.	Upgrade to a valid k470 license, or assign four or fewer CPUs to the k170v.
License Expiration	Warning	One or more of the licenses is set to expire in fewer than 90 days. The description indicates the licenses and number of days until expiry.	
Ransomware Activity	Warning	A Ransomware activity is detected on a client. Please check the RWP client <clientName> for more details.	Power off/reboot or isolate the system, investigate and recover from a Ransomware attack. Refer to How to Investigate and Recover After a Ransomware Attack for details.

The CipherTrust Manager server also has dynamic alarms that are triggered based on server or client record conditions. Refer to: [Configuring alarm triggers based on a record](#).

Consult documentation for a specific connector for information on interpreting client alarms.

To list all alarms

```
$ ksctl alarms list
```

Example response:

```
{
  "skip": 0,
  "limit": 10,
  "total": 8,
  "resources": [
    {
      "name": "License Violation",
      "state": "off",
      "triggeredAt": "2021-09-30T10:07:52.5529Z",
      "description": "Alarm triggers if CPU count exceeds the limit allowed
in the license",
      "severity": "critical"
    },
    {
      "name": "Cluster Node Certificate Expiration",
      "state": "off",
      "triggeredAt": "2021-09-30T10:07:52.547729Z",
      "description": "Alarm triggers 30 days to certificate expiration (the
certificate is currently valid for 2.0 years)",
      "severity": "critical"
    },
    {
      "name": "Cluster Node Down",
      "state": "off",
      "triggeredAt": "2021-09-30T10:07:52.541277Z",
      "description": "Cluster nodes down alarm triggers when any node is do
wn (currently all nodes are up)",
      "severity": "critical"
    },
    {
      "name": "Disk Full",
      "state": "off",
      "triggeredAt": "2021-09-30T10:07:47.535595Z",
      "description": "Disk full alarm triggers above 80% of capacity (curre
ntly at 47%)",
      "severity": "critical"
    },
    {
      "name": "HSM Offline",
      "state": "off",
      "triggeredAt": "2021-09-30T10:07:44.730768Z",
      "description": "HSM is offline",
      "severity": "critical"
    },
    {
      "name": "Deprecated TLS version Enabled",
      "state": "unknown",
      "triggeredAt": "2021-09-30T10:07:37.527468Z",
      "description": "Deprecated TLS version(TLSv1.0/1.1) enabled on NAE/KM
IP interface",
      "severity": "critical"
    }
  ]
}
```

```

    },
    {
      "name": "KMIP Debug Logs Unmask Enabled",
      "state": "off",
      "triggeredAt": "2021-09-30T10:07:27.516841Z",
      "description": "Unmasking of sensitive data for KMIP debug logs is en
abled",
      "severity": "critical"
    },
    {
      "name": "NAE TLS Disabled",
      "state": "off",
      "triggeredAt": "2021-09-30T10:07:17.510478Z",
      "description": "TLS is disabled on the NAE interface",
      "severity": "critical"
    }
  ]
}

```

To acknowledge that an alarm is under investigation:

```
$ ksctl alarms acknowledge -i 1650789b-e8e6-4915-91ab-b067e073f39f
```

To clear (i.e. turn off) an alarm:

```
$ ksctl alarms clear -i 1650789b-e8e6-4915-91ab-b067e073f39f
```

Syslogs

Timezone Configuration

CipherTrust Manager server audit records and client audit records are always recorded in UTC time zone, in keeping with [RFC 3339](#). This is important to note when you configure any external logging system such as a log forwarder or legacy syslog connection.

Add a new Syslog Server

The preferred Syslog configuration is through [Connection Manager](#) and [Log Forwarders](#) commands and menus. The syslog configuration is specific to the [domain](#) it is created in.

Note

Upgraded CipherTrust Manager instances can have [existing syslog connections](#) through [Admin Settings](#), which will be removed in a future release. Syslog servers configured as log forwarders can forward client audit records, while syslog servers configured through **Admin Settings** cannot.

Add a Syslog Connection with Connection Manager

The preferred Syslog configuration is through [Connection Manager](#). Provide the following values:

- **Host:** IP address or hostname of the Syslog server.
- **Port:** port number for connecting to the Syslog server.
- **Transport Format:** select the transport mode for sending data. The **TLS** mode requires a trusted CA certificate in the PEM format.

Note

If you set the transport format to **UDP**, log messages are limited to a size of 1024 bytes. After this size, the log message is truncated.

- **CA Cert:** either upload the CA certificate or paste the certificate content. Make sure the server certificate contains the valid IP SANs.
- **Upload CSR:** select and click **Upload CSR** to upload the trusted CA certificate from your machine.
- **Text:** select and paste the certificate content in the text field.
- **Message Format:** select the log message format.

Add a Syslog Log Forwarder

Once you have added a syslog connection, you can create a syslog log forwarder on CipherTrust Manager to forward KMIP activity logs, NAE activity logs, server audit records, and client audit records to Syslog server.

To add a Syslog log forwarder, you must provide:

- a connection ID of the Syslog connection manager (refer to [Connection Manager](#) for details)
- a connection name for the log forwarder configuration

You can optionally activate/deactivate:

- forward logs for activity kmip
- forward logs for activity nae
- forward logs for client audit records
- forward logs for server audit records

Syntax for Syslog

```
ksctl log-forwarders add syslog --name <name of log forwarder> --connection-id <Syslog ConnectionID/Name> --forward-client-audit-records <true/false> --forward-logs-activity-kmip <true/false> --forward-logs-activity-nae <true/false> --forward-server-audit-records <true/false>]
```

Log Forwarder Redirection

Log forwarder redirection is a setting to send audit records, NAE activity logs, or KMIP activity logs from a child domain to the log forwarder configured for a higher-level domain.

When an audit record, NAE activity log, or KMIP activity log is created in a child domain with log forwarder redirection, CipherTrust Manager checks the domain's immediate parent for a configured log forwarder. If the immediate parent also has log redirection enabled, CipherTrust Manager also checks its parent, the child domain's grandparent, for a configured log forwarder. CipherTrust Manager continues checking for log forwarders further up the domain hierarchy every time it finds the log redirection setting. Every time CipherTrust Manager finds a configured log forwarder in a domain, it forwards log messages to that log forwarder.

Note

- Log forwarder redirection can be enabled even if higher level domains do not have a log forwarder configured. In that situation, logs from the child domain are not forwarded.
- If you have log redirection enabled and log forwarders configured at multiple consecutive levels in a domain hierarchy, the same log message can be forwarded to multiple log forwarders. To consolidate messages, we recommend configuring only one log forwarder in a domain hierarchy, at the highest domain level possible.
- We recommend that administrators of parent domains are aware of log forwarding settings for child domains. If there is a need to track where logs from a particular domain are forwarded, employees from your organization might need to login to several domains to view log forwarder and log forwarder redirection settings.

Users in the Domain Admins or Application Administrators group, such as `admin` can enable or disable log forwarder redirection. When a new child domain is created, log forwarder redirection is enabled by default. Existing child domains which were created in a version lower than 2.16.0 have log forwarder redirection disabled after upgrade.

Enabling or disabling log forwarder redirection in web console UI:

- Login to the child domain as a user in the Domain Admins or Application Administrators group, such as `admin`.
- Navigate to **Admin Settings > Logs**.
- Click the **Redirect Log Forwarder** toggle to enable or disable log forwarder redirection.

Enabling or disabling log forwarder redirection in ksctl CLI:

Login to the child domain as a user in the Domain Admins or Application Administrators group, such as `admin`. For example:

```
ksctl login --user <user_with_domain_admin_permission> --password <user_password>
--domain <domain_name_or_id>
```

- The following command disables log forwarder redirection.

```
ksctl domains log-forwarders-redirection disable
```

- The following command enables log forwarder redirection.

```
ksctl domains log-forwarders-redirection enable
```

- The following command displays log forwarder redirection status

```
ksctl domains log-forwarders-redirection status
```

Modifying Legacy Connection to a Syslog Server

Updating a syslog server connection managed through Connection Manager is described on the [Connection Manager page](#). The following instructions describe how to update legacy syslog server connections managed through admin settings. The table below indicates editable parameters.

Parameter	Description
Hostname or IP address	Hostname or IP address of the Syslog server.
Port	Port of the Syslog server. The default port is 514.
Log Format	Format in which the audit records are transferred to the Syslog server. The options are: ▪ CEF ▪ LEEF ▪ RFC5424 ▪ Plain Message The default log format is RFC5424. This format adheres to the Syslog Protocol RFC 5424 guidelines.
Transport	Transport protocol for the Syslog connection. The options are UDP, TCP, and TLS. The default protocol is UDP. With UDP, log messages are limited to a size of 1024 bytes. After this size, the log message is truncated.

Parameter	Description
Certificate	Trusted CA certificate in the PEM format. This field is available when the transport protocol is TLS.

To modify the connection to a legacy Syslog server:

1. Log on to the CipherTrust Manager console as administrator.
2. Go to **Admin Settings > Syslog**. The Syslog section is displayed on the right. This section displays the configured connections to Syslog servers.
3. Click the ellipsis icon corresponding to the desired connection and click **Edit**.

Note

To delete a connection, click **Delete**.

4. Modify the fields as required.
5. Save the changes.

Managing Syslog Messages Redirection to Parent Domain (Legacy Syslog) using ksctl

Syslog messages redirection allows you to send the legacy syslog messages of the current domain to the syslog server configured in its parent domain. If the current domain is receiving the syslog messages from its child domain, those syslog messages will also be sent to the syslog server configured in the parent domain of the current domain.

Debug Logs

The CipherTrust Manager maintains a variety of debug logs to record administrative actions, network activity, cryptography requests, and more. These logs can be useful for debugging, error handling, troubleshooting, and for chronologically tracing failures and system events with Thales customer support. [Server Audit Records](#) are recommended for more routine monitoring.

All of these logs are specific to the node, and are not clustered or included in backup files. High access permissions are required to download these logs. The user must be part of the System Admins and Admin groups, and the user must be logged into the root domain.

The following types of logs are recorded:

- Connection request received
- Configuration changes
- Client requests

-
- Service starts, stops, and restarts
 - System and user input errors
 - Successful and failed operations

The debug logs can be downloaded using the CipherTrust Manager CLI, API, and web console.

The downloaded file contains following debug log files in *.gzip* format:

- auth.log
- cloud-init.log
- hostd.log
- kern.log
- keysecure.system.log
- syslog

The extracted file type is LTSV. The `ksadmin` user can also forward a subset of these logs, called **host logs**, to an external syslog server.

Activity Logs

In addition to the debug logs, various activity logs can be extracted in a *.gzip* package. As with the debug logs, they are only available to download in the root domain to users who are part of the **System Admins** and **Admin** groups. The downloaded logs include activity from every domain. **Log forwarders** allow finer control, forwarding activity logs from specific domains.

Web activity logs are always included are part of the web console download. In the CLI and API, these logs are available individually or as part of the `all-logs` option.

As well, KMIP and NAE Crypto Activity logs can be **enabled as a system property**. They are then included in the web console download. In the CLI and API, these logs are then available individually or as part of the `all-logs` option.

Downloading Logs using ksctl

To download CipherTrust Manager logs as a gzipped tarball, run:

Syntax

```
ksctl logs download --file <filename> --ca-id <Local-CA-ID> --type <Type-of-Logs>
--start-datetime <start-date-time-of-logs> --end-datetime <end-date-time-of-logs>
```

- file - file path where the gzipped tarball is saved (for example, logs.tar.gz).
- ca-id - ID of the trusted local CA to issue a certificate and sign the log file. You can optionally provide a Certificate Authority (CA) for **issuing a signing certificate**. If no CA is provided, the CipherTrust Root CA issues

the certificate.

- **type** - type of logs to download. For every log type, both current and rotated logs are downloaded. Valid values for **type** parameter are:
 - **all-logs**: Includes all types of logs available to the current user and domain.
 - **debug-logs**: Includes the debug logs on the CipherTrust Manager.
 - **kmip-activity-logs**: Includes the KMIP activity logs on the CipherTrust Manager
 - **nae-activity-logs**: Includes the NAE activity logs on the CipherTrust Manager.
 - **web-activity-logs**: Includes the web activity logs on the CipherTrust Manager.
- **start-datetime** - filters results based on the start date and time of logs. The timestamp should be in the UTC format. For example, 2023-09-11 11:24:22. This is an optional parameter.
- **end-datetime** - filters results based on the end date and time of logs. The timestamp should be in the UTC format. For example, 2023-09-11 11:24:22. This is an optional parameter.

To download the NAE Crypto Activity logs

```
ksctl logs download --file <filename> --type "nae-activity-logs"
```

To download all debug logs

```
ksctl logs download --file <filename> --type "debug-logs"
```

To download all logs

```
ksctl logs download --file <filename> --type "all-logs"
```

To download the logs that are within specified start date and end date

```
ksctl logs download --file <filename> --type "all-logs" --start-datetime '2023-09-11 11:24:22' --end-datetime '2023-10-30 12:24:22'
```

Downloading Logs from the Web Console

`activity.kmip.log` and `activity.nae.log` files can be present in the downloaded package if **KMIP and NAE Crypto activity logs are also enabled**. `activity.web.log` is always present.

Perform the following steps to download these logs using the GUI:

1. Navigate to **Admin Settings > Logs**.
2. Change the CA to [issue a signing certificate](#), if desired. By default, the CipherTrust Root CA issues the

certificate.

3. Download the logs:

- To download the current **Debug Logs**, click **Download**.
- To download all logs, enable the **Download all logs** option and then click **Download**. The downloaded log file will contain all logs that are up to 4 weeks old.

Preserving integrity of downloaded logs

CipherTrust Manager preserves integrity of downloaded logs by performing following steps:

- SHA512 hash of downloaded zip file is calculated, which is then signed by a dynamically generated asymmetric key pair.
- Certificate is issued by the CipherTrust Manager CA. This CA is selected while downloading the logs. By default, the CipherTrust Root CA issues the certificate.
- Certificate to verify the signed hash is contained in the downloaded content.

Verifying and viewing the downloaded logs

You must have installed OpenSSL on your machine to verify the log file. You can use the procedure in both Windows and UNIX/Linux environments.

Perform the following steps on the downloaded log file to verify its integrity:

1. Unzip (extract) the log file.
2. View the downloaded logs. The steps to view the logs depend on operating system.

Caution

Do not change, add, or delete any file in the extracted logs directory before verification. These actions will cause signature verification to fail.

- **For Windows users**

Note

Do not use WinRAR for extracting the log files. Use of 7-Zip is recommended.

- a. Extract the `tar.gz` file using 7-Zip.
- b. Open files ending with the `.log` extension in a reader to view the logs. The available logs depend on download options.

- **For UNIX/Linux users**

Use the following commands in the Terminal:

- a. Decompress the `tar.gz` using this command: `gzip -d keySecureLogs.tar.gz`

You will get `keySecureLogs.tar`.

- b. Extract the log files using this command: `tar xvf keySecureLogs.tar`

The available logs depend on download options.

3. Run the `verify-logs.sh` script in the extracted logs directory.

```
./verify-logs.sh
```

The response for a successful verification is `There is no difference in log files digest`.

The response for a failed verification is `There are differences in log files digest`.

Managing KMIP/NAE Activity Log Settings

The CipherTrust Manager logs:

- All KMIP activities and operations
- All NAE crypto activities and operations

Note

Logging both KMIP and NAE activities is memory-intensive due to the high volume of cryptographic operations performed every second. As a result, enabling this setting can significantly degrade the performance of KMIP and NAE operations. For this reason, it is not recommended to enable these logging settings in a production environment.

These activity logs are output in JSON format. You can also send these logs to [log forwarders](#).

Enabling/Disabling KMIP or NAE Crypto Activity Logs through the Web Console

The CipherTrust Manager logs all KMIP and NAE crypto activities and operations. You can configure CipherTrust Manager to keep a record of various KMIP and NAE crypto activities and operations. [Application Administrators](#),

such as the `admin` user, can enable and disable these logs in the root domain.

To record KMIP or NAE crypto activities and operations:

1. Login to the root domain as an Application Administrator.
2. Navigate to **Admin Settings > Properties**.
3. In **LOGS PROPERTIES**, select the **KMIP Activity Logs** toggle button to enable/disable KMIP activity logs.
4. In **LOGS PROPERTIES**, select the **NAE Crypto Activity Logs** toggle button to enable/disable NAE crypto activity logs.

These logs are now available for download through the [web console](#), [CLI](#), and [API](#), in the root domain for users who are part of **System Admins** and **Admin** groups.

As well, these logs can now be sent to [log forwarders](#).

Enabling/Disabling KMIP or NAE Crypto Activity Logs through ksctl

To enable KMIP activity logs, run:

```
ksctl properties modify --name ENABLE_KMIP_ACTIVITY_LOGS --value true
```

To disable KMIP activity logs, run:

```
ksctl properties modify --name ENABLE_KMIP_ACTIVITY_LOGS --value false
```

To enable NAE crypto activity logs, run:

```
ksctl properties modify --name ENABLE_NAE_ACTIVITY_LOGS --value true
```

To disable NAE crypto activity logs, run:

```
ksctl properties modify --name ENABLE_NAE_ACTIVITY_LOGS --value false
```

Log Forwarding

Warning

After upgrading from 2.8 or below, the old connections of Loki and Elasticsearch created using the log forwarders API will be deleted.

Users who belong to the `System Admins` group can forward [server and client audit records](#) and [KMIP and NAE activity logs](#) to Elasticsearch, Loki, or a syslog server.

Elasticsearch and Loki are part of logging stacks — Elastic Stack or Grafana Loki — which provide powerful tools for querying, analyzing and visualizing CipherTrust Manager log entries. See [Elastic Stack documentation](#) and [Grafana Loki documentation](#) for full details on logging system operations and capabilities.

CipherTrust Manager always stores logs locally in addition to forwarding to configured log forwarders.

In a clustered environment, log forwarder configuration is replicated across the cluster. The currently active node sends log forwarder messages. This means that if you perform a logged operation on a node, that node sends the log record to the log forwarder.

Note

Currently, the log forwarders are not configured to use the system's proxy configuration. If proxy is configured, the log forwarders bypass the proxy servers.

Note

The server certificate associated with Syslog, Loki, and Elasticsearch log forwarders are checked against the Certificate Revocation List (CRL). If the certificate contains a CRL and is identified as revoked, the connection will be terminated, and log forwarding will stop. Additionally, an alarm will be activated for that specific connection.

If an error occurs during the evaluation of the certificate's revocation status, the connection will remain in a soft fail state and will be allowed to continue. For instance, this may happen if the CRL is unable to be downloaded.

The CipherTrust Manager allows to fetch the most recent list of revoked certificates (CRL) from issuing authorities by setting up a [Synchronize CRLs Scheduler](#).

High-Level Workflow

To configure CipherTrust Manager to forward to a logging system, there are two phases.

1. Create a log forwarder connection in [Connection Manager](#). This establishes communication between CipherTrust Manager and the external logging system.

Note

CipherTrust Manager can have a total of 64 log forwarder connections. Each [Elasticsearch](#), [Loki](#), and [Syslog](#) connection is counted towards the 64 connection total.

2. Create a log forwarder resource on CipherTrust Manager. This object controls additional CipherTrust Manager-specific settings for the logs and records before sending them to the external logging system.

- [Configuring Elasticsearch Log Forwarder](#)

Controls which index names are applied to the different CipherTrust Manager log and record types.

- [Configuring Loki Log Forwarder](#)

Controls which labels are applied to the different CipherTrust Manager log and record types.

- [Configuring Syslog Log Forwarder](#)

Controls which CipherTrust Manager log and record types are forwarded to syslog.

Timezone Configuration

CipherTrust Manager server audit records and client audit records are always recorded in UTC time zone, in keeping with [RFC 3339](#). This is important to note when you configure any external logging system such as a log forwarder or legacy syslog connection.

Log Forwarder Redirection

Log forwarder redirection is a setting to send audit records, NAE activity logs, and KMIP activity logs from a child domain to the log forwarder configured for a higher-level domain.

When an audit record, NAE activity log, or KMIP activity log is created in a child domain with log forwarder redirection, CipherTrust Manager checks the domain's immediate parent for a configured log forwarder. If the immediate parent also has log redirection enabled, CipherTrust Manager also checks its parent, the child domain's grandparent, for a configured log forwarder. CipherTrust Manager continues checking for log forwarders further up the domain hierarchy every time it finds the log redirection setting. Every time CipherTrust Manager finds a configured log forwarder in a domain, it forwards log messages to that log forwarder.

Note

- Log forwarder redirection can be enabled even if higher level domains do not have a log forwarder configured. In that situation, logs from the child domain are not forwarded.
- If you have log redirection enabled and log forwarders configured at multiple consecutive levels in a domain hierarchy, the same log message can be forwarded to multiple log forwarders. To consolidate messages, we recommend configuring only one log forwarder in a domain hierarchy, at the highest domain level possible.
- We recommend that administrators of parent domains are aware of log forwarding settings for child domains. If there is a need to track where logs from a particular domain are forwarded, employees from your organization might need to login to several domains to view log forwarder and log forwarder redirection settings.

Users in the Domain Admins or Application Administrators group, such as `admin` can enable or disable log forwarder redirection. When a new child domain is created, log forwarder redirection is enabled by default. Existing child domains which were created in a version lower than 2.16.0 have log forwarder redirection disabled

after upgrade.

Enabling or disabling log forwarder redirection in web console UI:

1. Login to the child domain as a user in the Domain Admins or Application Administrators group, such as `admin`.
2. Navigate to **Admin Settings > Logs**.
3. Click the **Redirect Log Forwarder** toggle to enable or disable log forwarder redirection.

Enabling or disabling log forwarder redirection in ksctl CLI:

Login to the child domain as a user in the Domain Admins or Application Administrators group, such as `admin`. For example:

```
ksctl login --user <user_with_domain_admin_permission> --password <user_password>
--domain <domain_name_or_id>
```

- The following command disables log forwarder redirection.

```
ksctl domains log-forwarders-redirection disable
```

- The following command enables log forwarder redirection.

```
ksctl domains log-forwarders-redirection enable
```

- The following command displays log forwarder redirection status

```
ksctl domains log-forwarders-redirection status
```

Configuring Elasticsearch Log Forwarder

The CipherTrust Manager log forwarder is compatible with Elasticsearch version 7 and 8.

You can add an index name to KMIP activity logs, NAE activity logs, server audit records, and client audit records to help with queries in the Elasticsearch environment.

To add an Elasticsearch connection you need to provide the following values:

- a connection ID of the Elasticsearch connection manager (refer to [Connection Manager](#) for details)
- a connection name for the log forwarder configuration

You can optionally provide:

- an index name for KMIP activity logs
- an index name for NAE activity logs

- an index name for server audit records
- an index name for client audit records

Syntax for Elasticsearch

```
ksctl log-forwarders add elasticsearch --name <name of log forwarder> --connection-id <ES connectionID/Name> --index-activity-kmip <kmip_index_name> --index-activity-nae <nae_index_name> --index-server-audit-records <server_audit_records_index_name> --index-client-audit-records <client_audit_records_index_name>
```

Configuring Loki Log Forwarder

You can add labels to KMIP activity logs, NAE activity logs, server audit records, and client audit records to help with queries in the Loki Grafana environment.

To add a Loki log forwarder, you must provide the following values:

- a connection ID of the Loki connection manager (refer to [Connection Manager](#) for details)
- a connection name for the log forwarder configuration

You can optionally provide:

- labels field for KMIP activity logs
- labels field for NAE activity logs
- labels field for server audit records
- labels field for client audit records

Syntax for Loki

```
ksctl log-forwarders add loki --name <name of log forwarder> --connection-id <Loki ConnectionID/Name> --labels-activity-kmip <kmip_label> --labels-activity-nae <nae_label> --labels-server-audit-records <server_audit_records_label> --labels-client-audit-records <client_audit_records_label>
```

Configuring Syslog Log Forwarder

Note

- Upgraded CipherTrust Manager instances can have existing syslog connections through Admin Settings, which continue to be supported. Syslog servers configured as log forwarders can forward client audit

records, while syslog servers configured through **Admin Settings** cannot.

- The [Syslog message redirection](#) is not supported in Syslog log-forwarders.

Once you have added a syslog connection, you can create a syslog log forwarder on CipherTrust Manager to forward KMIP activity logs, NAE activity logs, server audit records, and client audit records to Syslog server.

To add a Syslog log forwarder, you must provide:

- a connection ID of the Syslog connection manager (refer to [Connection Manager](#) for details)
- a connection name for the log forwarder configuration

You can optionally activate/deactivate:

- forward logs for activity kmip
- forward logs for activity nae
- forward logs for client audit records
- forward logs for server audit records

Syntax for Syslog

```
ksctl log-forwarders add syslog --name <name of log forwarder> --connection-id <Syslog ConnectionID/Name> --forward-client-audit-records <true/false> --forward-logs-activity-kmip <true/false> --forward-logs-activity-nae <true/false> --forward-server-audit-records <true/false>]
```

Viewing Log Forwarders

You can use `ksctl log-forwarders get --id <log-forwarder-identifier>` to view details for a particular log forwarder.

You can use `ksctl log-forwarders list` to view details for all log forwarders.

The returned details include ID, name, type (Loki or Elasticsearch), CipherTrust Manager user account, hostname, port, ElasticSearch indices, and Loki labels.

Updating Elasticsearch Log Forwarder

For Elasticsearch log forwarder, you can modify:

- a name for the log forwarder configuration
- a connection ID of the Elasticsearch connection manager

- an index name for KMIP activity logs
- an index name for NAE activity logs
- an index name for server audit records
- an index name for server client records

Syntax for Updating Elasticsearch Log Forwarder

```
ksctl log-forwarders modify elasticsearch --id <LogForwarder ID/Name> --name <name of log forwarder> --connection-id <ES connectionID/Name> --index-activity-kmip <kmip_index_name> --index-activity-nae <nae_index_name> --index-server-audit-records <server_audit_records_index_name> --index-client-audit-records <client_audit_records_index_name>
```

Updating Loki Log Forwarder

For Loki log forwarder, you can modify:

- a connection name for the log forwarder configuration
- a connection ID of the Loki connection manager
- Labels field for KMIP activity logs
- Labels field for NAE activity logs
- Labels field for server audit records
- Labels field for client audit records

Syntax for Updating Loki Log Forwarder

```
ksctl log-forwarders modify loki --id <LogForwarder ID/Name> --name <name of log forwarder> --connection-id <Loki ConnectionID/Name> --labels-activity-kmip <kmip_label> --labels-activity-nae <nae_label> --labels-server-audit-records <server_audit_records_label> --labels-client-audit-records <client_audit_records_label>
```

Monitoring of Log Forwarder Connections

The CipherTrust Manager monitors the status of log forwarder connections at regular time intervals. If any connection is detected to be in an unhealthy state, an alarm is raised.

For the connections configured in the domain, the alarms are raised in that domain.

If an unhealthy log forwarder connection becomes healthy after some time, the alarm disappears for that connection. If there are multiple connection failures, the alarm will only disappear when all the connections will return to a healthy state.

When you try to clear the alarm, an error "Not Allowed to clear the log forwarder connections health check alarm" is returned.

For a cluster, alarms are node-specific because connections may fail at one node but not at another.

Alarm Name	Source	State	Triggered At	Cleared At	Cleared By	Acknowledged At	Acknowledged By	Account	Severity
ElasticSearch-Connections-Offline	ciphertrust	on	07 Mar 2024, 11:56 +07					kylo:kylo-e52061d9-7221-451b-b650-61ac5fa8e252:adminaccounts:kylo-e52061d9-7221-451b-b650-61ac5fa8e252	critical
Alarm Detail <pre> { "connections": [{ "id": "cd9f6691-0f68-4b0d-b1c3-c281de33960", "name": "es-conn1" }] } </pre> Alarm Description *Alarm triggers when one or more elasticsearch connections are down									
Loki-Connections-Offline	ciphertrust	on	07 Mar 2024, 11:56 +07					kylo:kylo-e52061d9-7221-451b-b650-61ac5fa8e252:adminaccounts:kylo-e52061d9-7221-451b-b650-61ac5fa8e252	critical
Alarm Detail <pre> { "connections": [{ "id": "e6d98318-0a61-4b01-b071-2a6d70a3f60", "name": "loki-conn1" }] } </pre> Alarm Description *Alarm triggers when one or more loki connections are down									
Syslog-Connections-Offline	ciphertrust	on	07 Mar 2024, 11:56 +07					kylo:kylo-e52061d9-7221-451b-b650-61ac5fa8e252:adminaccounts:kylo-e52061d9-7221-451b-b650-61ac5fa8e252	critical
Alarm Detail <pre> { "connections": [{ "id": "2d81d0d4-1efc-4bec-bc79-2a30e6c4c860", "name": "syslog-conn1" }] } </pre> Alarm Description *Alarm triggers when one or more syslog connections are down									

In the above screenshot, you can see that each alarm contains the `id` and `name` of the unhealthy log forwarder connection.

Note

To identify the error details, perform a test connection.

Let's consider a scenario where one more log forwarder connection "es-conn2" is added in the same domain where the "es-conn" connection already exists. Eventually, the new connection "es-conn2" also becomes unhealthy and the alarm is already raised for the existing unhealthy log forwarder connection "es-conn" as shown in the above screenshot.

In such scenarios, the alarm details will be updated only when the next cycle of monitoring completes as shown below:

Alarm Name	Source	State	Triggered At	Cleared At	Cleared By	Acknowledged At	Acknowledged By	Account	Severity
ElasticSearch-Connections-Offline	ciphertrust	on	07 Mar 2024, 11:56 +07					kylo:kylo-e52061d9-7221-451b-b650-61ac5fa8e252:adminaccounts:kylo-e52061d9-7221-451b-b650-61ac5fa8e252	critical
Alarm Detail <pre> { "connections": [{ "id": "cd9f6691-0f68-4b0d-b1c3-c281de33960", "name": "es-conn1" }, { "id": "2d81d0d4-1efc-4bec-bc79-2a30e6c4c860", "name": "es-conn2" }] } </pre> Alarm Description *Alarm triggers when one or more elasticsearch connections are down									

Updating Syslog Log Forwarder

For Syslog log forwarder, you can modify:

- a connection name for the log forwarder configuration
- a connection ID of the Syslog connection manager
- forward logs for KMIP activity logs
- forward logs for NAE activity logs
- forward logs for server audit records
- forward logs for client audit records

Syntax for Updating Syslog Log Forwarder

```
ksctl log-forwarders modify syslog --id <LogForwarder ID/Name> --name <name of log forwarder> --connection-id <Syslog ConnectionID/Name> --forward-client-audit-records <true/false> --forward-logs-activity-kmip <true/false> --forward-logs-activity-nae <true/false> --forward-server-audit-records <true/false>
```

Deleting a Log Forwarder

To delete a log forwarder, use `ksctl log-forwarders delete --id <log-forwarder-unique-identifier>`.

Prometheus Metrics Endpoint

You can use the Prometheus metrics endpoint to connect the [Prometheus monitoring system](#) to CipherTrust Manager. You can set Prometheus to scrape the CipherTrust Manager continuously, providing metrics over time to help monitor overall system health, performance, and cryptographic activity.

A sample configuration with Prometheus and Grafana docker images is available on [Github](#). The [Grafana data visualization application](#) provides graph visualizations of the Prometheus-collected metrics.

Prometheus and Grafana can only connect to the root [domain](#).

Prerequisites for Sample Configuration

- CipherTrust Manager 2.7.0 or later
- Docker
- Docker Compose (`docker-compose`)

Sample Configuration Setup

1. On your CipherTrust Manager, enable Prometheus metrics, either through a POST to the `/v1/system/metrics/prometheus/enable` endpoint, or with the `ksctl metrics prometheus enable` CLI command.

A token is returned, which Prometheus needs to scrape CipherTrust Manager.

Note

This token does not expire, but can be manually renewed with `ksctl metrics prometheus renew-token` or a POST to `/v1/system/metrics/prometheus/renew-token`.

2. Get the token which Prometheus needs to scrape CipherTrust Manager, if needed. You can use `GET` with the `/v1/system/metrics/prometheus/status` endpoint or `ksctl metrics prometheus status`.
3. In the Prometheus Metrics directory, edit the `prometheus.yml` file.

At minimum, you must provide the CipherTrust Manager hostname/IP in `targets` and the prometheus API token in `bearer token`. Prometheus can scrape multiple CipherTrust Managers, which might or might not share the API Token. This is an example configuration file with three CipherTrust Manager nodes, of which two share the same Prometheus API token:

```
scrape_configs:
  - job_name: "CipherTrust Manager"
    scheme: "https"
    tls_config:
      #ca_file: "/trusted_cas/web-keysecure-local.pem"
      #server_name: "web.keysecure.local"
      insecure_skip_verify: true
    bearer_token: "1zplR4njZsRN5dNeWAFXhkL1x7MU9q4H"
    metrics_path: "/api/v1/system/metrics/prometheus"
    static_configs:
      - targets:
        - "1.1.1.1"
        - "1.1.1.2"
  - job_name: "CipherTrust Manager Staging"
    scheme: "https"
    tls_config:
      #ca_file: "/trusted_cas/web-keysecure-local.pem"
      #server_name: "web.keysecure.local"
      insecure_skip_verify: true
    bearer_token: "TnRHpdL9v8MnWv8DhN9xuAaKgPevMEZs"
    metrics_path: "/api/v1/system/metrics/prometheus"
    static_configs:
      - targets:
        - "1.1.1.3"
```

4. Set up TLS authentication. By default, the Prometheus configuration sets `insecure_skip_verify: true` which is not recommended for production deployments as it skips SSL/TLS certificate validation for the CipherTrust Manager server.

- a. On CipherTrust Manager, download the certificate associated with the **web** interface. Export to a pem format.

```
ksctl interfaces certificate get --name web --icertfile <desired-filename>.pem
```

- b. Use openssl to retrieve the Common Name (CN) of the certificate, which will become the `server_name` value in Prometheus.

```
openssl x509 -noout -subject -in <your-file>.pem
```

Example response:

```
subject=C = US, ST = MD, L = Belcamp, O = Gemalto, CN = web.keysecure.local
```

The CN value, `web.keysecure.local`, is the value needed for Prometheus.

- c. Copy the certificate file to the `trusted_cas` folder in the Prometheus Metrics directory.
- d. Edit the `prometheus.yaml` file to include the `ca_file` path and `server_name` of the certificate, and disable the `insecure_skip_verify` parameter. For example:

```
scrape_configs:
  - job_name: "CipherTrust Manager"
    scheme: "https"
    tls_config:
      ca_file: "/trusted_cas/web-keysecure-local.pem"
      server_name: "web.keysecure.local"
      #insecure_skip_verify: true
    bearer_token: "TnRHpdL9v8MnWv8DhN9xuAaKgPevMEZs"
    metrics_path: "/api/v1/system/metrics/prometheus"
    static_configs:
      - targets:
        - "1.1.1.1"
```

5. In the Prometheus directory run `make up` to start the stack.

Note

You can run `make down` to stop the stack and `make clear` to stop the stack and all persisted data.

6. Visit the Prometheus Dashboard in a browser at `http://localhost:9090`.
 - a. Navigate to **Status > Target** to ensure that Prometheus is scraping CipherTrust Manager. The state should display as `UP` for each node, with no errors.
 - b. If you detect a problem, verify the metrics endpoint on CipherTrust Manager with `ksctl metrics prometheus get --api-token <api-token>`, or `curl -k 'https://<hostname>/api/v1/system/metrics/prometheus' -H 'Authorization: Bearer <api-token>' --compressed`). You can also use The Docker Compose logs to debug problem, with `docker-compose logs -f`.
7. Visit the Grafana Dashboard in a browser at `http://localhost:3000`.
 - a. Login with the user `admin` and the password `admin`. Set a new password when prompted.
 - b. Go to **Dashboards -> Home** to view the included dashboards.

Metrics Prefixes

The following high-level categories of metrics are returned from the endpoint:

Prefixes	Metrics Type	Prometheus Exporter, Package, or Integration
<code>ciphertrust_</code>	Metrics for specific CipherTrust resource applications. For example, size of the server audit log or key cache hits.	N/A
<code>dummy_</code>	Custom internal metrics that can be disregarded.	N/A
<code>docker_</code>	Metrics for the Docker containers that underlie CipherTrust Manager microservices, such as state, lifecycle, and resource usage.	<code>docker_exporter</code>
<code>go_</code>	Metrics gathered by the Go runtime. Most useful for debugging purposes with Thales engineers.	<code>go-metrics</code>
<code>http_</code>	Metrics about HTTP traffic to and from the CipherTrust Manager REST API endpoints and entities external to CipherTrust Manager. For example, response time and	N/A

Prefixes	Metrics Type	Prometheus Exporter, Package, or Integration
	number of requests to an endpoint.	
<code>httpclient_</code>	Metrics about HTTP traffic between internal CipherTrust Manager microservices. For example, response time and number of requests.	N/A
<code>node_</code>	Metrics for the CipherTrust Manager host, such as CPU and disk details.	Node exporter
<code>process_</code>	Metrics for microservices written in Go, including CPU, memory and file descriptor usage as well as the process start time.	Process collector of the go Prometheus package
<code>promhttp_</code>	Measures number of times individual microservices are called divided by HTTP code.	promhttp Go package
<code>sql_</code>	CipherTrust-specific metrics collected to analyze performance issues. Most useful for debugging purposes with Thales engineers.	N/A

Available Metrics Dashboards

The following dashboards are displayed in Grafana for CipherTrust Manager:

- **CipherTrust Manager Developer** - Metrics relevant to internal CipherTrust Manager developers to debug problems. This includes:
 - Average JWT processing time
 - Applications and Accounts Totals
 - Key Encryption Key (KEK)s Count
 - Authorization Policies Cache Hits pr Minute
 - Average Prometheus Metrics Scraping Response Time
- **CipherTrust Manager Host** - Metrics about the health of the CipherTrust Manager host, including CPU details, memory details, network details, network connections, and disk details.
- **CipherTrust Manager HTTP Traffic** - Metrics about HTTP traffic to the CipherTrust Manager. This includes:
 - Average HTTP Response Time Per Minute
 - HTTP Requests in the Last Minute

-
- Average Network Latency Per Minute
 - Average CM HTTP Client Response Time Per Minute
 - HTTP 500 Errors in the Last Minute
 - **CipherTrust Manager NAE** - Basic metrics about the performance of the **NAE-XML** cryptographic interface. This includes XML response time and XML processing time.
 - **CipherTrust Manager NAE Developer** - More detailed metrics about operations and performance on the NAE-XML interface, intended for debugging. This includes:
 - Key Info Cache Misses Time Per Minute
 - Key Info Cache Hits Time Per Minute
 - XML Total Processing time
 - XML Parsing Time
 - XML Transmit Time
 - XML Receive Time
 - XML Execution Time
 - **CipherTrust Manager Resources** - Metrics about creation and use of objects on CipherTrust Manager, such as audit records, keys, licenses, backup, and users.

Caution

Some of these metrics are in technical preview status. This means they are intended for evaluation in non-production environments. Details and functionality are subject to change. As well, it can take up to an hour for changes CipherTrust Manager resources to be reflected in the displayed technical preview metrics.

Metrics include:

- Audit Records Created Per Second Over The Last Minute
- Audit Records Created In Last Five Minutes
- Total Number of Audit Records
- Total Number of Keys By Algorithm
- Crypto Operations Per Second Over The Last Minute

-
- Total Number Of Connector Licenses Deployed
 - Number of License Units Consumed
 - License Unit Consumption by Percentage
 - Total Number Of Group Users in the System
 - Total Number Of Key Rotations
 - Key Rotations In Last Five Minutes
 - Time Taken To Create Backup
 - Number of Backups taken
 - Total Number of Keys by States and Algorithm. There a technical preview column indicating the domain account associated with the key count. Domain account information is available for the ten direct child domains with the most usage.
 - (Technical preview data) Total Number of Keys by Domains - This metric covers the ten direct child domains with the most usage.
 - (Technical preview data) Total Number of Orphaned Keys by Deleted Domain Accounts
 - (Technical preview data) Total Numbers of Subdomains by Domains - This metric covers the ten direct child domains with the most usage.
 - **CipherTrust Manager Services** - Metrics about the performance of individual microservices within CipherTrust Manager, intended for debug purposes. This includes:
 - CPU percentage
 - Memory usage
 - Network I/O (transmitting and receiving)
 - Disk I/O (reading and writing)
 - **CipherTrust Manager Node Metrics** - Metrics of the nodes in a clustered system showing the node connection information. This includes:
 - Write, flush, and replay lags
 - Sent, write, flush, and replay lag sizes
 - Whether replication is blocked
 - Whether node is connected
 - Connect time for a node
-

-
- Uptime for a connection
 - **CipherTrust Manager CTE Resources** - More detailed metrics about operations and performance on the CTE resources, intended for debugging. This includes:
 - CTE Clients
 - CTE Clients Health Status
 - CTE Groups
 - CTE Guard Points State
 - **CipherTrust Manager Secrets** - Metrics for the various secrets (Akeyless) gateway operations. This includes:
 - CPU utilization metrics
 - Disk I/O metrics
 - CPU load metrics
 - Memory utilization metrics
 - Network interface I/O metrics & TCP connection metrics
 - Current transaction number
 - Total transaction by an admin client
 - Total transaction limit per hour
 - Status of HTTP response
 - Total number of requests

For more details, click [here](#).

- **CipherTrust Manager Active TCP connections** - Metrics will show the number of active TCP connections across network interfaces and ports. They also monitor traffic distribution and identify load hot spots in the system.

Note

If you add a new network interface or modify an existing one and need to obtain the accurate TCP connection metrics, you can either wait for the network interfaces to refresh automatically—which occurs every five minutes—or [restart the system](#).

Policies

CipherTrust Manager authorizes operations using an [Attribute-Based Access Control \(ABAC\)](#) model. In ABAC, access control policies are defined by the attributes of the components involved in a request. These components typically include the user, the target resource (such as a cryptographic key), and the request's environment (which encompasses attributes like time of day, source IP address, geographic region, and incoming port).

Warning

Modifying policies, including the default ones, is an advanced operation. Incorrect changes can lead to system lockouts and render CipherTrust Manager unusable. It is strongly recommended that policy modifications are only performed in consultation with Thales Technical Support.

CipherTrust Manager ABAC system is inspired by [XACML](#) and [Amazon AWS policies](#).

An ABAC policy can use the attributes of any of these components to decide if the policy applies to a given request. For example, an ABAC policy could state: "Allow reading Key XYZ, but only at night, if the Key is 'enabled', the user belongs to the 'hr' department and is an admin, and the request originates from the internal port."

ABAC is advantageous as it encompasses other commonly used access control systems, such as Role-Based Access Control (RBAC). RBAC can be implemented using ABAC policies defined in terms of a user's "role" attribute.

CipherTrust Manager ABAC policies are expressed in terms of two resources: Policies and Policy Attachments.

Policy composition

A Policy is composed of:

- a set of actions (i.e. operations)
- a set of targets
- a rule, either "allow" or "deny"
- a set of conditions (optional)

Conditions are matching rules tested against the request to further refine the set of requests to which a Policy applies. For example, a simple policy might be: `allow read and write to key ABC and key XYZ`.

Conditions can further refine this policy. For example, the policy `allow read and write to key ABC and key XYZ` could be made more specific with the following cumulative conditions:

- The key is intended for document signing.
- The user belongs to the "Signers" group.
- The request originates from an internal IP address.

Policy attachments

Policies are not enforced on their own; they must be attached to a set of users. This is achieved by creating Policy Attachments.

A Policy Attachment links a single policy to a set of users. This set of users is defined by a matching rule (similar to policy conditions) that is tested against the user information in the API token. For example, policy attachments can match a single user, all users in a specific group, users who have a particular flag in their metadata, or all users. If the `principalSelector` field in the policy attachment JSON is an empty object (`{}`), the policy applies to all users.

For example, if the policy described earlier is named "Internal HR Key Usage," it could be enforced by attaching it to users in the HR department. The Policy Attachment would specify:

- Policy: Internal HR Key Usage
- Attached To: Users in the HR department

This might seem confusing because a policy's conditions can also refer to user attributes. For instance, the "Signers" group condition could be moved from the policy to the policy attachment with the same outcome. This overlap is intentional, providing flexibility in how policies are expressed and applied.

Default policies

On launch, CipherTrust Manager is pre-installed with a set of Policies and Policy Attachments, which define a default set of group based access controls. However, admins can add to, revise, or replace these Policies with their own, to implement almost any kind of access control regime.

Warning

As emphasized previously, modifying policies, including the default ones, is an advanced operation. Incorrect changes can lead to system lockouts and render CipherTrust Manager unusable. It is strongly recommended that policy modifications are only performed in consultation with Thales Technical Support.

The default policies enforce the following access control rules:

- The `admin` user has unrestricted access.
- Members of the `admin` group have unrestricted access.
- All users can create keys
- Users can perform any operation on keys they created (that is, they "own" the keys).
- Key owners can grant group permissions to their keys.
- Users in the `global` group can perform most operations on keys that have the `global` flag set.

Taken together, these policies implement an authorization model that mimics SafeNet KeySecure Classic:

-
- All Users can create keys, which they "own"
 - Users have admin privileges to the keys they own
 - Key owners can grant user groups limited usage privileges to their keys

Usage

Policies are powerful primitives. Misconfiguration can lead to locking out all access to CipherTrust Manager. Future enhancements may include pre-defined policy sets for common access control models (like RBAC) and safeguards against irreversible admin lockouts.

Managing login authentication for local users using policies

CipherTrust Manager allows you to create policies to manage local user authentication based on [interfaces](#) and [client applications](#). Policies are not enforced automatically; they must be assigned to users or groups via [policy attachment](#).

Interfaces

To create a policy based on interface attributes, use one of the following options for conditions:

- Interface name - specify the `path` as `context.environment.interface.name` and specify the required values in the `values` parameter, for example, "values": ["nae_all_123"].
- Interface port - specify the `path` as `context.environment.interface.port` and specify the required values in the `values` parameter, for example, "values": [1234].
- Interface type - specify the `path` as `context.environment.interface.type` and specify the required values in the `values` parameter, for example, "values": ["nae"]

The CipherTrust Manager supports the following values for Interface type:

- `nae` for NAE interface
- `kmip` for KMIP interface
- `web` for any interface other than NAE and KMIP (for example, UI, API playground, ksctl, direct API requests).

Example - Block users in a group from logging in to Web interface using policies

The `web` interface type covers login requests from the UI, API playground, ksctl, and direct API requests from external clients. To block a group of users from logging in via the `web` interface, follow these steps:

Create a policy

First, create a policy. In this example, the policy is named `Blocked Web Users`.

```
{
  "name": "Blocked Web Users",
  "effect": "deny",
  "resources": [],
  "actions": [
    "IssueJWT"
  ],
  "conditions": [
    {
      "op": "equals",
      "path": "context.environment.interface.type",
      "values": [
        "web"
      ]
    }
  ]
}
```

Create a policy attachment

Then, create a policy attachment to apply this policy to users in the "Blocked Web Users" group.

```
{
  "policy": "id-of-the-policy-from-previous-api",
  "principalSelector": {
    "cust": {
      "groups": ["Blocked Web Users"]
    }
  }
}
```

Client IP

Caution

- Exercise extreme caution when configuring IP address allowlists. An incorrect allowlist configuration can lock out all users, including administrators, rendering the entire cluster inaccessible and unmodifiable.
- The Admin Disaster Recovery feature cannot resolve this type of lockout. If this occurs, contact Thales Customer Support for assistance.
- When creating allowlisting policies, ensure you account for all client IPs that communicate with any node in the CipherTrust Manager cluster. Policies apply cluster-wide, not just to a specific node.

Note

Policies created to manage authentication based on the Client IP parameter don't apply to the requests coming to the NAE and KMIP interfaces with "anonymous login" mode enabled.

Client IP-based authentication policies do not apply to requests on NAE and KMIP interfaces if "anonymous login" mode is enabled for those interfaces.

CipherTrust Manager allows you to create policies that permit or deny local user authentication based on client IP addresses. Only users whose requests match these IP-based policies (and other authentication criteria) can obtain a JWT to access CipherTrust Manager.

When defining conditions for client IP policies, use the following parameters:

- `effect`: Set the value to "deny".
- `path`: Use `context.environment.client_ip` for IP-based conditions. For policies specific to certain ports, you'll typically combine an IP condition with an interface port condition (for example, `context.environment.interface.port` in a separate condition object).
- `values`: Specify the IPs to be allowed or blocked.
- `negate`: Set value to `true` to allow the IPs. Default value is `false`.

Note

To control access based on client IPs for JWT issuance:

- **To allowlist specific IPs (Permit only listed IPs)**: Set `"effect": "deny"` and `"negate": true` in the policy condition. This configuration denies access to all IPs *not* explicitly listed in the `values` array.
- **To denylist specific IPs (Block listed IPs)**: Set `"effect": "deny"` and `"negate": false` (or omit `negate`, as `false` is the default). This configuration denies access to all IPs explicitly listed in the `values` array.

In both scenarios, the `actions` array in the policy should include `"IssueJWT"`.

Create a policy to allow client IPs

To create a policy to allow client IPs:

Syntax

```
ksctl policy create --jsonfile <json-file>
```

Example Request

```
ksctl policy create --jsonfile allow-specific-ips-policy.json
```

allow-specific-ips-policy.json

```
{
  "name": "Allow Certain ip",
  "effect": "deny",
  "resources": [],
  "actions": [ "IssueJWT" ],
  "conditions": [ {
    "negate": true,
    "op": "equals",
    "path": "context.environment.client_ip",
    "values": [
      "192.168.1.5", "192.168.5.2"
    ]
  } ]
}
```

Example Response

```
{
  "id": "07d10481-7d88-4a48-ad2a-8dc2efd6b40f",
  "uri": "kylo:kylo:admin:policies:allow-certain-ip-07d10481-7d88-4a48-ad2a-8dc2efd6b40f",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-07-26T07:58:45.006746Z",
  "name": "Allow Certain ip",
  "deletedAt": null,
  "updatedAt": "2023-07-26T07:58:45.006746Z",
  "actions": [
    "IssueJWT"
  ],
  "resources": [],
  "allow": false,
  "effect": "deny",
  "conditions": [
    {
      "path": "context.environment.client_ip",
      "op": "equals",
      "values": [
        "192.168.1.5",
        "192.168.5.2"
      ],
      "negate": true
    }
  ]
}
```

Attaching the Policy

To enforce the policy, attach it to the desired principals. Use the `id` returned when the policy was created. An empty `principalSelector` (`{}`) in the attachment JSON applies the policy to all users.

Syntax

```
ksctl polattach create --polid <policy-id> --prinjson <principal-selector-json-file>
```

Example Request

```
ksctl polattach create --polid 07d10481-7d88-4a48-ad2a-8dc2efd6b40f --prinjson attachment.json
```

attachment.json

```
{}
```

Example Response

```
{
  "id": "4deb1d53-31d7-452a-8b55-0400d502e8ee",
  "uri": "kylo:kylo:admin:policy-attachments:4deb1d53-31d7-452a-8b55-0400d502e8ee",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-07-26T07:59:02.935252Z",
  "updatedAt": "2023-07-26T07:59:02.935252Z",
  "policy": "kylo:kylo:admin:policies:allow-certain-ip-07d10481-7d88-4a48-ad2a-8dc2efd6b40f",
  "jurisdiction": "kylo:kylo:admin:accounts:kylo",
  "principalSelector": {},
  "includeExternalPrincipals": false,
  "name": "Allow Certain ip",
  "actions": [
    "IssueJWT"
  ],
  "resources": [],
  "allow": false,
  "effect": "deny",
  "conditions": [
    {
      "path": "context.environment.client_ip",
      "op": "equals",
      "values": [
        "192.168.1.5",
        "192.168.5.2"
      ],
      "negate": true
    }
  ]
}
```

Create a policy to block client IPs

To create a policy to block client IPs:

Syntax

```
ksctl policy create --jsonfile <json-file>
```

Example Request

```
ksctl policy create --jsonfile block-example.json
```

block-example.json

```
{
  "name": "Block ips",
  "effect": "deny",
  "resources": [],
  "actions": [
    "IssueJWT"
  ],
  "conditions": [
    {
      "op": "equals",
      "path": "context.environment.client_ip",
      "values": [
        "192.168.1.5",
        "192.168.5.2"
      ]
    }
  ]
}
```

Example Response

```
{
  "id": "bbc6e24a-fdae-44c6-bd1d-dacb1a3ef874",
  "uri": "kylo:kylo:admin:policies:block-aman-bbc6e24a-fdae-44c6-bd1d-dacb1a3ef874",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-07-25T09:46:01.863493Z",
  "name": "Block ips",
  "deletedAt": null,
  "updatedAt": "2023-07-25T09:46:01.863493Z",
  "actions": [
    "IssueJWT"
  ],
  "resources": [],
  "allow": false,
  "effect": "deny",
  "conditions": [
    {

```

```

        "path": "context.environment.client_ip",
        "op": "equals",
        "values": [
            "192.168.1.5",
            "192.168.5.2"
        ]
    }
]
}

```

Attaching the Policy

To enforce the policy, attach it. Use the `id` from the policy creation response.

Syntax

```
ksctl polattach create --polid <policy-id> --prinjson <principal-selector-json-file>
```

Example Request

```
ksctl polattach create --polid bbc6e24a-fdae-44c6-bd1d-dacbla3ef874 --prinjson block-attachment.json
```

block-attachment.json

```
{}
```

Example Response

```

{
  "id": "5e188e61-78c9-4711-851f-025363214cbc",
  "uri": "kylo:kylo:admin:policy-attachments:5e188e61-78c9-4711-851f-025363214cbc",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-07-25T09:47:22.751751Z",
  "updatedAt": "2023-07-25T09:47:22.751751Z",
  "policy": "kylo:kylo:admin:policies:block-aman-bbc6e24a-fdae-44c6-bd1d-dacbla3ef874",
  "jurisdiction": "kylo:kylo:admin:accounts:kylo",
  "principalSelector": {},
  "includeExternalPrincipals": false,
  "name": "Block ip",
  "actions": [
    "IssueJWT"
  ],
  "resources": [],
  "allow": false,
  "effect": "deny",
  "conditions": [

```

```
{
  "path": "context.environment.client_ip",
  "op": "equals",
  "values": [
    "192.168.1.5",
    "192.168.5.2"
  ]
}
```

Create a policy to allow client IPs of a subnet

To create a policy to allow client IPs of a specific subnet:

Syntax

```
ksctl policy create --jsonfile <json-file>
```

Example Request

```
ksctl policy create --jsonfile allow-subnet-policy.json
```

allow-subnet-policy.json

```
{
  "name": "Allow IPs of a Subnet",
  "effect": "deny",
  "resources": [],
  "actions": [ "IssueJWT" ],
  "conditions": [ {
    "negate": true,
    "op": "regex",
    "path": "context.environment.client_ip",
    "values": [
      "165.225.*"
    ]
  } ]
}
```

Example Response

```
{
  "id": "367969df-e591-4497-a7e6-55ecfa3e9c08",
  "uri": "kylo:kylo:admin:policies:allow-certain-ip-367969df-e591-4497-a7e6-55ecfa3e9c08",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-11-17T09:11:56.054262Z",
  "name": "Allow IPs of a Subnet",
  "deletedAt": null,
}
```

```

    "updatedAt": "2023-11-17T09:11:56.05426Z",
    "actions": [
      "IssueJWT"
    ],
    "resources": [],
    "allow": false,
    "effect": "deny",
    "conditions": [
      {
        "path": "context.environment.client_ip",
        "op": "regex",
        "values": [
          "165.225.*"
        ],
        "negate": true
      }
    ]
  }
}

```

In the above example, all the IPs of the subnet `165.225.0.0/16` will be allowed.

Attaching the Policy

To enforce the policy, attach it. Use the `id` from the policy creation response.

Syntax

```
ksctl polattach create --polid <policy-id> --prinjson <principal-selector-json-file>
```

Example Request

```
ksctl polattach create --polid 367969df-e591-4497-a7e6-55ecfa3e9c08 --prinjson attachment.json
```

attachment.json

```
{}
```

Create a policy to block client IPs of a subnet

To create a policy to block client IPs of a specific subnet:

Syntax

```
ksctl policy create --jsonfile <json-file>
```

Example Request

```
ksctl policy create --jsonfile block-subnet-policy.json
```

block-subnet-policy.json

```
{
  "name": "Block IPs of a Subnet",
  "effect": "deny",
  "resources": [],
  "actions": [ "IssueJWT" ],
  "conditions": [ {
    "op": "regex",
    "path": "context.environment.client_ip",
    "values": [
      "165.225.*"
    ]
  } ]
}
```

Example Response

```
{
  "id": "21c8cfe4-cba2-436a-85fc-2df7b22afbd7",
  "uri": "kylo:kylo:admin:policies:block-certain-ip-21c8cfe4-cba2-436a-85fc-2df7b22afbd7",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-11-17T09:12:24.23122Z",
  "name": "Block IPs of a Subnet",
  "deletedAt": null,
  "updatedAt": "2023-11-17T09:12:24.23122Z",
  "actions": [
    "IssueJWT"
  ],
  "resources": [],
  "allow": false,
  "effect": "deny",
  "conditions": [
    {
      "path": "context.environment.client_ip",
      "op": "regex",
      "values": [
        "165.225.*"
      ]
    }
  ]
}
```

In the above example, all the IPs of the subnet `165.225.0.0/16` will be blocked.

Attaching the Policy

To enforce the policy, attach it. Use the `id` from the policy creation response.

Syntax

```
ksctl polattach create --polid <policy-id> --prinjson <principal-selector-json-fi
```



```
le>
```

Example Request

```
ksctl polattach create --polid 21c8cfe4-cba2-436a-85fc-2df7b22afbd7 --prinjson block-attachment.json
```

block-attachment.json

```
{}
```

Create a policy to allow specific client IPs for ports

To create a policy to allow specific client IPs for ports:

Syntax

```
ksctl policy create --jsonfile <json-file>
```

Example Request

```
ksctl policy create --jsonfile allow-ip-for-ports-policy.json
```

allow-ip-for-ports-policy.json

```
{
  "name": "Allow ip for a specific port",
  "effect": "deny",
  "resources": [],
  "actions": [ "IssueJWT" ],
  "conditions": [ {
    "op": "equals",
    "path": "context.environment.interface.port",
    "values": [
      "9001", "9902", "9903"
    ]
  },
  {
    "negate" : true,
    "op": "equals",
    "path": "context.environment.client_ip",
    "values": [
      "122.185.33.186"
    ]
  } ]
}
```

Example Response

```
{
```

```

    "id": "282b2e01-02f7-429a-aaa6-a7e9af939dee",
    "uri": "kylo:kylo:admin:policies:allow-certain-ip2-282b2e01-02f7-429a-aaa6-a7e9af939dee",
    "account": "kylo:kylo:admin:accounts:kylo",
    "application": "ncryptify:gemalto:admin:apps:kylo",
    "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
    "createdAt": "2023-11-17T09:08:46.127197Z",
    "name": "Allow ip for a specific port",
    "deletedAt": null,
    "updatedAt": "2023-11-17T09:08:46.127197Z",
    "actions": [
      "IssueJWT"
    ],
    "resources": [],
    "allow": false,
    "effect": "deny",
    "conditions": [
      {
        "op": "equals",
        "path": "context.environment.interface.port",
        "values": [
          "9001", "9902", "9903"
        ]
      },
      {
        "path": "context.environment.client_ip",
        "op": "equals",
        "values": [
          "122.185.33.186"
        ],
        "negate": true
      }
    ]
  }
}

```

In the above example, only 122.185.33.186 IP will be allowed for ports "9001, 9902, and 9903".

Attaching the Policy

To enforce the policy, attach it. Use the `id` from the policy creation response.

Syntax

```
ksctl polattach create --polid <policy-id> --prinjson <principal-selector-json-file>
```

Example Request

```
ksctl polattach create --polid 282b2e01-02f7-429a-aaa6-a7e9af939dee --prinjson attachment.json
```

attachment.json

```
{ }
```

Create a policy to block specific client IPs for ports

To create a policy to block specific client IPs for ports:

Syntax

```
ksctl policy create --jsonfile <json-file>
```

Example Request

```
ksctl policy create --jsonfile block-ip-for-ports-policy.json
```

block-ip-for-ports-policy.json

```
{
  "name": "Block ips for a specific port",
  "effect": "deny",
  "resources": [],
  "actions": [ "IssueJWT" ],
  "conditions": [ {
    "op": "equals",
    "path": "context.environment.interface.port",
    "values": [
      "9001", "9902", "9903"
    ]
  },
  {
    "op": "equals",
    "path": "context.environment.client_ip",
    "values": [
      "122.185.33.186"
    ]
  }
  ]
}
```

Example Response

```
{
  "id": "ae705635-056a-47a4-801c-0bdf8681b74a",
  "uri": "kylo:kylo:admin:policies:block-certain-ip2-ae705635-056a-47a4-801c-0bdf8681b74a",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-11-17T07:43:35.614821Z",
  "name": "Block ips for a specific port",
  "deletedAt": null,
  "updatedAt": "2023-11-17T07:43:35.614821Z",
  "actions": [
    "IssueJWT"
  ],
  "resources": [],
  "allow": false,
  "effect": "deny",
}
```

```

    "conditions": [ {
      "op": "equals",
      "path": "context.environment.interface.port",
      "values": [
        "9001", "9902", "9903"
      ]
    },
    {
      "path": "context.environment.client_ip",
      "op": "equals",
      "values": [
        "122.185.33.186"
      ]
    }
  ]
}

```

In the above example, only 122.185.33.186 IP will be blocked for "9001, 9902, and 9903" ports.

Attaching the Policy

To enforce the policy, attach it. Use the `id` from the policy creation response.

Syntax

```
ksctl polattach create --polid <policy-id> --prinjson <principal-selector-json-file>
```

Example Request

```
ksctl polattach create --polid ae705635-056a-47a4-801c-0bdf8681b74a --prinjson attachment.json
```

attachment.json

```
{ }
```

Client applications

For policies based on client applications, use the `path` as `context.environment.principal.client_app`. Specify the application types in the `values` parameter (for example, `"values": ["nae"]`).

- nae
- kmip

Example - Login to nae and kmip interfaces using policies

To allow login to nae and kmip interfaces irrespective of the authentication methods set for the user, perform the following steps:

Create a policy

```
{
  "name": "Allow Login to NAE/KMIP",
  "effect": "allow",
  "resources": [],
  "actions": [
    "IssueJWT"
  ],
  "conditions": [
    {
      "op": "equals",
      "path": "context.environment.principal.client_app",
      "values": [
        "nae",
        "kmip"
      ]
    }
  ]
}
```

Create a policy attachment

```
{
  "policy": "id-of-the-policy-from-previous-api",
  "principalSelector": {}
}
```

Control operations on Web interface ports

You can control (allow or deny) operations on Web interfaces by creating custom policies.

Example policy: Blocking CreateKey operation from web port.

```
{
  "name": "Blocked CreateKey operation from web port",
  "effect": "deny",
  "resources": [],
  "actions": [
    "CreateKey"
  ],
  "conditions": [
    {
      "op": "equals",
      "path": "context.environment.interface.port",
      "values": [
        443
      ]
    }
  ]
}
```

The above policy blocks CreateKey operation on port 443 of custom Web interface.

Banners

There are two kind of banners, the pre-authentication banner is shown before the user logs in to the CipherTrust Manager server, and the post-authentication banner is shown after login. Both are optional and not set by default.

The pre-authentication banner is useful for displaying disclaimer or legal security policy. You must accept the terms identified in this banner in order to login. Not accepting the terms will prevent you from logging in to the CipherTrust Manager server.

The post-authentication banner is useful for displaying information about the system. This banner message can be used to provide general information about the server to the user such as upcoming maintenance. The user is not asked to accept this banner message.

You can view and update the banner pages using the CLI, GUI and API.

Banners support the unicode standard and the Markdown markup language.

Viewing the Banners using the ksctl CLI

Using the CLI, any authenticated user in any domain can view the pre- and post-authentication banner by using the `banners get` command:

Example `get` commands:

```
$ ksctl banners get --name pre-auth
```

or

```
$ ksctl banners get --name post-auth
```

Default command response before updating the banner message, for each command is:

```
{
  "value": ""
}
```

Example response after updating the pre- or post authentication banner:

```
{
  "value": "Example text of the banner."
}
```

Updating the Banners using the ksctl CLI

This action requires you to login to the root `domain` as a user in the `admingroup`, such as `admin`.

You can update the pre- and post-authentication banners using the `banners update` command:

There are two ways to provide a new banner message:

- using the argument `--message <banner-message>` where `<banner-message>` is new banner message
- using the argument `--file <banner-message-file>`, where `<banner-message-file>` is a path and filename in Markdown notation containing the new banner message

Example: `/path/banner-message.md`

Pre-auth Example `update --message` command:

```
$ ksctl banners update --name pre-auth --message "Before logging in to CipherTrust Manager, you must accept the following 'Terms of Use': ..."
```

Response:

```
{
  "value": "Before logging in to CipherTrust Manager, you must accept the following 'Terms of Use': ..."
}
```

Post-auth Example `update --message` command:

```
$ ksctl banners update --name post-auth --message "Welcome to **CipherTrust k170v Server**."
```

Response:

```
{
  "value": "Welcome to **CipherTrust k170v Server**."
}
```

Pre-auth Example `update --file` command:

```
$ ksctl banners update --name pre-auth --file </path/banner-message.md>
```

If the markdown file contains: "Before logging in to CipherTrust Manager, you must accept the following 'Terms of Use': ...", the response is:

```
{
  "value": "Before logging in to CipherTrust Manager, you must accept the following 'Terms of Use': ..."
}
```

Post-auth Example `update --file` command:

```
$ ksctl banners update --name post-auth --file </path/banner-message.md>
```

If the markdown file contains: "Welcome to **CipherTrust k170v Server**.", the response is:

```
{
  "value": "Welcome to **CipherTrust k170v Server**."
}
```

Updating the Banners Using the CipherTrust Web UI

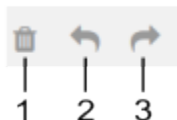
1. Log in to the root domain as a user in the `admin` group, such as `admin`.
2. Navigate to **Admin Settings > Login Banners**.
3. Select the **Pre Authentication** or **Post Authentication** depending on the desired banner type.
4. In the left text box, type the banner message.
5. Format the message as desired. You can use markdown syntax directly, or the editing icons to format your message. The right text box renders the formatted message, allowing you to preview the banner appearance.

The formatting icons are as follows:



- 1 - Insert a heading.
- 2 - Format the selected text in bold.
- 3 - Format the selected text in italics.
- 4 - Format the selected text in underline.
- 5 - Format the selected text in strikethrough.
- 6 - Insert a bulleted list.
- 7 - Insert a numbered list.
- 8 - Format the selected text as a blockquote.
- 9 - Insert a horizontal rule.
- 10 - Format the selected text formatted as inline code.
- 11 - Format the selected text as a codeblock.
- 12 - Insert a table.
- 13 - Format the selected text as a hyperlink.

1. If necessary, use the delete all (1), undo (2), and redo (3) icons.



2. Click **Save** to save your changes.

Basic Interface Configuration

Interfaces are services the CipherTrust Manager hosts. Most interfaces are listening on a particular port, but may also represent other input channels, like local shell access or serial port access. To view the Interfaces page in the UI, go to **Admin Settings > Interfaces**.

The CipherTrust Manager currently supports six interfaces:

- web: The HTTP server on port 80 and 443. This interface serves both the GUI and the REST API.
- ssh: The SSH server on port 22.
- nae: The NAE-XML server on port 9000.
- kmip: The KMIP server on port 5696.
- SNMP: The SNMP Agent on port 161.
- preboot: Used for ksctl CLI or REST API communication when the **disk is encrypted** and locked. Only present if disk encryption has been enabled.

Note

- The Web, KMIP, NAE, and preboot interfaces have several options to control authentication to those interfaces. The [Authentication Settings for NAE, KMIP, Web, and Preboot interfaces](#) page describes these settings in detail.
- Viewing the preboot interface and [managing its certificate](#) are the only operations available for that interface.
- The CTE clients must not be registered or used with [custom web interfaces](#). However, CTE clients will continue to function properly when used with the default web interface.

Effect of updating interface settings

Warning

If you have active NAE, KMIP, or web connections, we recommend that you plan for connection downtime before updating interface settings, especially for updates to the initial default interfaces.

Note

- When you update the port of the **default interface** of NAE, KMIP, and WEB, all nodes in the cluster restart. However, if you update any other setting apart from the port, no node in the cluster restarts.
- If you update any setting on the **non-default interfaces** of NAE, KMIP, and WEB, no node in the cluster restarts.

There are some interface changes that are applied immediately and trigger an automatic services restart. This restart can disrupt running NAE, KMIP, and web connections and cause an immediate downtime of a few minutes. If your CipherTrust Manager is part of a [cluster](#), the interface settings change can also replicate to other nodes, and disrupt running NAE, KMIP, and web connections to those nodes. Therefore, plan for some downtime before updating interfaces.

Interface settings changes that are known to cause an immediate loss of connection are:

- [Updating the port](#) for NAE, KMIP, or web interface
- Enabling the **Hard Delete** option for the KMIP interface

Other interface changes require a manual [Services Restart](#).

View the initial set of interfaces using ksctl

```
ksctl interfaces list
```

Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 4,
  "resources": [
    {
      "id": "ee242373-2555-48c7-923b-86ed3e785504",
      "name": "kmip",
```

```

    "mode": "tls-cert-pw-opt",
    "cert_user_field": "CN",
    "auto_gen_ca_id": "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acff
f84a",
    "trusted_cas": {
      "local": [
        "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acfff84a"
      ]
    },
    "createdAt": "2020-07-20T12:59:14.776939Z",
    "updatedAt": "2020-07-20T12:59:24.703806Z",
    "default_connection": "local_account",
    "port": 5696,
    "network_interface": "all",
    "interface_type": "kmip",
    "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
    },
    "enabled": true
  },
  {
    "id": "a4db15b4-64fc-40d3-8465-9935866bbe09",
    "name": "nae",
    "mode": "no-tls-pw-req",
    "cert_user_field": "CN",
    "auto_gen_ca_id": "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acff
f84a",
    "trusted_cas": {
      "local": [
        "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acfff84a"
      ]
    },
    "createdAt": "2020-07-20T12:59:14.770938Z",
    "updatedAt": "2020-07-20T16:21:43.427252Z",
    "default_connection": "local_account",
    "port": 9000,
    "network_interface": "all",
    "interface_type": "nae",
    "local_auto_gen_attributes": {
      "cn": "nae.keysecure.local",
      "email_addresses": [
        "support@gemalto.com"
      ],
      "names": [
        {
          "C": "US",

```

```

        "ST": "MD",
        "L": "Belcamp",
        "O": "Gemalto",
        "OU": ""
    }
],
    "generated": false
},
    "enabled": true
},
{
    "id": "c8c6e4c1-30d7-4317-9d37-a9ec217ffb17",
    "name": "ssh",
    "trusted_cas": {},
    "createdAt": "2020-07-20T12:59:14.7787Z",
    "updatedAt": "2020-07-20T12:59:14.7787Z",
    "port": 22,
    "network_interface": "all",
    "interface_type": "ssh",
    "local_auto_gen_attributes": {
        "cn": "ssh.keysecure.local",
        "email_addresses": [
            "support@gemalto.com"
        ],
        "names": [
            {
                "C": "US",
                "ST": "MD",
                "L": "Belcamp",
                "O": "Gemalto",
                "OU": ""
            }
        ],
        "generated": false
    },
    "enabled": true
},
{
    "id": "e64d84b0-e952-4fa3-83c9-e1d1bc52a996",
    "name": "web",
    "mode": "tls-cert-opt-pw-opt",
    "cert_user_field": "CN",
    "auto_gen_ca_id": "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acff
f84a",
    "trusted_cas": {
        "local": [
            "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acfff84a"
        ]
    },
    "createdAt": "2020-07-20T12:59:14.774618Z",
    "updatedAt": "2020-07-20T12:59:23.37516Z",
    "port": 443,
    "network_interface": "all",
    "interface_type": "web",
    "local_auto_gen_attributes": {
        "cn": "web.keysecure.local",
        "email_addresses": [
            "support@gemalto.com"
        ],
    },

```

```

    "names": [
      {
        "C": "US",
        "ST": "MD",
        "L": "Belcamp",
        "O": "Gemalto",
        "OU": ""
      }
    ],
    "generated": false
  },
  "enabled": true
}
]
}

```

Updating interface ports

You can change ports for the following interfaces:

- WEB
- NAE
- KMIP

Syntax

```
ksctl interfaces modify --name <interface-name> --port <port-number>
```

Note

- You can find <Interface-Name> using the command `ksctl interfaces list`.
- Port 9100 is used for an internal service and is not available for CipherTrust Manager interfaces.

Example 1: Changing the default WEB interface port to 8443

```
ksctl interfaces modify --name web --port 8443
```

Response

```

{
  "id": "f1af6f94-43af-4350-84d0-ec6b08639e5b",
  "name": "web",
  "mode": "tls-cert-opt-pw-opt",

```

```

    "cert_user_field": "CN",
    "auto_gen_ca_id": "kylo:kylo:naboo:localca:a9319f59-5914-41a2-886e-32b8930d08
2c",
    "trusted_cas": {
      "local": [
        "kylo:kylo:naboo:localca:a9319f59-5914-41a2-886e-32b8930d082c"
      ]
    },
    "createdAt": "2020-12-09T14:25:14.596482Z",
    "updatedAt": "2020-12-09T14:29:50.246509Z",
    "port": 8443,
    "network_interface": "all",
    "interface_type": "web",
    "local_auto_gen_attributes": {
      "cn": "web.keysecure.local",
      "email_addresses": [
        "support@gemalto.com"
      ],
      "names": [
        {
          "C": "US",
          "ST": "MD",
          "L": "Belcamp",
          "O": "Gemalto",
          "OU": ""
        }
      ],
      "generated": false
    },
    "enabled": true
  }
}

```

Example 2: Changing the default NAE interface port to 443

```
ksctl interfaces modify --name nae --port 443
```

Response

```

{
  "id": "2228f2aa-f973-4fda-b633-ead376db3e19",
  "name": "nae",
  "mode": "unauth-tls-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:a9319f59-5914-41a2-886e-32b8930d08
2c",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:a9319f59-5914-41a2-886e-32b8930d082c"
    ]
  },
  "createdAt": "2020-12-09T14:25:14.594319Z",
  "updatedAt": "2020-12-09T14:35:22.75339Z",
  "default_connection": "local_account",
  "port": 443,
}

```

```

"network_interface": "all",
"interface_type": "nae",
"minimum_tls_version": "tls_1_2",
"local_auto_gen_attributes": {
  "cn": "nae.keysecure.local",
  "email_addresses": [
    "support@gemalto.com"
  ],
  "names": [
    {
      "C": "US",
      "ST": "MD",
      "L": "Belcamp",
      "O": "Gemalto",
      "OU": ""
    }
  ],
  "generated": false
},
"enabled": true
}

```

Enabling/disabling interfaces

You can enable or disable the following interfaces:

- **SSH**
- **NAE**
- **KMIP**
- **Custom Web**

To enable/disable an interface using the GUI, click **Action Button > Enable/Disable**.

Note

After the user has enabled/disabled an interface, it remains in same state even after restarting the device and it will get replicated if the device is part of a cluster.
After an interface has been disabled, the CipherTrust Manager drops all incoming and existing connections on that interface.

Enable or disable an interface using ksctl

When using ksctl:

- To enable an interface: `ksctl interfaces enable --name <interface-name>`
- To disable an interface: `ksctl interfaces disable --name <interface-name>`

Note

Only the SSH, KMIP, and NAE interfaces can be enabled or disabled. Replace <interface-name> in the above commands with `ssh`, `kmip`, or `nae` for these interfaces.

Example: Disabling the SSH interface

```
ksctl interfaces disable --name ssh
```

Response

```
{
  "id": "2e8d2344-c40b-466c-8202-d05d2cb6738a",
  "name": "ssh",
  "trusted_cas": {},
  "createdAt": "2020-08-13T10:22:32.792266Z",
  "updatedAt": "2020-08-14T11:11:28.564276Z",
  "port": 22,
  "network_interface": "all",
  "interface_type": "ssh",
  "local_auto_gen_attributes": {
    "cn": "ssh.keysecure.local",
    "email_addresses": [
      "support@gemalto.com"
    ],
    "names": [
      {
        "C": "US",
        "ST": "MD",
        "L": "Belcamp",
        "O": "Gemalto",
        "OU": ""
      }
    ],
    "generated": false
  },
  "enabled": false
}
```

Adding and removing interfaces (NAE, KMIP, and SNMP)

You can add and remove interfaces other than default interfaces (NAE, KMIP, and WEB). Currently, the Create and Delete commands are supported only for the NAE, KMIP, SNMP and custom Web interfaces.

Adding custom Web interface

Note

Only CCKM supports custom web interface.

1. Navigate to **Admin Settings > Interfaces**
2. Select **+ Add Interface**.
3. Select **Web** and click **Next**.
4. In **Network Form**, provide the following values.
 - **Port:** A port for the Web interface to listen on. 8080 is the default recommended port.
 - A **Network Interface:** The CipherTrust Manager network interface to communicate on.
 - An **Interface Mode**.
 - **Allowed Identity Types:** Select `User`, `Client`, or both. JWTs are issued accordingly to the user, client, or both.

Note

Ensure that a `user` is selected in at least one web interface. Otherwise, you won't be able to log into CipherTrust Manager using web port.

5. Click **Next**.
6. In **Add Certificates**,
 - a. Select **Username Location in Certificate**.
 - b. Select the **Auto Generation** checkbox to generate server certificate automatically.
 - a. Select **Local CA for Automatic Server Certificate Generation**.
 - b. (Optional) Select **Number of Days before expiry to Auto Gen Cert**.
 - c. Add **Local Trusted CAs** and **External Trusted CAs**.
7. Click **Save**.

Note

You can also control operations on custom web interfaces using custom policies. See [Control operations on Web interface ports](#) for more details.

Adding the SNMP interface

To configure [SNMP](#) agent, you must add the SNMP interface.

Adding the SNMP interface in the GUI

1. Navigate to **Admin Settings > Interfaces**
2. Select **+ Add Interface**.
3. Select **SNMP** and click **Next**.
4. Provide the following values:
 - A friendly name.
 - A port for the SNMP interface to listen on. 161 is the default recommended port.
 - A network interface.
5. Select **Save**.

Adding the SNMP interface using ksctl

The following command adds the SNMP interface:

```
ksctl interfaces create --type snmp --name <friendly_name_for_the_interface> --port <port_to_listen_on> --network-interface <network_interface_on_host>
```

The default, recommended port is 161. Valid network interfaces are `all` or a particular network interface name such as `ens32`. You can use `ksctl network interfaces list` to view available network interfaces.

Creating a new NAE interface

Example

```
ksctl interfaces create -o 9009 -y nae
```

Response

```
{
  "id": "456eb374-ec5c-40e8-bc89-4ab485c20c6c",
  "name": "nae_all_9009",
  "mode": "unauth-tls-pw-opt",
  "cert_user_field": "CN",
  "auto_gen_ca_id": "kylo:kylo:naboo:localca:61476734-4778-40ec-a3be-06654d123513",
  "trusted_cas": {
    "local": [
      "kylo:kylo:naboo:localca:61476734-4778-40ec-a3be-06654d123513"
    ],
    "external": []
  },
  "createdAt": "2019-01-21T05:52:57.657447Z",
  "updatedAt": "2019-01-21T05:52:57.657447Z",
  "default_connection": "local_account",
  "custom_uid_size": 0,
  "port": 9009,
  "network_interface": "all",
  "interface_type": "nae"
}
```

Creating 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"
}
```

Deleting an interface

You can delete an NAE, KMIP, or SNMP interface.

Delete all [SNMP configurations](#), including communities, users, and management stations, before deleting the SNMP interface.

Example

```
ksctl interfaces delete -n nae_all_9009
```

Support for Post-quantum cryptography (PQC)

Post-quantum cryptography (PQC) algorithms are intended to be secure against threats posed by quantum computers, which may be capable of compromising existing encryption techniques.

TLS Key Exchange groups for Web interface

TLS Key Exchange Groups include a list of the key exchange algorithms that the system offers for the WEB interface during the TLS handshake. By default, the following classic key exchange algorithms are enabled: 'x25519', 'secp256r1', 'x448', 'secp521r1', 'secp384r1', 'ffdhe2048', 'ffdhe3072', 'ffdhe4096', 'ffdhe6144', and 'ffdhe8192'. The post-quantum key exchange algorithms such as 'X25519MLKEM768', 'SecP256r1MLKEM768', 'MLKEM1024', 'MLKEM768', and 'MLKEM512' are disabled.

The behavior regarding the sequence of TLS groups for the specified TLS versions is described below:

- For TLS 1.3, regardless of the server's configured preference order for the groups, the server will automatically accept the first group specified in the key exchange groups list that the client includes in its "ClientHello" message, provided that it is also supported by the server.
- For TLS v1.2, the client cannot present its key exchange groups list in the "ClientHello" message beforehand; consequently, the server will request that the client aligns itself with the server's most preferred group that the client also supports. This behavior is in accordance with OpenSSL.

However, the order of the groups is still significant. If the server chooses a less preferred, yet supported group, it sends a supported groups list back to the client in the "ServerHello" message. This allows the client to remember the server's preference and adjust its key exchange groups list accordingly for the subsequent requests.

Caution

Exercise caution when disabling any EC curve. For example, if a user disables all EC TLS groups (x25519, secp256r1, and secp384r1) except for secp521r1, the Chrome browser does not support the secp521r1 TLS group, which can prevent CipherTrust Manager from loading in Chrome. However, it still functions in Firefox.

The clients available for public testing of post-quantum key exchange TLS groups:

Sample command for clients

OpenSSL 3.5 version <https://github.com/openssl/openssl/releases/download/openssl-3.5.0/openssl-3.5.0.tar.gz>

```
openssl s_client -connect CM_IP:443 -groups X25519MLKEM768 -tls1_3
```

<https://hub.docker.com/r/openquantumsafe/curl>

```
docker run -it oqs-curl curl --curves x25519 -k -v 'https://ip/api/v1/vault/keys
2/?skip=0&limit=10' -H 'Authorization: Bearer token' -H 'accept: application/jso
n'
```

<https://hub.docker.com/r/openquantumsafe/oqs-openssl3>

```
docker run -it openquantumsafe/oqs-openssl3 openssl s_client -connect CM_IP:443 --g
roups mlkem768 --curves X25519MLKEM768 -tls1_3 -provider oqsprovider
```

Example

To specify TLS groups:

```
ksctl modify interfaces --tls-groups-file tls-group.json --tls-ciphers-file tls-c
iphers.json --name web
```

tls-group.json

```
{
  "group_name": "x25519", "enabled": true,
  {"group_name": "secp256r1", "enabled": true},
  {"group_name": "x448", "enabled": true},
  {"group_name": "secp521r1", "enabled": true},
  {"group_name": "secp384r1", "enabled": true},
  {"group_name": "ffdhe2048", "enabled": true},
  {"group_name": "ffdhe3072", "enabled": true},
  {"group_name": "ffdhe4096", "enabled": true},
  {"group_name": "ffdhe6144", "enabled": true},
  {"group_name": "ffdhe8192", "enabled": true},
  {"group_name": "X25519MLKEM768", "enabled": false},
  {"group_name": "SecP256r1MLKEM768", "enabled": false},
  {"group_name": "MLKEM768", "enabled": false},
  {"group_name": "MLKEM1024", "enabled": false},
  {"group_name": "MLKEM512", "enabled": false}
}
```

tls-ciphers.json

```
{
  "tls_ciphers": [
    {
      "cipher_suite": "TLS_AES_256_GCM_SHA384",
      "enabled": true
    },
    {
      "cipher_suite": "TLS_CHACHA20_POLY1305_SHA256",
      "enabled": true
    },
    {
      "cipher_suite": "TLS_AES_128_GCM_SHA256",
      "enabled": true
    },
    {
      "cipher_suite": "ECDHE-ECDSA-AES256-GCM-SHA384",

```

```

    "enabled": true
  },
  {
    "cipher_suite": "ECDHE-RSA-AES256-GCM-SHA384",
    "enabled": true
  },
  {
    "cipher_suite": "ECDHE-ECDSA-AES128-GCM-SHA256",
    "enabled": true
  },
  {
    "cipher_suite": "ECDHE-RSA-AES128-GCM-SHA256",
    "enabled": true
  },
  {
    "cipher_suite": "ECDHE-ECDSA-AES256-SHA384",
    "enabled": false
  },
  {
    "cipher_suite": "ECDHE-ECDSA-AES128-SHA256",
    "enabled": false
  },
  {
    "cipher_suite": "ECDHE-RSA-AES256-SHA384",
    "enabled": false
  },
  {
    "cipher_suite": "ECDHE-RSA-AES128-SHA256",
    "enabled": false
  },
  {
    "cipher_suite": "ECDHE-ECDSA-AES256-SHA",
    "enabled": false
  },
  {
    "cipher_suite": "ECDHE-ECDSA-AES128-SHA",
    "enabled": false
  },
  {
    "cipher_suite": "ECDHE-RSA-AES256-SHA",
    "enabled": false
  },
  {
    "cipher_suite": "ECDHE-RSA-AES128-SHA",
    "enabled": false
  }
]
}

```

Output

```

"tls_ciphers":
[{"cipher_suite": "TLS_AES_256_GCM_SHA384", "enabled": true, "configurable": false, "hex_code": "0x1302"},
{"cipher_suite": "TLS_CHACHA20_POLY1305_SHA256", "enabled": true, "configurable": false, "hex_code": "0x1303"},
{"cipher_suite": "TLS_AES_128_GCM_SHA256", "enabled": true, "configurable": false, "hex_code": "0x1301"}]

```

```
e,"hex_code": "0x1301"},
{"cipher_suite": "ECDHE-ECDSA-AES256-GCM-SHA384","enabled": true,"configurable": true,"hex_code": "0xC02C"},
{"cipher_suite": "ECDHE-RSA-AES256-GCM-SHA384","enabled": true,"configurable": true,"hex_code": "0xC030"},
{"cipher_suite": "ECDHE-ECDSA-AES128-GCM-SHA256","enabled": true,"configurable": true,"hex_code": "0xC02B"},
{"cipher_suite": "ECDHE-RSA-AES128-GCM-SHA256","enabled": true,"configurable": true,"hex_code": "0xC02F"},
{"cipher_suite": "ECDHE-ECDSA-AES256-SHA384","enabled": false,"configurable": true,"hex_code": "0xC024"},
{"cipher_suite": "ECDHE-ECDSA-AES128-SHA256","enabled": false,"configurable": true,"hex_code": "0xC023"},
{"cipher_suite": "ECDHE-RSA-AES256-SHA384","enabled": false,"configurable": true,"hex_code": "0xC028"},
{"cipher_suite": "ECDHE-RSA-AES128-SHA256","enabled": false,"configurable": true,"hex_code": "0xC027"},
{"cipher_suite": "ECDHE-ECDSA-AES256-SHA","enabled": false,"configurable": true,"hex_code": "0xC00A"},
{"cipher_suite": "ECDHE-ECDSA-AES128-SHA","enabled": false,"configurable": true,"hex_code": "0xC009"},
{"cipher_suite": "ECDHE-RSA-AES256-SHA","enabled": false,"configurable": true,"hex_code": "0xC014"},
{"cipher_suite": "ECDHE-RSA-AES128-SHA","enabled": false,"configurable": true,"hex_code": "0xC013"}],
"tls_groups":
[{"group_name": "x25519","enabled": true},
{"group_name": "secp256r1","enabled": true},
{"group_name": "x448","enabled": true},
{"group_name": "secp521r1","enabled": true},
{"group_name": "secp384r1","enabled": true},
{"group_name": "ffdhe2048","enabled": true},
{"group_name": "ffdhe3072","enabled": true},
{"group_name": "ffdhe4096","enabled": true},
{"group_name": "ffdhe6144","enabled": true},
{"group_name": "ffdhe8192","enabled": true},
{"group_name": "X25519MLKEM768","enabled": false},
{"group_name": "SecP256r1MLKEM768","enabled": false},
{"group_name": "MLKEM768","enabled": false},
{"group_name": "MLKEM1024","enabled": false},
{"group_name": "MLKEM512","enabled": false}]
```

Key Exchange algorithms (KEX) for SSH interface

Caution

This feature is a technical preview for evaluation in non-production environments. Details and functionality are subject to change.

Key exchange algorithms are cryptographic protocols that allow two parties to securely exchange cryptographic keys over an insecure channel. These algorithms ensure that both parties can share a secret key, which can then be used for further encryption and decryption of messages.

The CipherTrust Manager provides the option to enable or disable the available key exchange algorithms. The supported key exchange algorithms include: 'diffie-hellman-group-exchange-sha256', 'curve25519-sha256@libssh.org', and 'mlkem768x25519-sha256'.

To get the Key exchange algorithms:

Syntax

```
ksctl ssh kex get
```

Example

```
ksctl ssh kex get
```

Response

```
{
  "kex_algorithms": [
    {
      "name": "diffie-hellman-group-exchange-sha256",
      "enabled": true
    },
    {
      "name": "curve25519-sha256@libssh.org",
      "enabled": true
    },
    {
      "name": "mlkem768x25519-sha256",
      "enabled": true
    }
  ]
}
```

To modify the status of key exchange algorithms:

Syntax

```
ksctl ssh kex modify --kex-algo-file <algo-file-name>
```

Example

```
ksctl ssh kex modify --kex-algo-file kex.json
```

Response

```
{
  "kex_algorithms": [
    {
      "name": "curve25519-sha256@libssh.org",
      "enabled": true
    },
    {
```



```

    "name": "diffie-hellman-group-exchange-sha256",
    "enabled": true
  },
  {
    "name": "mlkem768x25519-sha256",
    "enabled": false
  }
]
}

```

kex.json file

```

#bash
{
  "kex_algorithms": [
    {
      "name": "diffie-hellman-group-exchange-sha256",
      "enabled": true
    },
    {
      "name": "curve25519-sha256@libssh.org",
      "enabled": true
    },
    {
      "name": "mlkem768x25519-sha256",
      "enabled": true
    }
  ]
}

```

SSH command to connect with specific key exchange algorithm

Syntax

```
ssh -i key.pem -oKexAlgorithms=<key-exchange-algorithm><ksadmin@CM_IP>
```

Example

```
ssh -i key.pem -oKexAlgorithms=diffie-hellman-group-exchange-sha256
ksadmin@xx.xx.xxx.xx
```

Note

To connect to 'mlkem768x25519-sha256', the OpenSSH client must be running a minimum version of *OpenSSH_9.9p1-3*.

Proxy Configuration

The proxy configuration on the CipherTrust Manager allows you to configure what traffic should be forwarded through the proxy to the external network.

If you want to forward `https` traffic through a proxy, you need to set up the `https_proxy` environment on the CipherTrust Manager. Similarly, to forward the `http` traffic through a proxy, set up the `http_proxy` environment on the CipherTrust Manager. System restart is required after adding a new proxy or changing proxy settings.

Note

Proxy configurations, used in the [proxy test APIs](#) are directly used for proxy testing. These configurations are not stored as system environment variables. Restart is not required while using these configurations.

Let's discuss some of the scenarios based on the proxy configuration settings on the CipherTrust Manager and the actual proxy servers.

Scenario 1 - Same proxy servers for different traffic

When you don't have different proxy servers for different traffic, for example, having the same proxy server for `http` and `https` traffic. In such a scenario, it is recommended to configure both the `http_proxy` and `https_proxy` on the CipherTrust Manager to the same proxy server.

Example

```
http_proxy=http://test.com
```

```
https_proxy=http://test.com
```

Scenario 2 - HTTP proxy server supports HTTP tunneling

When you have an `http` proxy server that supports HTTP tunneling (For example, HTTP CONNECT), in such scenarios, you can set the `https_proxy` environment on the CipherTrust Manager to the same `http` proxy server.

Example

```
https_proxy=http://test.com
```

Scenario 3 - Protocol is not specified explicitly on the CipherTrust Manager

When you don't specify the protocol explicitly on the CipherTrust Manager, by default, the CipherTrust Manager assumes the protocol to be the same as specified in the proxy environment. However, both environments may not be identical every time and lead to non-functional configurations.

For example, if you set `https_proxy=test.com`, it is assumed by the CipherTrust Manager that the proxy server (`test.com`) supports `https`. In such a scenario, if the proxy server works on `http` only, then this configuration won't work.

Therefore, it is recommended to be *explicit* about the protocol your proxy server supports while configuring the CipherTrust Manager environment.

Example

```
https_proxy=https://test.com
```

```
http_proxy=http://test.com
```

The following connections use the configured proxy:

- [CipherTrust Cloud Key Manager](#) external connections to cloud services and key sources.
- CipherTrust Manager [root of trust connection](#) to the Thales DPoD HSM on Demand Service. Consult the [DPoD Client Network Connectivity page](#) or the `Chrystoki.conf` file in the client package for the FQDNs to whitelist on the proxy.
- [Data Discovery and Classification](#) connections to Hadoop Services, including Thales Data Platform. Configure the proxy server directly to access TDP, to resolve its name, and to allow HTTPS connections to port 8443.

DDC Agent connections are inbound, and so the CipherTrust Manager proxy configuration is not applied to them. If desired, you can set up a proxy outside of CipherTrust Manager for this connection, in keeping with the broader network.

Note

- If a proxy host is added using the proxy hostname, then IP mapping is required in the DNS. For details, refer to [Configuring DNS Hosts](#).
- For https proxy, the CipherTrust Manager allows you to add the ca-certificates to the list of trusted ca-certificates.

Note

- For CCKM, configure an HTTPS proxy.
- If your proxy server does not support HTTP CONNECT, then CCKM GCP connections from the CipherTrust Manager will not work using proxy with a certificate. Instead, add an exception (`cloudkms.googleapis.com`) with `no-proxy` or use proxy with username and password.

In the subsequent sections, let's learn to configure proxy using [GUI](#) and [CLI](#).

Configuring proxy using GUI

To set up a proxy host:

1. Log on to the CipherTrust Manager GUI.
2. Navigate to **Admin Settings > Proxy**. The list of available Proxy hosts is displayed, if available.
3. On the Proxy page, click **Add Proxy Host**.
4. On the Add Proxy Host screen:
 - a. Select Protocol type as **HTTP** or **HTTPS**.

-
- b. In the Hostname field, enter the proxy server IP address or URL.
 - c. In the Port field, enter proxy server's **Port**.
 - d. Add CA certificate. This field will appear on the UI if *HTTPS* proxy is selected. You can either upload the CA certificate file or paste the file content. This is an optional field.
 - e. Select the **proxy server required password** checkbox to authenticate the proxy server. Enter **Username**, **Password**, and **Test URL**. It is an optional step required only when proxy server is not already authenticated.

Note

If the proxy server password contains special characters, then replace these characters with encoded value. For example, if password is **tmp^123#** then the updated password is **tmp%5E123%23**.

- f. Click **Test Credentials** to check whether the Proxy configuration is successful. If the test is successful, the status is **OK** else the status is **Failed**.
- g. Click **Add Proxy Host**.

Configuring proxies using ksctl

The following operations can be performed:

- Add proxies
- Get/list proxies
- Delete proxies
- Update proxies
- Test proxies

Adding proxies

To add a proxy, run:

Syntax

```
ksctl proxy add --http-proxy <HTTP-proxy-address> --https-proxy <HTTPS-proxy-addr  
ess> --no-proxy <List-of-noproxy-addresses> --ca-cert-file <ca-certificate-file>
```

- **http-proxy** - proxy server URL capable of handling http network traffic.

- `https-proxy` - HTTPS proxy server URL for proxy configurations.
- `no-proxy` - Server URL to which the network traffic will be redirected bypassing the proxy.
- `ca-cert-file` - CA certificate to trust for proxy. This field is applicable only for *https* proxy. It is an optional field.

Note

If the proxy server password contains special characters, then replace these characters with encoded value. For example, if password is **tmp^123#** then the updated password is **tmp%5E123%23**.

Example Request 1

```
ksctl proxy add --https-proxy https://my.proxy.server:8081 --no-proxy localhost,
127.0.0.1 --ca-cert-file cacertfile
```

Example Response 1

```
{
  "https_proxy": "https://my.proxy.server:8081",
  "no_proxy":
  [
    "localhost",
    "127.0.0.1"
  ],
  "certificate": "-----BEGIN CERTIFICATE-----\nMIIEEuDCCAgCgAwIBAgIQOivQNtvy1bs
D+ZtTiktSbjANBgqhkiG9w0BAQsFADBa\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUQxEDAOBgNVBA
cTB0JlbGNhbXAxEADAQ\nnBgNVBAoTB0dlbWFSdG8xGjAYBgNVBAMTEUtleVNlY3VyZSBSb290IENBMB4XD
TIx\nnMDUwOTE1MTMwOFoXDTIzMDUwOTE1MTMwOFowJTEOMAwGA1UEAxMFYWRtaW4xEzAR\nnBgoJkiaJk/
IsZAEBAEwMxMjMwggEiMA0GCSqGSIb3DQEBQUAA4IBDwAwggEKAoIB\nnAQDf0/15sDKlmZ940mc3YAmPd
EHmAPf6kDZgtqpuN9ftXji65WIHywZ5VN/5YYVD\nnREdbS96kAdNMNyec8As0E0lbgirxaW2HFOzVcdfU
yh8FnQWq4kAcGBdL19gvdEm6\nnoZOaX6XlKZq3REfvFXjPg3YkhOvmaiF/9WFOvafCplpgpib3kiijd3m
1ZUHP+uxW\nnkfJ6ddxMs3Qe3gltfmpnjohY433rzH2CFr/W5wufRKZWmlu2OBwTKJsixJbcRJR1\nn93+X
VELt6r7UmrycZjmi3RIMkJ0WC+KpkL0ZetYtXL/7IykrKz1qAwKI4mpyJjAS\nn/3yQgJKCdSBz80Bzmbn
DevQ9AgMBAAGjga4wgaswDgYDVR0PAQH/BAQDAgOIMBMG\nnA1UdJQQMMAoGCCsGAQUFBwMCAwGA1UdEw
EB/wQMAAwHwYDVR0jBBgwFoAUl4YP\nnF8V39/lnMb8i5iDOptXjQ4owVQYDVR0fBE4wTDBKoEigRoZEa
HR0cDovL2tleXNl\nnY3VyZS5sb2NhbC9jcmxzL2Y4YWFiZDRmLTU0NTktNGNiMClhMjZhLWRmZDg4MTI
5\nnYmQ1ZS5jcmwWdQYJKoZIhvcNAQELBQADggIBAAYsYXivy9vD+WMqs4ceC+W3O8Tx\nnIW/jaCHfWZKX
r4fk01n1Mh020T67wIKqQUoTKgp9U7vmNmd/RKrl1NS191Eh8sm\nnHxy7/bvcSDXajw2LpsmIRaWeqYg
O0qOTluMQMMnSBiLbdgSAXKEAjRMQqvQfzqUV\nneTSPWaWzyFbnfhSEfU0s46Xs61gWTFvwc1vB40Xk7H
KFTNUP/xPIfLlhT4H9J3Bx\nnyrWz5bJY1z6C95/gXsQptccmYik+WGY7IJofvNJD8ugc1t6SeVG2aEl8
fNiuS5a\nnp906ThUcM3MqHcL0c0lqm9+jzs5j8pUwBJ+7lsDS17Y+uFvHEJN8XGXQLhFf3p/4\nnvNgyMA
mB9uvC5rbqEsCKUgpxkNa0sm0WflVoIQ1h2ku01yqtG8krma9qr4zy+bML\nnO6Zk37Vn1/8pUjGYWHIPh
jX6e+/wlRIMufyqKg7M/OHlg0S6eOpaX13tXxYNnaVm\nngN2mKfvmN3W6sMdtCKiFRNeTcuF5R7ZRWXKq
Hp00Y6N2Tk2FyZjgWAxUtg7VnLPW\nnRfuQBQ/Jud7zVDWxtftv6nmrV1nlqErPPDnRt3D49AD5lj4+Jhd
zKz47F094T++8\nn+rauAODq6i+FZe/05RwSCB1fqWJ8ja9gwAWaBVXfQpIDIY3KFTC2tZhjUUOii++d\nn
P6WaJc1NqTcWns8H\nn-----END CERTIFICATE-----\n"
```

Example Request 2

```
ksctl proxy add --http-proxy http://username:password@my.proxy.server:8080 --http
s-proxy https://username:password@my.proxy.server:8081 --no-proxy localhost,12
7.0.0.1
```

Example Response 2

```
{
  "http_proxy": "http://username:xxxxxxx@my.proxy.server:8080",
  "https_proxy": "https://username:xxxxxxx@my.proxy.server:8081",
  "no_proxy":
  [
    "localhost",
    "127.0.0.1"
  ]
}
```

Getting List of proxies

To get a list of proxies, run:

Syntax

```
ksctl proxy list
```

Example Request

```
ksctl proxy list
```

Example Response

```
{
  "http_proxy": "http://username:xxxxxxx@my.proxy.server:8080",
  "https_proxy": "https://username:xxxxxxx@my.proxy.server:8081",
  "no_proxy":
  [
    "localhost",
    "127.0.0.1"
  ],
  "certificate": "-----BEGIN CERTIFICATE-----\nMIIEEuDCCAgCgAwIBAgIQOivQNtvy1bs
D+ZtTiktSbjANBgkqhkiG9w0BAQsFADBa\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUQxEDAOBgNVBA
cTB0JlBGNhbXAxEADAQ\nMBgNVBAoTB0dlbWFSdG8xGjAYBgNVBAMTEUtleVNlY3VyZSBSb290IENBMB4XD
TIx\nNMDUwOTE1MTMwOFoXDTIzMDUwOTE1MTMwOFowJTEOMAwGA1UEAxMFYWRtaW4xEzAR\nnBgoJkiaJk/
IsZAEBEwMxMjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB\nnAQDf0/15sDKlmZ940mc3YAMPd
EHmAPf6kDZgtqpuN9ftXji65WIIHywZ5VN/5YYVD\nnREdbS96kAdNMNyec8As0E0lbgirxaW2HFOzVcdfU
yh8FnQWq4kAcGBdL19gvdEm6\nnoZOaX6XlKZq3REfvFXjPg3YkhOvmaIF/9WFOvafCplpgpib3kiijd3m
1ZUHP+uxW\nnkfJ6ddxMs3Qe3gltfmpnjoHY433rzH2CFr/W5wufRKZWmlu2OBwTKJsixJbcRJR1\nn93+X
VELt6r7UmrycZjmi3RIMkJOWC+KpkL0ZetYtXL/7IykRkzlgAwKI4mpyJjAS\nn/3yQgJKCdSBz80Bzmbn
DevQ9AgMBAAGjga4wgaswDgYDVR0PAQH/BAQDAgOIMBMG\nnA1UdJQMMMAoGCCsGAQUFBwMCAwGA1UdEw
EB/wQCAAwHwYDVR0jBBgwFoAU14YP\nnF8V39/1nMb8i5iDOptXjQ4owVQYDVR0fBE4wTDBKoEigRoZEa
HR0cDovL2tleXNl\nnY3VyZS5sb2Nhbc9jcmxzL2Y4YWFiZDRmLTUyNTktNGNiMC1hMjZlLWZDg4MTI
5\nnYmQ1ZS5jcmwwDQYJKoZIhvcNAQELBQADggIBAAYsYXivy9vD+WMqs4ceC+W3O8Tx\nnIW/jaCHfWZKX
r4fk01n1Mh020T67wIKqQXUoTKgp9U7vmNMd/RKrrj1NS19lEh8sm\nnHxy7/bvcSDXajw2LpsmIRaWeqYg
```

```
O0qOTluMQMMnSBiLbdgSAXKEAjRMQQvQfzqUV\neTSPWaWzyFbnfhSEfU0s46Xs61gWTfvwclvB40Xk7H
KFTNUP/xPIfLlhT4H9J3Bx\nyrWz5bJY1z6Cx95/gXsQptccmYik+WGY7IJofvNJD8ugc1t6SeVG2aEl8
fNiuS5a\np9O6ThUcM3MqHcL0cOlqm9+jzs5j8pUWbJ+7lsDS17Y+uFvHEJN8XGXQLhFf3p/4\nvNgyMA
mB9uvC5rbqEsCKUgpxkNa0sm0WflVoIQ1h2ku01yqtG8krma9qr4zy+bML\nO6Zk37Vn1/8pUjGYWHIPh
jX6e+/w1RIMufyqKg7M/OHlg0S6eOpaX13tXxYNnaVm\ngN2mKfvmN3W6sMdtCKifrNeTcuF5R7ZRWXKq
Hp00Y6N2Tk2FyZjgWAxUtq7VnLPW\nRfuQBQ/Jud7zVDWxtftv6nmrV1nlqErPPDnRt3D49AD51j4+Jhd
zKz47F094T++8\n+rauAODq6i+FZe/05RwSCB1fqWJ8ja9gwAWaBVXfQpIDIY3KFTC2tZhjUUOii++d\n
P6WaJc1NqTcWns8H\n-----END CERTIFICATE-----\n"
}
```

Deleting proxies

To delete proxy configurations, run:

Syntax

```
ksctl proxy delete
```

Example Request

```
ksctl proxy delete
```

No response is displayed if the proxy is deleted successfully. The ca-certificate of the HTTPS proxy is also deleted.

Updating proxies

To update a proxy configuration, run:

Syntax

```
ksctl proxy update --http-proxy <HTTP-proxy-address> --https-proxy <HTTPS-proxy-a
ddress> --no-proxy <List-of-noproxy-addresses> --ca-cert-file <ca-certificate-fil
e>
```

Example Request

```
ksctl proxy update --http-proxy http://dummyproxy:3000 --https-proxy https://dumm
yproxy:3000
```

Example Response

```
{
  "http_proxy": "http://dummyproxy:3000",
  "https_proxy": "https://dummyproxy:3000",
  "no_proxy": [
    "localhost",
    "127.0.0.1"
  ],
  "certificate": "-----BEGIN CERTIFICATE-----\nMIIEEuDCCAqCgAwIBAgIQOivQNtvy1bs
```

```
D+ZtTiktSbjANBgkqhkiG9w0BAQsFADBa\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUQxEDAObgNVBA
cTB0JlBGNhbXAxEDA0\nBgNVBAoTB0dlbWFSdG8xGjAYBgNVBAMTEUtleVNlY3VyZSBSb290IENBMB4XD
TIx\nMDUwOTE1MTMwOFoXDTIzMDUwOTE1MTMwOFowJTEOMAwGA1UEAxMFYWRtaW4xEzAR\nnBgoJkiaJk/
IsZAEBEwMxMjMwggiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB\nnAQDf0/15sDKlmZ940mc3Yampd
EHmApf6kDZgtqpuN9ftXji65WIHywZ5VN/5YYVD\nnREdbS96kAdNMNyec8As0E0lbgirxaW2HFOzVcdfU
yh8FnQWq4kAcGBdL19gvdEm6\nnoZOaX6XlKZq3REfvFXjPg3YkhOvmaiF/9WFOvafCplpgpib3kiijd3m
1ZUHP+uxW\nnkfJ6ddxMs3Qe3gltfmpnjoHY433rzh2CFr/W5wufRKZWmlu2OBwTKJsixJbcRJR1\nn93+X
VELt6r7UmrycZjmi3RIMkJ0WC+KpkL0ZetYtXL/7IyRkz1qAwKI4mpyJjAS\nn/3yQgJKCdSBz80Bzmbn
DevQ9AgMBAAGjga4wgaswDgYDVR0PAQH/BAQDAgOIMBMG\nnA1UdJQQMMAoGCCsGAQUFBwMCAwGA1UdEw
EB/wQCAAwHwYDVR0jBBgwFoAUl4YP\nnF8V39/lnMb8i5iDOptXjQ4owVQYDVR0fBE4wTDBKoEigRoZEa
HR0cDovL2tleXNl\nnY3VyZS5sb2Nhbc9jcmxzL2Y4YWFiZDRmLTU0NTktNGNiMClhMjZhLWRmZDg4MTI
5\nnYmQ1ZS5jcmwWdQYJKoZIhvcNAQELBQADggIBAAysYXivy9vD+WMqs4ceC+W3O8Tx\nnIW/jaCHfWZKX
r4fk01n1Mh020T67wIKqQXUoTKgp9U7vmNMD/RKrrj1NS191Eh8sm\nnHxy7/bvcSDXajw2LpsmIRaWeqYg
O0qOTlUMQMMnSBiLbdgSAXKEAjRMQqvQfzqUV\nneTSPWaWzyFbnfhSEfU0s46Xs61gWTFvwclvB40Xk7H
KFTNUP/xPIfLlhT4H9J3Bx\nnyrWz5bJY1z6Ccx95/gXsQptccmYik+WGY7IJofovNJD8ugc1t6SeVG2aEl8
fNiuS5a\nnp9O6ThUcM3MqHcL0cOlqm9+jzs5j8pUwbJ+7lsDS17Y+uFvHEJN8XGXQLhFf3p/4\nnvNgyMA
mB9uvC5rbqEsCKUgpxkNa0sm0WflVoIQ1h2ku01yqtG8krma9qr4zy+bML\nnO6Zk37Vn1/8pUjGYWHIPh
jX6e+/wlRIMufyqKq7M/OHlg0S6eOpaX13tXxYNnaVm\nngN2mKfvmN3W6sMdtCKifrNeTcuF5R7ZRWXKq
Hp00Y6N2Tk2FyZjgWAxUtg7VnLPW\nnRfuQBQ/Jud7zVDWxtftv6nmrV1nlqErPPDnRt3D49AD5lj4+Jhd
zKz47F094T++8\nn+rauAODq6i+FZe/05RwSCB1fqWJ8ja9gwAWaBVXfQpIDIY3KFTC2tZhjUUOii++d\nn
P6WaJc1NqTcWns8H\nn-----END CERTIFICATE-----\n"
}
```

Testing proxies

To test proxy with a given URL, run:

Syntax

```
ksctl proxy test --http-proxy <HTTP-proxy-address> --test-url <Test-url> --https-
proxy <HTTPS-proxy-address> --ca-cert-file <ca-certificate-file>
```

Example Request 1

```
ksctl proxy test --https-proxy https://my.proxy.server:8081 --test-url https://ww
w.thalesdocs.com --ca-cert-file cacertfile
```

Example Response 1

```
{
  connection_ok: true
}
```

Example Request 2

```
ksctl proxy test --http-proxy http://my.proxy.server:8081 --test-url https://ww
w.thalesdocs.com
```

Example Response 2

```
{
```



```
connection_ok: true
}
```

Quorums

This feature allows an administrator to configure a system (or quorum) that can have multiple approvers for an operation. These approvers need to approve the operations before they can be successfully executed.

Note

Quorum policies are domain-scoped and can control only the operations taking place in the domain they are enabled.

The below [table](#) describes the list of operations supported by quorums and the respective authorized groups for approval. The approvers **must** be part of any respective authorized group to approve the corresponding operation.

Note

If you are using connectors and direct NAE scripts, do not enable the DeleteKey and AddUserToGroup quorums.

Operations Supported by Quorums

Operation	Authorized Group(s) for Approval	Auto-Executable
Deletekey	Key Admins	No
AddUserToGroup	User Admins	Yes
DownloadBackupKey [Root Domain Only]	Restore Admins and Backup Admins	No
RestoreBackup [Root Domain Only]	Restore Admins	No
DeleteDomain	Domain Admins	No
ManagePolicyAttachment	admin	No
DeletePolicy	admin	No
DownloadBackupKeyDomain	Domain Backup Admins and Domain Restore Admins	No
RestoreBackupDomain	Domain Restore Admins	No
DeleteClientCTE	CTE Admins	Yes
DeleteClientGroupCTE	CTE Admins	Yes

Operation	Authorized Group(s) for Approval	Auto-Executable
DeleteCSISStorageGroupCTE	CTE Admins	Yes
DeleteGuardPointCTE	CTE Admins	Yes
DeletePolicyCTE	CTE Admins	Yes
DeletePolicyElementsCTE	CTE Admins	Yes
DeleteProfileCTE	CTE Admins	Yes
UpdateClientGroupCTE	CTE Admins	Yes
UpdateCSISStorageGroupCTE	CTE Admins	Yes
UpdateGuardPointCTE	CTE Admins	Yes
UpdatePolicyCTE	CTE Admins	Yes
UpdatePolicyElementsCTE	CTE Admins	Yes
UpdateProfileCTE	CTE Admins	Yes
UpdateClientPasswordCTE	CTE Admins	Yes
UpdateClientGroupPasswordCTE	CTE Admins	Yes
CreateAttachedPolicy	admin	No
DeleteAttachedPolicy	admin	No
DeleteGoogleWorkspaceCSEEndpoint	CCKM Admins	No
DeleteEKMEndpoint	CCKM Admins	No
UpdateQuorumProfile	admin	Yes
ArchiveKey	Key Admins	No
RecoverKey	Key Admins	No
RevokeKey	Key Admins	No
ReActivateKey	Key Admins	No
CreateLink	Key Admins	No
UpdateLink	Key Admins	No
DeleteLink	Key Admins	No
UpdateAuthConnection (LDAP connection)	User Admins	No
DownloadBackup	Backup Admins and Restore Admins	No
DownloadBackupDomain	Domain Backup Admins and Domain Restore Admins	No

The above table also defines whether the list of operations supported by quorums are auto-executable after approval. If auto-execution for the quorum profile is enabled for any operation, after approval, that operation will execute automatically. Hence, eliminates the need to re-trigger that operation manually. The `auto_executable` flag is not configurable.

Currently, the issue with the quorum policies is that an admin may, for some reason or maliciously, change the number of voters, excluded and voter groups information on a quorum policy. This may lead to bypassing the quorum requirements. To handle this issue, the `UpdateQuorumProfile` operation has been introduced. Once a quorum is enabled for `UpdateQuorumProfile`, the admin has to go through the quorum's approval process to modify the quorum policies. To enable `UpdateQuorumProfile`, refer to [Activating the Quorum Policy](#).

The authorized group(s) for quorum approval associated with operations are configurable. To configure, refer to [Updating Quorum Profile](#).

CTE and CTE UserSpace support CipherTrust Manager's quorum feature. Refer to the Quorum Control section for details on supported [CTE](#) and [CTE UserSpace](#) operations and resources.

Managing Quorum on Policies

The CipherTrust Manager allows you to enable quorum for:

- `ManagePolicyAttachment`
- `DeletePolicy`

These policies can be system policies, user policies, or quorum policies.

Once a quorum is enabled for `ManagePolicyAttachment` and `DeletePolicy`, the activation and deactivation of a quorum for any supported operation has to go through the quorum's approval process, similar to any other operation on the CipherTrust Manager.

Caution

Do not enable quorum on the `ManagePolicyAttachment` and `DeletePolicy` operations until all the CipherTrust Manager nodes in a cluster are upgraded to 2.10 or a higher version.

Note

It is recommended to enable quorum on both `DeletePolicy` and `ManagePolicyAttachment` operations together in the same order to bring the entire policy creation and deletion under quorum.

ManagePolicyAttachment

The `ManagePolicyAttachment` operation gives you a provision to enable a quorum for:

- Creating policy attachments
- Deleting policy attachments

When you activate a quorum, a policy and a policy attachment is created internally. If you want to bring activation of a quorum policy under a quorum, activate the quorum on the `ManagePolicyAttachment`.

DeletePolicy

The DeletePolicy operation is used to enable quorum on policy deletion. When you deactivate a quorum, a policy is deleted internally. If you want to bring deactivation of a quorum policy under a quorum, activate the quorum on the DeletePolicy action.

Refer to [Enabling Quorum on ManagePolicyAttachment and DeletePolicy](#) for more details.

Quorum Policies

To enable the quorum for any operation, first you need to activate the quorum policy. After the quorum policy is active for an operation, a quorum gets created in a `pre-active` state on performing the corresponding operation. To activate a quorum policy, refer to [Activating the Quorum Policy](#). For more details on quorum policies, refer to [Managing Quorums Policies using ksctl](#).

Quorum Profile

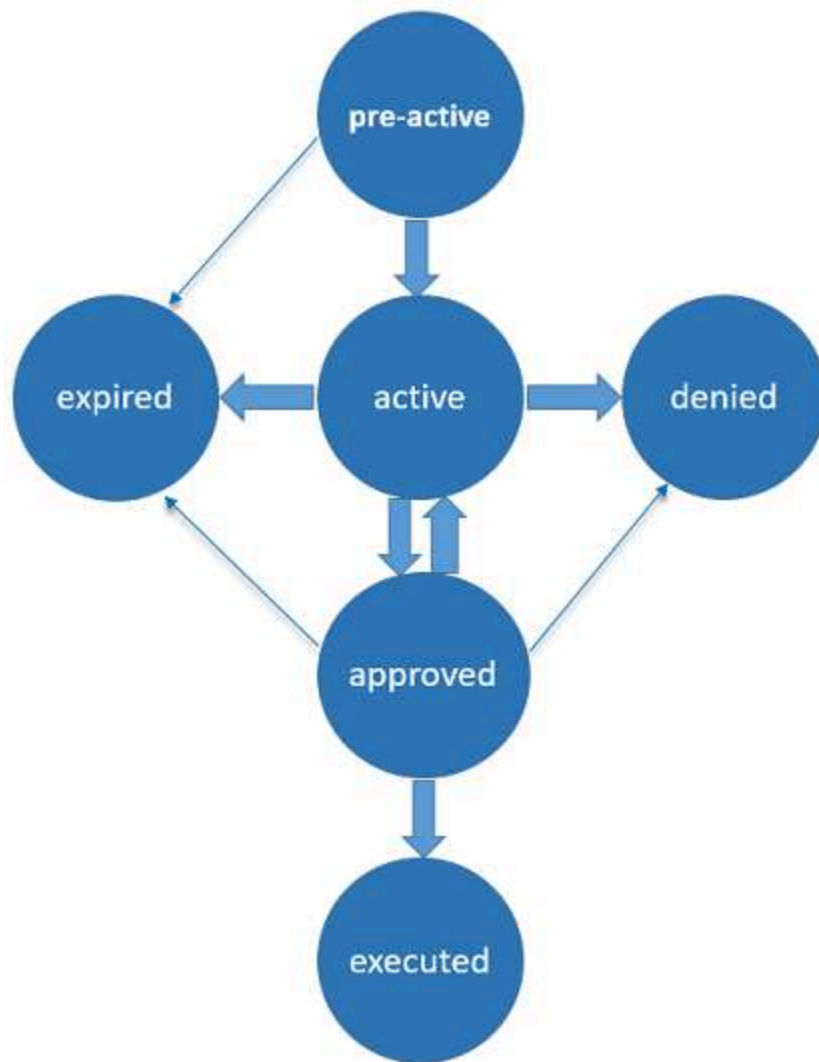
It's a configuration that defines the expiration time, number of approvals, and the voter groups for a quorum. For more details on quorum profile, refer to [Managing Quorums Profiles using ksctl](#).

States & Life-cycle of a Quorum Request

If an administrator has configured a quorum for any operation and the user initiates that operation, then that operation is denied and leads to creation of a quorum request.

The quorum request life-cycle is as follows:

- The quorum is created in a `pre-active` state. A quorum or a quorum request can have one of these six states at a time as depicted in the below diagram.

**Note**

The user/requester of this quorum must activate the quorum before any of the approvers can review the quorum request.

- Once the quorum is in the `active` state, it is available to the approvers to review it. The user/requester can optionally add a reason for initiating the request while activating it.
- When the required approvals are granted, the quorum is set to an `approved` state.
- A quorum request is set to an `executed` state if the operation has been completed with an approved quorum.
- A deny vote forces a quorum to be in a `denied` state implying that the operation can not be performed and the quorum is terminated.

In case, a quorum already exists for the same operation, the status of the existing quorum is used to determine the outcome of the operation. Any approver can vote either to approve or deny a quorum request. A note can be added with the vote for additional information.

Note

A quorum in a `pre-active` state is prone to an expedited expiry. It expires after 15 minutes of creation, if not activated.

Note

Every quorum expires after seven (7) days of the creation and this leads to the termination of that quorum.

Managing Quorums Policies

The following operations can be performed:

- Activate the quorum policy
- Get status of the quorum policy
- Deactivate the quorum policy

You can perform these operations using the `ksctl` or CipherTrust Manager GUI interface.

Activating Quorum Policy

ksctl

To activate quorum policy using `ksctl`, use the following command.

Syntax

```
ksctl quorum-policy activate --actions <Actions>
```

See [Operations Supported by Quorums](#) to see supported actions.

Example 1

Request

```
ksctl quorum-policy activate --actions "DeleteKey"
```

Response

```
{
  "Policy": {
    "id": "fd6f911b-f280-492c-9eac-aed35590d530",
    "uri": "kylo:kylo:admin:policies:enablequorum-fd6f911b-f280-492c-9eac-aed35590d530",
    "account": "kylo:kylo:admin:accounts:kylo",
    "application": "ncryptify:gemalto:admin:apps:kylo",
    "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
    "createdAt": "2021-06-01T04:45:48.917228Z",
    "name": "EnableQuorum",
    "actions": [
      "DeleteKey"
    ],
    "resources": null,
    "allow": false,
    "effect": "obligate_on_allow",
    "conditions": [
      {
        "path": "context.principal.cust.groups",
        "op": "contains",
        "negate": true
      }
    ],
    "updatedAt": "2023-06-01T10:11:11.000186Z"
  },
  "required_approvals": 2
}
```

Example 2

Request

```
ksctl quorum-policy activate --actions "UpdateQuorumProfile"
```

Response

```
{
  "Policy": {
    "id": "d75872e9-4ff5-4ae7-92ec-483d334bc5f7",
    "uri": "kylo:kylo:admin:policies:updatequorumprofile-requires-quorum-d75872e9-4ff5-4ae7-92ec-483d334bc5f7",
    "account": "kylo:kylo:admin:accounts:kylo",
    "application": "ncryptify:gemalto:admin:apps:kylo",
    "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
    "createdAt": "2023-07-17T04:15:14.556005Z",
    "name": "UpdateQuorumProfile requires quorum",
    "deletedAt": null,
    "updatedAt": "2023-07-17T04:15:14.556005Z",
    "actions": [
      "UpdateQuorumProfile"
    ],
    "resources": null,
    "allow": false,
    "effect": "obligate_on_allow",
    "conditions": [
      {

```

```

        "path": "context.principal.cust.groups",
        "op": "contains",
        "values": null,
        "negate": true
    },
    ],
    "obligations": [
    {
        "type": "quorum",
        "attributes": {
            "fingerprint": [
                "{{action}}",
                "{{resource}}",
                "{{context.environment.updates}}"
            ],
            "quorum_profile": "UpdateQuorumProfile requires quorum"
        }
    }
    ],
    },
    "required_approvals": 3
}

```

GUI

To activate a quorum policy using the CipherTrust Manager GUI:

1. Open the CipherTrust Manager GUI.
2. Under **Admin Settings**, select **Quorum Policy**.
3. Search the quorum policy you want to activate.
4. Turn on the toggle switch under **Enable Quorum Policy** for the desired quorum policy.

Getting Status of Quorum Policy

ksctl

To know whether the quorum policy is in active state or inactive state, use the following command.

Example Request

```
ksctl quorum-policy status
```

Example Response

```

{
  "skip": 0,
  "limit": 35,
  "total": 35,
  "resources": [
    {

```



```

        "operation": [
            "AddUserToGroup"
        ],
        "active": false,
        "profile": "AddUserToGroup requires quorum",
        "description": "Adding users to groups will require approvals from User Admins"
    },
    {
        "operation": [
            "DeleteCSISStorageGroupCTE"
        ],
        "active": false,
        "profile": "DeleteCSISStorageGroupCTE requires quorum",
        "description": "Delete CTE CSI Storage Group will require approvals from CTE Admins"
    },
    {
        "operation": [
            "DeleteClientCTE"
        ],
        "active": false,
        "profile": "DeleteClientCTE requires quorum",
        "description": "Deleting CTE Client will require approvals from CTE Admins"
    },
    {
        "operation": [
            "DeleteClientGroupCTE"
        ],
        "active": false,
        "profile": "DeleteClientGroupCTE requires quorum",
        "description": "Deleting CTE Client Group will require approvals from CTE Admins"
    },
    {
        "operation": [
            "DeleteDomain"
        ],
        "active": false,
        "profile": "DeleteDomain requires quorum",
        "description": "Deleting domains requires approvals from Domain Admins"
    },
    {
        "operation": [
            "DeleteEKMEndpoint"
        ],
        "active": false,
        "profile": "DeleteEKMEndpoint requires quorum",
        "description": "Deleting EKM endpoints requires approvals from CCKM Admins"
    },
    {
        "operation": [
            "DeleteGoogleWorkspaceCSEndpoint"
        ],
        "active": false,
        "profile": "DeleteGoogleWorkspaceCSEndpoint requires quorum",

```

```

        "description": "Deleting GoogleWorkspace CSE Endpoints requires approvals from CCKM Admins"
    },
    {
        "operation": [
            "DeleteGuardPointCTE"
        ],
        "active": false,
        "profile": "DeleteGuardPointCTE requires quorum",
        "description": "Deleting CTE Guard-Point will require approvals from CTE Admins"
    },
    {
        "operation": [
            "DeleteKey"
        ],
        "active": true,
        "profile": "DeleteKey requires quorum",
        "description": "Deleting Keys will require approvals from Key Admins"
    },
    {
        "operation": [
            "DeletePolicy"
        ],
        "active": false,
        "profile": "DeletePolicy requires quorum",
        "description": "Deleting Policies will require approvals from admins"
    },
    {
        "operation": [
            "DeletePolicyCTE"
        ],
        "active": false,
        "profile": "DeletePolicyCTE requires quorum",
        "description": "Delete CTE Policy will require approvals from CTE Admins"
    },
    {
        "operation": [
            "DeletePolicyElementsCTE"
        ],
        "active": false,
        "profile": "DeletePolicyElementsCTE requires quorum",
        "description": "Delete CTE Policy elements will require approvals from CTE Admins"
    },
    {
        "operation": [
            "DeleteProfileCTE"
        ],
        "active": false,
        "profile": "DeleteProfileCTE requires quorum",
        "description": "Delete CTE Profile will require approvals from CTE Admins"
    },
    {
        "operation": [
            "DownloadBackupkey"
        ],

```

```

        "active": false,
        "profile": "DownloadBackupkey requires quorum",
        "description": "Downloading System Backup Keys will require approvals
from either Backup Admins or Restore Admins"
    },
    {
        "operation": [
            "DownloadBackupkeyDomain"
        ],
        "active": false,
        "profile": "DownloadBackupkeyDomain requires quorum",
        "description": "Downloading Domain Backup Keys will require approvals
from either Domain Backup Admins or Domain Restore Admins"
    },
    {
        "operation": [
            "ManagePolicyAttachment"
        ],
        "active": false,
        "profile": "ManagePolicyAttachment requires quorum",
        "description": "ManagePolicyAttachment includes creating and deleting
a policy attachment. This will require approvals from admins."
    },
    {
        "operation": [
            "RestoreBackup"
        ],
        "active": false,
        "profile": "RestoreBackup requires quorum",
        "description": "Restoring System Backups will require approvals from
Restore Admins"
    },
    {
        "operation": [
            "RestoreBackupDomain"
        ],
        "active": false,
        "profile": "RestoreBackupDomain requires quorum",
        "description": "Restoring Domain Backups will require approvals from
Domain Restore Admins"
    },
    {
        "operation": [
            "UpdateCSISStorageGroupCTE"
        ],
        "active": false,
        "profile": "UpdateCSISStorageGroupCTE requires quorum",
        "description": "Update CTE CSI Storage Group will require approvals f
rom CTE Admins"
    },
    {
        "operation": [
            "UpdateClientGroupCTE"
        ],
        "active": false,
        "profile": "UpdateClientGroupCTE requires quorum",
        "description": "Update CTE Client Group will require approvals from C
TE Admins"
    },

```

```

    {
      "operation": [
        "UpdateGuardPointCTE"
      ],
      "active": false,
      "profile": "UpdateGuardPointCTE requires quorum",
      "description": "Update CTE GuardPoint will require approvals from CTE
Admins"
    },
    {
      "operation": [
        "UpdatePolicyCTE"
      ],
      "active": false,
      "profile": "UpdatePolicyCTE requires quorum",
      "description": "Update CTE Policy will require approvals from CTE Adm
ins"
    },
    {
      "operation": [
        "UpdatePolicyElementsCTE"
      ],
      "active": false,
      "profile": "UpdatePolicyElementsCTE requires quorum",
      "description": "Update CTE Policy elements will require approvals fro
m CTE Admins"
    },
    {
      "operation": [
        "UpdateProfileCTE"
      ],
      "active": false,
      "profile": "UpdateProfileCTE requires quorum",
      "description": "Update CTE Profile will require approvals from CTE Ad
mins"
    },
    {
      "operation": [
        "UpdateQuorumProfile"
      ],
      "active": false,
      "profile": "UpdateQuorumProfile requires quorum",
      "description": "Update quorum profile requires approvals from Admins"
    },
    {
      "operation": [
        "ArchiveKey"
      ],
      "active": false,
      "profile": "ArchiveKey requires quorum",
      "description": "Archive Keys will require approvals from Key Admins"
    },
    {
      "operation": [
        "RecoverKey"
      ],
      "active": false,
      "profile": "RecoverKey requires quorum",
      "description": "Recover Keys will require approvals from Key Admins"
    }
  ]
}

```

```

    },
    {
      "operation": [
        "RevokeKey"
      ],
      "active": false,
      "profile": "RevokeKey requires quorum",
      "description": "Revoke Keys will require approvals from Key Admins"
    },
    {
      "operation": [
        "ReActivateKey"
      ],
      "active": false,
      "profile": "ReActivateKey requires quorum",
      "description": "Reactivate Keys will require approvals from Key Admins"
    },
    {
      "operation": [
        "CreateLink"
      ],
      "active": false,
      "profile": "CreateLink requires quorum",
      "description": "CreateLink will require approvals from Key Admins"
    },
    {
      "operation": [
        "UpdateLink"
      ],
      "active": false,
      "profile": "UpdateLink requires quorum",
      "description": "UpdateLink will require approvals from Key Admins"
    },
    {
      "operation": [
        "DeleteLink"
      ],
      "active": false,
      "profile": "DeleteLink requires quorum",
      "description": "DeleteLink will require approvals from Key Admins"
    },
    {
      "operation": [
        "UpdateAuthConnection"
      ],
      "active": false,
      "profile": "UpdateAuthConnection requires quorum",
      "description": "Updating connections will require approvals from User Admins"
    },
    {
      "operation": [
        "DownloadBackup"
      ],
      "active": false,
      "profile": "DownloadBackup requires quorum",
      "description": "Downloading System Backup will require approvals from either Backup Admins or Restore Admins"
    }
  ],
  "active": true,
  "profile": "CipherTrust Manager requires quorum",
  "description": "CipherTrust Manager will require approvals from Key Admins"
}

```

```

    },
    {
      "operation": [
        "DownloadBackupDomain"
      ],
      "active": false,
      "profile": "DownloadBackupDomain requires quorum",
      "description": "Downloading Domain Backup will require approvals from
either Domain Backup Admins or Domain Restore Admins"
    }
  ]
}

```

The 'active' field represents whether policy is active or inactive. It returns `true` if the policy is active and `false` otherwise.

GUI

In the CipherTrust Manager GUI, the toggle switch under the **Enable Quorum Policy** column on the **Quorum Policy** page indicates the active/inactive status of a quorum policy. When switched on, it indicates that the policy is active, while when switched off, it indicates that the policy is inactive.

Deactivating Quorum Policy

ksctl

To deactivate an already active quorum policy for any specific operation, use the following command syntax.

Syntax

```
ksctl quorum-policy deactivate --actions <Actions>
```

Example Request

```
ksctl quorum-policy deactivate "DeleteKey"
```

Example Response

There will be no response if quorum policy is deactivated successfully.

GUI

To deactivate an active quorum policy:

1. Open the CipherTrust Manager GUI.
2. Under **Admin Settings**, select **Quorum Policy**.
3. Search the desired quorum policy.
4. Turn off the toggle switch under **Enable Quorum Policy** for the desired quorum policy.

Enabling Quorum on ManagePolicyAttachment and DeletePolicy

Enabling a quorum on ManagePolicyAttachment and DeletePolicy brings activation/deactivation of a quorum policy under a quorum. It implies that you won't be able to activate/deactivate any quorum policy on your own without creating a quorum for that request.

Let's consider a scenario where you have [activated a quorum](#) for the "DeleteKey" operation. Currently, you are allowed to [deactivate the created quorum](#).

However, if a quorum is enabled on ManagePolicyAttachment and DeletePolicy, you cannot activate/deactivate any quorum.

You can use the ksctl or CipherTrust Manager GUI to enable or disable a quorum on ManagePolicyAttachment and DeletePolicy.

ksctl

ManagePolicyAttachement

Enable ManagePolicyAttachment quorum

```
ksctl quorum-policy activate --actions "ManagePolicyAttachment"
```

Disable ManagePolicyAttachment quorum

```
ksctl quorum-policy deactivate --actions "ManagePolicyAttachment"
```

Enable Response

```
{
  "Policy": {
    "id": "865baeb0-cace-4e76-943c-3fea78c0252d",
    "uri": "kylo:kylo:admin:policies:managepolicyattachment-requires-quorum-865baeb0-cace-4e76-943c-3fea78c0252d",
    "account": "kylo:kylo:admin:accounts:kylo",
    "application": "ncryptify:gemalto:admin:apps:kylo",
    "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
    "createdAt": "2022-09-08T11:05:48.488623Z",
    "name": "ManagePolicyAttachment requires quorum",
    "actions": [
      "CreateAttachedPolicy",
      "DeleteAttachedPolicy"
    ],
    "resources": null,
    "allow": false,
    "effect": "obligate_on_allow",
    "conditions": [
      {
        "path": "context.principal.cust.groups",
        "op": "contains",
        "negate": true
      }
    ]
  },
}
```

```

      "updatedAt": "2023-06-01T10:18:59.469775Z"
    },
    "required_approvals": 3
  }

```

DeletePolicy

Enable DeletePolicy quorum

```
ksctl quorum-policy activate --actions "DeletePolicy"
```

Disable DeletePolicy quorum

```
ksctl quorum-policy deactivate --actions "DeletePolicy"
```

Enable Response

```

{
  "Policy": {
    "id": "e735b3e8-4698-45ad-814a-4ddfad75c572",
    "uri": "kylo:kylo:admin:policies:deletepolicy-requires-quorum-e735b3e8-4698-45ad-814a-4ddfad75c572",
    "account": "kylo:kylo:admin:accounts:kylo",
    "application": "ncryptify:gemalto:admin:apps:kylo",
    "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
    "createdAt": "2022-09-08T11:05:45.325665Z",
    "name": "DeletePolicy requires quorum",
    "actions": [
      "DeletePolicy"
    ],
    "resources": null,
    "allow": false,
    "effect": "obligate_on_allow",
    "conditions": [
      {
        "path": "context.principal.cust.groups",
        "op": "contains",
        "negate": true
      }
    ],
    "updatedAt": "2023-06-01T10:19:39.524036Z"
  },
  "required_approvals": 3
}

```

After activating the "DeletePolicy" quorum, if you try to [deactivate a quorum policy](#) for "DeleteKey" (created above), an error is thrown as shown below and the deactivation of quorum is not allowed:

```

{
  "code": 4,
  "codeDesc": "NCERRInsufficientPermissions",
  "message": "A Quorum has been created with ID (fd6f911b-f280-492c-9eac-aed355

```



```
90d530) in pre-active state. Please activate it."
}
```

GUI

To enable (or disable) ManagePolicyAttachment or DeletePolicy:

1. Open the CipherTrust Manager GUI.
2. Under **Admin Settings**, select **Quorum Policy**.
3. Search for 'Manage Policy Attachment' or 'Delete Policy' in the search bar.
4. Turn on (off) the toggle switch under **Enable Quorum Policy** for the desired policy.

Managing Quorums

The following operations can be performed:

- Activate quorum
- Approve quorum
- Deny quorum
- List/Search quorums
- Get quorum
- Delete quorum
- Revoke vote

You can perform these operations using the ksctl or CipherTrust Manager GUI interface.

ksctl

Activating Quorums

To activate a quorum, run:

Syntax

```
ksctl quorum activate --id <QuorumId> --quorum-reason <Reason-to-activate>
```

It changes the state of the quorum from `pre-active` to `active`. A quorum can be approved only when it is in the `active` state.

After quorum is active, it is available for the approval process.

Example Request

```
ksctl quorum activate --id 558622f8-bf20-4ded-9f43-e72bfaaf73a5
```

Example Response

```
{
  "id": "558622f8-bf20-4ded-9f43-e72bfaaf73a5",
  "uri": "kylo:kylo:sallyport:quorum:558622f8-bf20-4ded-9f43-e72bfaaf73a5",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-06-01T10:25:21.691222Z",
  "updatedAt": "2023-06-01T10:26:02.944903Z",
  "requester_id": "local|2e8bcf46-dcba-4f5f-89d2-8ea16c7c305f",
  "requester_username": "admin",
  "operation_fingerprint": "872a47b84aae3c37e2c966e135b931fcf6642fc23bc7df6cbc3c27eeae759546",
  "operation": "DeleteKey",
  "resource_uri": "kylo:kylo:vault:keys:test123-v0",
  "state": "active",
  "required_approvals": 3,
  "meta": {
    "details": {
      "policy": {
        "type": "policy_link",
        "value": {
          "id": "558622f8-bf20-4ded-9f43-e72bfaaf73a5",
          "name": "Deletekey requires quorum"
        }
      }
    }
  },
  "votes": [],
  "auto_executable": false
}
```

Note

The requester of the operation is also the owner of the associated quorum. Only the requester/owner has permission to activate the associated quorum.

Approving Quorums

To approve a quorum, run:

Syntax

```
ksctl quorum approve --id <QuorumId> --note <Additional-note-for-approval>
```

When all the required approvals are available, then quorum moves to the `approved` state and you can re-initiate the associated operation.

Example Request

```
ksctl quorum approve --id 558622f8-bf20-4ded-9f43-e72bfaaf73a5
```

Example Response

There will be no response if quorum is approved successfully.

Denying Quorums

To deny a approval to a quorum, run:

Syntax

```
ksctl quorum deny --id <QuorumId> --note <Additional-note-for-denial>
```

This command moves a quorum to the denied state. This is a terminal state and quorum is unusable after deny.

Example Request

```
ksctl quorum deny --id 558622f8-bf20-4ded-9f43-e72bfaaf73a5
```

Example Response

There will be no response if quorum is denied successfully.

Getting List of Quorums

To get the list of quorums, run:

Syntax

```
ksctl quorum list
```

Example Request

```
ksctl quorum list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "id": "335f45bb-9609-4e19-9996-a2977d7927bd",
      "uri": "kylo:kylo:sallyport:quorum:335f45bb-9609-4e19-9996-a2977d7927bd",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-09-11T06:38:18.712862Z",
      "updatedAt": "2023-09-11T06:38:19.055473Z",
```

```

    "requester_id": "local|a6ae00b0-2499-4f61-b5e0-b3da274d4060",
    "operation_fingerprint": "72fedf72283da5ad91f596c2ba8b9efeada151cf8ad
ec3eaf4c94efaec2494e9",
    "operation": "DeleteKey",
    "requester_username": "admin",
    "resource_uri": "kylo:kylo:vault:keys:test-v0",
    "state": "active",
    "required_approvals": 1,
    "meta": {
      "details": {
        "key": {
          "type": "key_link",
          "value": {
            "id": "dc845651e39e48efbae7470a5de13f8c501c6c547de34c
88a3e85f4f193c0625",
            "name": "test"
          }
        }
      }
    },
    "expires_at": "2023-09-18T06:38:19.05516Z",
    "expired_at": "0001-01-01T00:00:00Z",
    "votes": [],
    "auto_executable": false,
    "category": "Keys",
    "operation_label": "Delete Keys"
  },
  {
    "id": "fd719292-ce5e-4314-8adc-171bf0d2a613",
    "uri": "kylo:kylo:sallyport:quorum:fd719292-ce5e-4314-8adc-171bf0d2a6
13",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2023-09-11T11:08:18.968876Z",
    "updatedAt": "2023-09-11T11:08:19.365663Z",
    "requester_id": "local|a6ae00b0-2499-4f61-b5e0-b3da274d4060",
    "operation_fingerprint": "8126d3efd6db7005c36dfb6888ced8c8845d647c5bd
246ab428405120486de9e",
    "operation": "AddUserToGroup",
    "requester_username": "admin",
    "resource_uri": "admin",
    "state": "active",
    "required_approvals": 1,
    "meta": {
      "details": {
        "user": {
          "type": "user_link",
          "value": {
            "id": "local|f1bd14e1-ec8e-4955-8b16-03f47e73d56e",
            "name": "test1"
          }
        },
        "group": {
          "type": "group_link",
          "value": {
            "name": "admin"
          }
        }
      }
    }
  },

```

```

    "expires_at": "2023-09-18T11:08:19.365407Z",
    "expired_at": "0001-01-01T00:00:00Z",
    "votes": [],
    "auto_executable": true,
    "category": "Access Management",
    "operation_label": "Add Users To Groups"
  }
]
}

```

Getting Details of Quorums

To get details of a quorum, run:

Syntax

```
ksctl quorum get --id <QuorumId>
```

Example Request

```
ksctl quorum get --id fd719292-ce5e-4314-8adc-171bf0d2a613
```

Example Response

```

{
  "id": "fd719292-ce5e-4314-8adc-171bf0d2a613",
  "uri": "kylo:kylo:sallyport:quorum:fd719292-ce5e-4314-8adc-171bf0d2a613",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-09-11T11:08:18.968876Z",
  "updatedAt": "2023-09-11T11:08:19.365663Z",
  "requester_id": "local|a6ae00b0-2499-4f61-b5e0-b3da274d4060",
  "operation_fingerprint": "8126d3efd6db7005c36dfb6888ced8c8845d647c5bd246ab428405120486de9e",
  "operation": "AddUserToGroup",
  "requester_username": "admin",
  "resource_uri": "admin",
  "state": "active",
  "required_approvals": 1,
  "meta": {
    "details": {
      "user": {
        "type": "user_link",
        "value": {
          "id": "local|f1bd14e1-ec8e-4955-8b16-03f47e73d56e",
          "name": "jigs"
        }
      },
      "group": {
        "type": "group_link",
        "value": {
          "name": "admin"
        }
      }
    }
  }
}

```

```

    },
    "expires_at": "2023-09-18T11:08:19.365407Z",
    "expired_at": "0001-01-01T00:00:00Z",
    "votes": [],
    "auto_executable": true,
    "category": "Access Management",
    "operation_label": "Add Users To Groups"
  }

```

Deleting Quorums

To delete a quorum, run:

Syntax

```
ksctl quorum delete --id <QuorumId>
```

Example Request

```
ksctl quorum delete --id 558622f8-bf20-4ded-9f43-e72bfaaf73a5
```

There will be no response if quorum is deleted successfully.

Note

The requester of the operation is the owner of the associated quorum. Only the requester/owner has permission to delete the quorum.

Revoking Vote for Quorums

To revoke your vote from an already approved quorum, run:

Syntax

```
ksctl quorum revoke --id <QuorumId>
```

If the number of approvals falls below the required approvals, the quorum moves back to the `active` state.

Example Request

```
ksctl quorum revoke --id 558622f8-bf20-4ded-9f43-e72bfaaf73a5
```

Example Response

There will be no response if vote is revoked successfully.

GUI

Activating Quorums

A quorum is automatically activated when approval is requested from the GUI while performing the operation on which the quorum is created.

It changes the state of the quorum from `pre-active` to `active`. A quorum can be approved only when it is in the `active` state.

After quorum is active, it is available for the approval process.

Note

The requester of the operation is also the owner of the associated quorum. Only the requester/owner has permission to activate the associated quorum.

Approving Quorums

To approve a quorum:

1. Open the CipherTrust Manager GUI.
2. Go to **Quorums**.
3. Click the **Inbox** tab. It will list all the operations where your approval is requested.
4. Go to the operation you want to approve and click the **Approve Quorum** button.

When all the required approvals are available, quorum moves to the `approved` state. If a quorum is auto-executable, the operation will be automatically executed, else you need to re-initiate the operation. The *Auto Executable* field displays whether a quorum is auto-executable.

Note

The user who created the quorum request is not allowed to vote for that quorum.

Denying Quorums

To deny a quorum:

1. Open the CipherTrust Manager GUI.
2. Go to **Quorums**.
3. Click the **Inbox** tab. It will list all the operations where you are requested as a approver.

4. Go to the operation you want to deny and click the **Reject Quorum** button.

Note

The user who created the quorum request is not allowed to vote for that quorum.

Getting List of Quorums

To get the list of quorums:

1. Open the CipherTrust Manager GUI.
2. Go to **Quorums**. It displays the three tabs:
 - **My Requests** - lists all the quorum requests you have created.
 - **Inbox** - lists all the quorums where you are requested as a approver.
 - **Resolved** - lists all the quorum requests that have been executed, rejected, or expired.

Getting Details of Quorums

To get details of a quorum:

1. Open the CipherTrust Manager GUI.
2. Go to **Quorums**. It displays the three tabs:
 - **My Requests** - lists the quorum requests you have created.
 - **Inbox** - lists all the quorums where you are requested as a approver.
 - **Resolved** - lists the quorum requests that have been executed, rejected, or expired.

Expand the drop-down arrow adjacent to operation to get the details of a respective quorum.

Deleting Quorums

To delete a quorum:

1. Open the CipherTrust Manager GUI.
2. Go to **Quorums**.
3. Click the **My Requests** tab. It will list all the quorums you have created.
4. Go to desired quorum and click the Delete button.

Note

The requester of the operation is the owner of the associated quorum. Only the requester/owner has permission to delete the quorum.

Revoking Vote for Quorums

To revoke your vote from an already approved quorum, run:

1. Open the CipherTrust Manager GUI.
2. Go to **Quorums**.
3. Click the **Inbox** tab. It will list all the operations where you are requested as a approver.
4. Go to the operation click the **Revoke Quorum Vote** button to revoke your vote.

If the number of approvals falls below the required approvals, the quorum moves back to the `active` state.

Managing Quorum Profiles using ksctl

The following operations can be performed:

- Get quorum profile
- List/Search quorum profiles
- Update quorum profile

Getting Details of the Quorum Profile

To get the details of quorum profile, run:

Syntax

```
ksctl quorum-profiles get --profile-id <ProfileID>
```

Example Request

```
ksctl quorum-profiles get --profile-id "0e75ff2e-0d76-4681-98d3-cb3e42d5a4de"
```

Example Response

```
{
  "id": "0e75ff2e-0d76-4681-98d3-cb3e42d5a4de",
  "uri": "kylo:kylo:sallyport:quorum-profile:updatekey-requires-quorum-0e75ff2"
```

```
e-0d76-4681-98d3-cb3e42d5a4de",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-05-31T10:11:38.038906Z",
  "updatedAt": "2023-06-05T04:19:00.095912Z",
  "name": "UpdateKey requires quorum",
  "required_approvals": 3,
  "voter_groups": [
    "Key Admins"
  ],
  "excluded_groups": [
    "Key Users"
  ]
}
```

Getting List of Quorum Profiles

To get the list of all quorum profiles, run:

Syntax

```
ksctl quorum-profiles list
```

Example Request

```
ksctl quorum-profiles list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 28,
  "resources": [
    {
      "id": "bf16fb8e-1e19-4875-9c78-7f2dd50a0a2d",
      "uri": "kylo:kylo:sallyport:quorum-profile:updateprofilecte-requires-quorum-bf16fb8e-1e19-4875-9c78-7f2dd50a0a2d",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-05-31T10:11:38.000799Z",
      "updatedAt": "2023-05-31T10:11:38.000197Z",
      "name": "UpdateProfileCTE requires quorum",
      "required_approvals": 3,
      "voter_groups": [
        "CTE Admins"
      ],
      "excluded_groups": null
    },
    {
      "id": "f929da9f-dc19-4c1f-871c-0f69fe5a3371",
      "uri": "kylo:kylo:sallyport:quorum-profile:updatepolicyelementscte-requires-quorum-f929da9f-dc19-4c1f-871c-0f69fe5a3371",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-05-31T10:11:38.014439Z",
      "updatedAt": "2023-05-31T10:11:38.013891Z",
```

```

        "name": "UpdatePolicyElementsCTE requires quorum",
        "required_approvals": 3,
        "voter_groups": [
            "CTE Admins"
        ],
        "excluded_groups": null
    },
    {
        "id": "cc8def90-b3fe-42e0-bc2c-bc1ea7be6eee",
        "uri": "kylo:kylo:sallyport:quorum-profile:updatepolicycte-requires-q
uorum-cc8def90-b3fe-42e0-bc2c-bc1ea7be6eee",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2023-05-31T10:11:37.950185Z",
        "updatedAt": "2023-05-31T10:11:37.949144Z",
        "name": "UpdatePolicyCTE requires quorum",
        "required_approvals": 3,
        "voter_groups": [
            "CTE Admins"
        ],
        "excluded_groups": null
    },
    {
        "id": "0e75ff2e-0d76-4681-98d3-cb3e42d5a4de",
        "uri": "kylo:kylo:sallyport:quorum-profile:updatekey-requires-quoru
m-0e75ff2e-0d76-4681-98d3-cb3e42d5a4de",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2023-05-31T10:11:38.038906Z",
        "updatedAt": "2023-05-31T10:11:38.038199Z",
        "name": "UpdateKey requires quorum",
        "required_approvals": 3,
        "voter_groups": [
            "Key Admins"
        ],
        "excluded_groups": null
    },
    {
        "id": "308ff9fd-ce83-4bef-94e6-2e6f2e9cb88e",
        "uri": "kylo:kylo:sallyport:quorum-profile:updateguardpointcte-requir
es-quorum-308ff9fd-ce83-4bef-94e6-2e6f2e9cb88e",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2023-05-31T10:11:37.976519Z",
        "updatedAt": "2023-05-31T10:11:37.970618Z",
        "name": "UpdateGuardPointCTE requires quorum",
        "required_approvals": 3,
        "voter_groups": [
            "CTE Admins"
        ],
        "excluded_groups": null
    },
    {
        "id": "b0e0c715-f931-46fd-81dd-254a024ad88e",
        "uri": "kylo:kylo:sallyport:quorum-profile:updatecsistoragegroupcte-r
equires-quorum-b0e0c715-f931-46fd-81dd-254a024ad88e",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2023-05-31T10:11:37.953287Z",
        "updatedAt": "2023-05-31T10:11:37.952172Z",
        "name": "UpdateCSISStorageGroupCTE requires quorum",
        "required_approvals": 3,
        "voter_groups": [

```

```

        "CTE Admins"
    ],
    "excluded_groups": null
},
{
    "id": "6a83aadf-b9a6-4cf4-b351-68719257d6ff",
    "uri": "kylo:kylo:sallyport:quorum-profile:updateclientgroupcte-requi
res-quorum-6a83aadf-b9a6-4cf4-b351-68719257d6ff",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2023-05-31T10:11:37.968485Z",
    "updatedAt": "2023-05-31T10:11:37.967915Z",
    "name": "UpdateClientGroupCTE requires quorum",
    "required_approvals": 3,
    "voter_groups": [
        "CTE Admins"
    ],
    "excluded_groups": null
},
{
    "id": "fdccelc4-2c89-479a-ad75-51e5e95a70fc",
    "uri": "kylo:kylo:sallyport:quorum-profile:restorebackup-requires-quo
rum-fdccelc4-2c89-479a-ad75-51e5e95a70fc",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2023-05-31T10:11:38.026042Z",
    "updatedAt": "2023-05-31T10:11:38.018999Z",
    "name": "RestoreBackup requires quorum",
    "required_approvals": 3,
    "voter_groups": [
        "Restore Admins"
    ],
    "excluded_groups": null
},
{
    "id": "02e0f0c6-5071-4b08-8296-de89c5e8aa08",
    "uri": "kylo:kylo:sallyport:quorum-profile:restorebackupdomain-requir
es-quorum-02e0f0c6-5071-4b08-8296-de89c5e8aa08",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2023-05-31T10:11:38.005456Z",
    "updatedAt": "2023-05-31T10:11:38.004807Z",
    "name": "RestoreBackupDomain requires quorum",
    "required_approvals": 3,
    "voter_groups": [
        "Domain Restore Admins"
    ],
    "excluded_groups": null
},
{
    "id": "bd52de83-6611-4aaa-a9f3-bd344c66923d",
    "uri": "kylo:kylo:sallyport:quorum-profile:managepolicyattachment-req
uires-quorum-bd52de83-6611-4aaa-a9f3-bd344c66923d",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2023-05-31T10:11:38.007084Z",
    "updatedAt": "2023-05-31T10:11:38.006579Z",
    "name": "ManagePolicyAttachment requires quorum",
    "required_approvals": 3,
    "voter_groups": [
        "admin"
    ],
    "excluded_groups": null
}

```

```
    }
  ]
}
```

Updating Quorum Profile

You can update the quorum profile for the following fields:

- `approvals` - the number of approvals (votes) required to move the quorum to the Approved state.
- `voter-groups` - the groups defined in the voter-groups field are allowed to vote for this profile. The users part of these groups can either approve or deny a quorum request.
- `excluded-groups` - the groups defined in the excluded-groups field are excluded from being a part of the quorum for the operation specified in the quorum profile. The quorum policies will not apply to all the users associated with the groups specified in the excluded_groups field. Let's take a scenario where the "group1" group is added to the list of excluded-groups. If user "user1" is part of a group "group1", then quorum policies won't apply to the "user1".

To update the quorum profile, run:

Syntax

```
ksctl quorum-profiles update --profile-id <ProfileID> --approvals <NumberOfApprovals> --voter-groups <VoterGroups> --excluded-groups <List-of-Excluded-Groups>
```

Example Request 1

```
ksctl quorum-profiles update --profile-id 61af169c-36b7-42b8-b396-284b92a52613 --approvals 2
```

Example Response 1 (Update for approvals)

```
{
  "id": "61af169c-36b7-42b8-b396-284b92a52613",
  "uri": "kylo:kylo:sallyport:quorum-profile:managepolicy-requires-quorum-61af169c-36b7-42b8-b396-284b92a52613",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-08-20T05:15:50.329875Z",
  "updatedAt": "2021-08-20T05:19:32.613434Z",
  "name": "ManagePolicy requires quorum",
  "required_approvals": 2,
  "voter_groups": [
    "admin"
  ]
}
```

In the above example, two (2) approvals are required to move the quorum to the Approved state.

Example Request 2 (Update for Voter groups)

```
ksctl quorum-profiles update --profile-id 61af169c-36b7-42b8-b396-284b92a52613 --
approvals 2 --voter-groups 'Domain Admins,Key Admins'
```

Example Response 2

```
{
  "id": "61af169c-36b7-42b8-b396-284b92a52613",
  "uri": "kylo:kylo:sallyport:quorum-profile:managepolicy-requires-quorum-61af1
69c-36b7-42b8-b396-284b92a52613",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-08-20T05:15:50.329875Z",
  "updatedAt": "2021-08-20T05:19:32.613434Z",
  "name": "ManagePolicy requires quorum",
  "required_approvals": 2,
  "voter_groups": [
    "Domain Admins"
    "Key Admins"
  ]
}
```

In the above example, only the "Domain Admins" and "Key Admins" groups are allowed to vote for this profile.

Example Request 3 (Update for excluded groups)

```
ksctl quorum-profiles update --profile-id "0e75ff2e-0d76-4681-98d3-cb3e42d5a4de"
--excluded-groups "Key Users"
```

Example Response 3

```
{
  "id": "0e75ff2e-0d76-4681-98d3-cb3e42d5a4de",
  "uri": "kylo:kylo:sallyport:quorum-profile:updatekey-requires-quorum-0e75ff2
e-0d76-4681-98d3-cb3e42d5a4de",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-05-31T10:11:38.038906Z",
  "updatedAt": "2023-06-05T04:19:00.095912Z",
  "name": "UpdateKey requires quorum",
  "required_approvals": 3,
  "voter_groups": [
    "Key Admins"
  ],
  "excluded_groups": [
    "Key Users"
  ]
}
```

In the above example, the users added to the "Key Users" groups are excluded from being a part of the quorum for the "UpdateKey requires quorum" operation.

SNMP

Overview

Simple Network Management Protocol (SNMP) is a request-response protocol used to communicate management information. SNMP enables network and system administrators to remotely monitor devices on the network, such as switches, routers, proxies, and hubs.

This protocol relies on three main concepts: Network Management Station (NMS), agent, and Management Information Base (MIB). The NMS is configured on a network node and runs SNMP management software; agents run on network devices that are being monitored by the NMS; and the MIB defines what kind of information can be exchanged between the agent and the NMS. You can configure an SNMP agent on CipherTrust Manager to monitor the health of the appliance.

Management information is communicated between an Network Management Station (NMS) and an agent. SNMP notifications (traps and informs), sent from agents to managers, might indicate a warning or error condition or otherwise notify the manager about the agent's state.

There are three versions of SNMP: SNMPv1, SNMPv2c and SNMPv3. SNMPv1 and SNMPv2c use community-based security model and SNMPv3 uses User-based Security Model (USM).

SNMP on CipherTrust Manager

CipherTrust Manager supports SNMP versions v1, v2c and v3. You must first [add an SNMP interface](#) before you can configure an SNMP agent.

SNMP v1/v2c

SNMPv1/v2c relies on the community name to provide a low level of security for communications between the NMS and agent. A pairing of an SNMP agent with some arbitrary set of SNMP application entities is called an SNMP community. Each SNMP community is named by a string of octets that is called the community name for said community. For an agent, a subset of objects in the MIB that pertain to that element is called a SNMP MIB view. Community names are associated with a certain access level (read-only, read-write) to SNMP MIB view.

When the CipherTrust Manager receives a SNMPv1/v2c request, the agent verifies that the community name in the request and can optionally validate the source IP of the request matches the community name. The request will be processed if a match is found, otherwise it is dropped. To configure the CipherTrust Manager to respond to SNMP v1/v2c requests, see [Configuring SNMPv1/v2c](#). You configure a community on CipherTrust Manager as part of this process.

To configure CipherTrust Manager's agent to send SNMP v1/v2c trap and inform notifications, you [add a management station](#) independently from configuring community strings to respond to SNMP v1/v2c GET, GETNEXT, GETBULK, and SET requests.

Both SNMPv1 and SNMPv2c are enabled when either community names or SNMPv1/v2c management stations are configured. Deleting all community names and all SNMPv1/v2 management stations is equivalent to disabling SNMP v1/v2c and stopping all future notifications through SNMPv1/v2c.

SNMP v3

SNMPv3 incorporated all the capabilities of SNMPv1/v2c, and introduced the concept of a User-based Security Model (USM). This security model supports data integrity, data origin authentication, data confidentiality and message timeliness and limited replay protection. Additionally, SNMPv3 enhanced the existing access control with View-based Access Control Model (VACM). CipherTrust Manager supports MD5, SHA, SHA-224, SHA-256,

SHA-384, and SHA-512 for authentication protocols; and DES, AES, AES-192, and AES-256 for privacy protocols. To configure the device for SNMP v3, see [configuring SNMPv3](#).

MIBs

CipherTrust Manager implements two MIBs access groups, **Standard** and **Enterprise**. These access groups are applied at the SNMP community or SNMP user level.

Note

There are no MIBs for CipherTrust Manager-specific resources. All MIBS are for hardware and network events.

CipherTrust Manager implements the following Standard MIBs:

- MIB-2 General network statistics (RFC 1213 and subsequent revisions)
- Host Resources MIB (RFC 1514 and 2790)
- SNMPv3 framework (RFCs 2571-2575, RFCs 3411-3418)

This framework is internal only, to manage SNMP users on CipherTrust Manager.

CipherTrust Manager implements the following Enterprise MIBs:

- DisMan Events and Schedule MIBS
- private UC Davis/Net-SNMP agent extensions

This includes specified processes and disks, memory, CPU, and load average.

The [README.agent-mibs for Net-SNMP](#) includes details about included OIDs.

Access Control

Structure of Management Information (SMI & SMIv2) defines ACCESS(SMIv1)/MAX-ACCESS(SMIv2) clause to allow which SNMP operations (GET, SET or Notification) are allowed for a MIB object. An ACCESS value of 'read-write' means that the object can be both read with GET or GET-NEXT request and written with a SET request, while an ACCESS value of 'not-accessible' means that the object can neither be read nor written to. In addition, access control in the CipherTrust Manager SNMP implementation makes it possible for agents to provide different levels of MIB group access to different managers. Access levels 'read-only' or 'read-write' can be provided to Standard and Enterprise MIB access groups. You can restrict access by allowing one NMS to read only standard MIBs and another NMS to read and write both standard MIBs and enterprise MIBs.

Notifications

The CipherTrust Manager can generate and send notifications (traps, informs or both) to configured management stations when an important system events occur such as Authentication failure, Interface link up/down, and DisMan events.

- **Trap:** Traps are unreliable notification message sent from a device to management station.
- **Inform:** Informs are reliable notification message sent from a device to management station that also requires an acknowledgment from the management station.

Traps/Notifications	Description	Example Trap
Authentication Failure	SNMP requests (GET, GET-NEXT, GET-BULK, SET) authentication failed	DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (27499) 0:04:34.99 SNMPv2-MIB::snmpTrapOID.0 = OID: SNMPv2-MIB::authenticationFailure SNMPv2-MIB::snmpTrapEnterprise.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
Link Up/Down Notifications	Communication link state change	Link Up Trap: DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (6033) 0:01:00.33 SNMPv2-MIB::snmpTrapOID.0 = OID: IF-MIB::linkUp IF-MIB::ifIndex.4 = INTEGER: 4 IF-MIB::ifAdminStatus.4 = INTEGER: up(1) IF-MIB::ifOperStatus.4 = INTEGER: up(1) SNMPv2-MIB::snmpTrapEnterprise.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10 Link Down Trap: DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (6033) 0:01:00.33 SNMPv2-MIB::snmpTrapOID.0 = OID: IF-MIB::linkDown IF-MIB::ifIndex.21 = INTEGER: 21 IF-MIB::ifAdminStatus.21 = INTEGER: down(2) IF-MIB::ifOperStatus.21 = INTEGER: down(2) SNMPv2-MIB::snmpTrapEnterprise.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
Temperature Notifications	Temperature sensors exceeds 68 degrees Centigrade	DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (128) 0:00:01.28 SNMPv2-MIB::snmpTrapOID.0 = OID: DISMAN-

Traps/Notifications	Description	Example Trap
		EVENTMIB::mteTriggerFired DISMAN- EVENTMIB::mteHotTrigger.0 = STRING: Temperature above threshold DISMAN- EVENTMIB::mteHotTargetName.0 = STRING: DISMAN- EVENT- MIB::mteHotContextName.0 = STRING: DISMAN- EVENTMIB::mteHotOID.0 = OID: LM- SENSORSMB::lmTempSensors Value.1 DISMANEVENT- MIB::mteHotValue.0 = INTEGER: 70000 SENSORSMB::lmTempSensors Device.1 = STRING: Core 0
Free Memory Notifications	Total memory available is less than 1 GB	DISMAN-EVENT- MIB::sysUpTimeInstance = Timeticks: (147) 0:00:01.47 SNMPv2-MIB::snmpTrapOID.0 = OID: DISMAN-EVENT- MIB::mteTriggerFired DISMAN-EVENT- MIB::mteHotTrigger.0 = STRING: MemFreeTotal DISMAN-EVENT- MIB::mteHotTargetName.0 = STRING: DISMAN-EVENT- MIB::mteHotContextName.0 = STRING: DISMAN-EVENT- MIB::mteHotOID.0 = OID: UCD-SNMP- MIB::memTotalFree.0 DISMAN-EVENT- MIB::mteHotValue.0 = INTEGER: 185952 UCD-SNMP- MIB::memTotalReal.0 = INTEGER: 16414160 kB
Processor Load Average Notifications	Load average exceeds: 30.00 in 1 minute, 10.00 in 5 minutes, 5.00 in 15 minutes.	Trap for 1 minute load average: mteHotTrigger= laTable mteHotTargetName = mteHotContextName = mteHotOID= .1.3.6.1.4.1.2021.10.1.10 0.1 mteHotValue = 0 \$6 = Load-1 \$7 = Trap for 5 minute load average: mteHotTrigger= laTable

Traps/Notifications	Description	Example Trap
		<pre> mteHotTargetName = metHotContextName = mteHotOID= .1.3.6.1.4.1.2021.10.1.10 0.1 mteHotValue = 0 \$6 = Load-5 \$7 = Trap for 15 minute load average: mteHotTrigger= laTable mteHotTargetName = metHotContextName = mteHotOID= .1.3.6.1.4.1.2021.10.1.10 0.1 mteHotValue = 0 \$6 = Load-15 \$7 = The OID is in UCD-SNMP-MIB. </pre>
Disk Utilization Notifications	Disk usage exceeds 80%	<pre> mteHotTrigger = dskTable mteHotTargetName = mteHotContextName = mteHotOID = .1.3.6.1.4.1.2021.9.1.100 .1 mteHotValue = 1 \$6 = / \$7 = / less than 20% free (=10%) The OID is in UCD-SNMP-MIB. </pre>

To receive SNMP notifications (traps and informs) from CipherTrust Manager, the [management station](#) needs to be configured.

Configuring SNMPv1/v2c

You must first [add an SNMP interface](#) before you can configure an SNMP agent.

To monitor and control the CipherTrust Manager using SNMPv1/v2c, community names must be configured. Once a community is configured, the SNMP agent starts responding to GET, GETNEXT, GETBULK, and SET requests from allowed management stations.

Configure Communities

Configure (add, read, update and delete) community names with management station host(s) filtering for SNMP v1 and SNMPv2c.

To add a community configuration

Add a SNMP community configuration with management station host(s) filtering and mib access control.

```
ksctl snmp communities add -community <community-name> --source <mgmt.-station-i
p-or-cidr> --mib-accesses <standard,enterprise>
```

Example 1:

Allow any management station to access standard MIB view using the community name 'public'

```
ksctl snmp communities add -community public
```

Example 2:

Allow a management station with IP '10.10.10.10' to access standard and enterprise MIB views using the community name 'private'.

```
ksctl snmp communities add -community private -source 10.10.10.10 -mib-accesses  
"standard,enterprise"
```

Example 3:

Allow a group of management stations using CIDR notation to access enterprise MIB view using the community name 'secret'

```
ksctl snmp communities add -community secret -source 10.10.10.0/24 -mib-accesses  
enterprise
```

To read a community configuration

To read a SNMP community configuration using its ID.

```
ksctl snmp communities get --id <community-id>
```

To list community configurations

To list all the SNMP community configurations.

```
ksctl snmp communities list
```

To delete community configuration

To delete a SNMP community configuration using its ID.

```
ksctl snmp communities delete --id <community-id>
```

Configuring SNMPv3

You must first [add an SNMP interface](#) before you can configure an SNMP agent.

To monitor and control the CipherTrust Manager and to receive notifications using SNMPv3, users must be configured.

Configuring SNMP USM Users

Configure (add, read, update and delete) SNMP USM users for management and to use with SNMP management station (notifications).

Once you add at least one SNMP USM user, the SNMP agent starts responding to GET, GETNEXT, GETBULK, and SET requests from allowed management stations. You can then [create an SNMP v3 management station](#) to start sending SNMP trap and inform notifications.

To add a SNMP USM user

Example 1:

To add a SNMP USM user configuration with no authentication and no privacy, and with read-only access to standard MIB view. No authentication and no privacy configuration is not recommended.

```
ksctl snmp users add --user-name rob --security-level noAuthNoPriv --mib-accesses standard
```

Example 2:

To add a SNMP USM user configuration with authentication but no privacy. It gives read-write access to standard and enterprise MIBs views.

```
ksctl snmp users add --user-name bob --security-level authNoPriv --auth-protocol SHA --auth-password --mib-accesses "standard,enterprise" --read-write
```

Example 3:

To add a SNMP USM user configuration for SNMPv3 notification with authentication and privacy. There is no MIB access provided for this user.

```
ksctl snmp users add --user-name trapuser --security-level authPriv --auth-protocol SHA --auth-password authpass --priv-protocol AES --priv-password privpass
```

Example 4:

To add a SNMP USM user configuration for inform notification with authentication and privacy. There is no MIB access provided for this user. EngineID of the notification receiver is required.

```
ksctl snmp users add --user-name informuser --engine-id 0x0102030405 --security-level authPriv --auth-protocol SHA --auth-password authpass --priv-protocol AES --priv-password privpass
```

Adding an SNMP Management Station

Add an SNMP management station configuration to start sending SNMP trap or inform notifications to a management station.

To add a SNMP v3 management station, you need at least one [SNMP USM user configured](#). If this user is used only for notification, then it can be configured without any access groups.

Example 1:

To send SNMPv1 trap using community name 'private'

```
ksctl snmp mgmt-stations add --host 1.1.1.1 --version 1 --security-name private
```

Example 2:

To send SNMPv2c inform using community name 'secret'

```
ksctl snmp mgmt-stations add --notification-type inform --host 1.1.1.2 --version 2c --security-name secret
```

Example 3:

To send SNMPv3 trap using USM user name 'trapuser' that is configured through 'ksctl snmp users' command.

```
ksctl snmp mgmt-stations add --notification-type trap --host 1.1.1.3 --port 1162 --version 3 --security-name trapuser
```

SMTP

SMTP stands for Simple Mail Transfer Protocol. This protocol facilitates the flow of emails over the Internet.

You can configure CipherTrust Manager to automatically send email notifications for [record-based alarms](#) to a set of email addresses, by configuring an SMTP server.

Note

Email notifications will be sent for built-in alarms such as Disk Full, Cluster Node Down, and NAE TLS Disabled.

If an SMTP server is present, CipherTrust Manager can also email password expiry reminders to users. Password expiry reminders are configured in user [password policies](#). SMTP retrieves email addresses for this purpose from [user information](#).

Note

CipherTrust Manager supports both TLS and TCP based SMTP connection. Default is TLS.

Adding a New SMTP Server

To add a new SMTP server:

1. Log on to the CipherTrust Manager console as administrator.
2. Go to **Admin Settings > SMTP**.
3. Click **New SMTP Server**. The **Configure a SMTP Server** dialog box is displayed.
4. Specify the following details:
 - **Hostname or IP Address**: Specify the hostname or IP address of the SMTP server.

Note

If using a **Hostname**, make sure that it is added to the CipherTrust Manager. To read more, go to [Configuring DNS Hosts](#).

- **Port**: Specify the port of the SMTP server.
- **Allow TCP**: Select the check box to allow TCP connection with SMTP server. The default value is false.

Note

When both TLS and TCP (allow TCP is true) connections are available with SMTP server, then by default TLS connection will be used.

- **Username**: Specify the Username for the SMTP server.
 - **Sender's Email**: Specify the email address from where the email is to be sent.
 - **Password**: Specify the Password for the SMTP server's Username.
5. Click **Save**. The SMTP server is configured. The server details will appear in the SMTP Server Settings section.

Note

You can configure only one SMTP server at a time. If you add a new SMTP server, the new server will replace the older SMTP server configuration.

Tip

After configuring the SMTP server, click **Test SMTP Connection** button to test the connection with the SMTP server. If the server is configured correctly, the Sender's email address will receive a test email.

Note

After updating the SMTP settings, wait for sometime (approx. 3-5 minutes) for the changes to be effective.

Adding/Deleting Emails in the Notification Address Lists

Note

Notification address lists are used to send emails for [record-based alarms](#). These lists are not used to send emails for password expiry reminders. SMTP retrieves email addresses for these password expiry reminders from [user information](#).

To view, add, or delete the email addresses that will receive the email notifications:

1. Log on to the CipherTrust Manager console as administrator.
2. Go to **Admin Settings > SMTP**.

The list of notification email address will be displayed under **Email Notification Settings** section.

- a. To add a new email address, click **Add Email**. Specify the desired email address and click **Save**.
- b. To delete an existing email address, click **Delete**.

Managing SMTP Servers using ksctl

The following operations can be performed:

- Add SMTP Servers
- List all SMTP Servers
- Delete SMTP Servers

Adding SMTP Servers

To add a SMTP server, run:

Syntax

```
ksctl notification smtp-servers add --server <Server-Name> --port <Port> --username <Username> --emailfrom <Email-Sender> --smtp pw <Password> --allow-tcp
```

Example Request

```
ksctl notification smtp-servers add --server smtp.gmail.com --port 587 --username abc@xyz.com --emailfrom abc@xyz.com --smtp pw admin123 --allow-tcp
```

Example Response

```
{
  "id": "404a8530-4d49-40bc-a372-8ccd496e1ad3",
  "uri": "kylo:kylo:notification:smtp_server:404a8530-4d49-40bc-a372-8ccd496e1ad3",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-02-18T06:35:40.542578Z",
  "server": "smtp.gmail.com",
  "port": 587,
  "username": "abc@xyz.com",
  "email_from": "abc@xyz.com",
  "allow_tcp": true,
  "updatedAt": "2021-02-18T06:36:10.154286327Z"
}
```

Getting List of SMTP Servers

To get a list of SMTP servers, run:

Syntax

```
ksctl notification smtp-servers list
```

Example Request

```
ksctl notification smtp-servers list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "404a8530-4d49-40bc-a372-8ccd496e1ad3",
      "uri": "kylo:kylo:notification:smtp_server:404a8530-4d49-40bc-a372-8c
cd496e1ad3",
      "account": "kylo:kylo:admin:accounts:kylo",
      "application": "ncryptify:gemalto:admin:apps:kylo",
      "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
      "createdAt": "2021-02-18T06:35:40.542578Z",
      "server": "smtp.gmail.com",
      "port": 587,
      "username": "abc@xyz.com",
      "email_from": "abc@xyz.com",
      "allow_tcp": true,
      "updatedAt": "2021-02-18T06:36:10.154286Z"
    }
  ]
}
```

Deleting SMTP Servers

To delete a SMTP server, run:

Syntax

```
ksctl notification smtp-servers delete --id
```

Example Request

```
ksctl notification smtp-servers delete --id 404a8530-4d49-40bc-a372-8ccd496e1ad3
```

There will be no response if SMTP server is deleted successfully.

Managing Email Notification Settings using ksctl

The following operations can be performed:

- Add email address to notification list
- Get details of email addresses from notification list
- Fetch email addresses from notification list
- Delete email addresses from notification list

Adding Email Address to Notification List

To add an email address to the notification list, run:

Syntax

```
ksctl notification email add --emailaddr <Email-Address> --meta <String/JSON-File>
```

Example Request 1

```
ksctl notification email add --emailaddr abc@xyz.com
```

Example Response 1

```
{
  "id": "104314bc-b39f-4a5c-be78-4c52b875ef4c",
  "uri": "kylo:kylo:notification:email-addresses:104314bc-b39f-4a5c-be78-4c52b875ef4c",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-02-19T09:36:43.419179Z",
  "email_id": "abc@xyz.com"
}
```

Example Request 2

```
ksctl notification email add --emailaddr abc@xyz.com --meta 'admin email address'
```

Example Response 2

```
{
  "id": "37ac2548-5deb-4e5d-bb8e-96a17c447e94",
  "uri": "kylo:kylo:notification:email-addresses:37ac2548-5deb-4e5d-bb8e-96a17c447e94",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-05-27T04:42:00.377132Z",
  "email_id": "abc@xyz.com",
  "meta": "admin email address"
}
```

Example Request 3

```
ksctl notification email add --emailaddr bob@xyz.com --meta-jsonfile emailMeta.js
on
```

Example Response 3

```
{
  "id": "f53c7e4a-0e9a-44de-839b-5142787e3b6a",
  "uri": "kylo:kylo:notification:email-addresses:f53c7e4a-0e9a-44de-839b-514278
7e3b6a",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-05-27T04:46:47.7163Z",
  "email_id": "bob@xyz.com",
  "meta": {
    "name": "bob"
  }
}
```

Note

Meta field is used to provide additional information (for example, notes/description) that helps to manage email notifications. You can pass meta information through String or JSON file.

Getting Details of Email Addresses from Notification List

To get details of an email address from the notification list, run:

Syntax

```
ksctl notification email get --id <Resource-ID>
```

Example Request

```
ksctl notification email get --id 104314bc-b39f-4a5c-be78-4c52b875ef4c
```

Example Response

```
{
  "id": "104314bc-b39f-4a5c-be78-4c52b875ef4c",
  "uri": "kylo:kylo:notification:email-addresses:104314bc-b39f-4a5c-be78-4c52b8
75ef4c",
  "account": "kylo:kylo:admin:accounts:kylo",
}
```

```
{
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2021-02-19T09:36:43.419179Z",
  "email_id": "abc@xyz.com"
}
```

Fetching Email Addresses from Notification List

To fetch email addresses from the notification list, run:

Syntax

```
ksctl notification email list
```

Example Request

```
ksctl notification email list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "104314bc-b39f-4a5c-be78-4c52b875ef4c",
      "uri": "kylo:kylo:notification:email-addresses:104314bc-b39f-4a5c-be78-4c52b875ef4c",
      "account": "kylo:kylo:admin:accounts:kylo",
      "application": "ncryptify:gemalto:admin:apps:kylo",
      "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
      "createdAt": "2021-02-19T09:36:43.419179Z",
      "email_id": "abc@xyz.com"
    }
  ]
}
```

Deleting Email Addresses from Notification List

To delete an email address from the notification list, run:

Syntax

```
ksctl notification email delete --id <Resource-ID>
```

Example Request

```
ksctl notification email delete --id 104314bc-b39f-4a5c-be78-4c52b875ef4c
```

There will be no response if notification email is deleted successfully.

Backups

Backups are generated on the appliance and are stored there until deleted. They can be downloaded from the appliance and uploaded back to the same or to another appliance. When a backup is created, it is always encrypted with a backup key. If a backup key is not specified, the default backup key is used to encrypt the backup. This backup key is required to restore the backup file.

Backups and backup keys are not automatically copied to other nodes in a cluster. They only exist on a node where they are created or copied to. Backups are treated as resources and are controlled through the API, ksctl, and GUI.

CipherTrust Manager supports system backups and domain scoped backups.

Note

Backup jobs can also be scheduled to run periodically in the background. Refer to [Scheduling Backups](#) for details.

Release compatibility

This compatibility applies to both system and domain backups.

By default, the CipherTrust Manager backups can be restored on the next *three* minor versions. For example, a backup of release 2.11 can be restored to 2.12, 2.13, and 2.14 instances. A `--force` option can override this restriction, allowing for restore of backups taken from three or more earlier minor versions.

For LTS versions, a backup from an LTS release can be restored on the next *three* minor versions, as stated above, and the next *two* LTS releases.

Caution

We recommend that you consult with customer support before restoring backups from *three* or more earlier minor versions. This is a rare scenario required only when an intermediate release might have a bug or limitation.

Backward compatibility is not supported. For example, a backup of release 2.10 cannot be restored to a 2.9 instance.

HSM-anchored domain resources

Resources from every domain are included in a system backup, and domain-scoped backup is available for [HSM-](#)

[anchored domains](#). The contents of an HSM-anchored domain are always encrypted by an HSM partition. This means that you can only restore to a domain or appliance with a connection to the same HSM partition.

System backups

System backups are a snapshot of the database in an appliance at a point in time. They are created on the appliance, and after creation, they can be used to:

- Restore the appliance to a previous state
- Transfer the backup to another appliance to restore on that appliance
- Create a new appliance with the backup

CipherTrust Manager users (such as Application Administrators), who are authorized to perform backup, can perform system backups and restores. Domain administrators are not authorized to perform system backups.

Note

System backups cannot be restored on a cluster node. The cluster Remove/Rebuild method should be used to restore a system backup.

Tying the system backup to an HSM

A system backup can optionally be tied to an HSM for increased protection of the data. This means that the backup can only be restored to an appliance that has the same active [HSM Root of Trust \(RoT\) key](#) as was used when the backup was taken. This feature improves the protection of backup files, but requires careful management of the HSM RoT key used for the backup. Otherwise, data loss is possible.

Unless the appliances share the same HSM partition and the same active HSM RoT key, a backup tied to the HSM cannot be restored to any other appliance. Also, if you manually delete the RoT key, or the HSM is tampered and the RoT key is deleted, a backup tied to that HSM can never be restored.

Rotating the HSM RoT key prevents the appliance from restoring the backup file. You can regain the ability to restore the backup by rotating the original RoT key back to an active state.

Backup Contents

A system backup includes the following:

- All domain resources
- Keys, key attributes, and key links
- Users, groups, and their relationships
- Local Certificate Authorities (CAs)
- External Certificate Authorities (CAs)

-
- Certificates issued by Local CAs, excluding certificates configured for the CipherTrust Manager interfaces
 - Interface configuration, aside from the certificate
 - Policies
 - Login banners
 - CSM customer fragments
 - Audit records

Caution

If local database audit logging is enabled by restoring a backup, system operation performance can degrade. We recommend [disabling local database audit logging](#) before taking a backup.

- Schedules
- Legacy syslog connections and connection manager connections
- LDAP connections
- OIDC connections for managing CipherTrust Manager users.
- Resources of all supported applications: Keys, Access Management, Admin Settings, Data Discovery, ProtectDB, ProtectFile, ProtectApp, and KMIP

The following objects are **excluded** from the backup:

- Backup keys
 - Cluster configuration
 - DNS hosts
 - NTP configuration
 - HSM configuration
 - Instance name
 - Virtual CipherTrust Manager license
 - Interface certificates
 - Debug, KMIP Activity, and NAE Crypto Activity Logs
 - Proxy configuration
-

Domain scoped backups

Domain scoped backup and restore is the ability to back up user-specified resources partially and incrementally from a **domain**. Domain scoped backups are created on the appliance, and after creation, can be used to:

- Restore the resources on the same domain to its previous state
- Restore on a different domain in the same appliance
- Transfer to another appliance to restore on a different domain

In all cases, a destination domain must exist for the backup to be restored to. For a backup taken from an HSM-anchored domain, the destination domain must have a connection to the same HSM partition as was configured for the backed up domain. You can only [add an HSM connection to a domain during creation](#).

CipherTrust Manager users (such as Domain Administrators), who are authorized to perform backup, can perform domain scoped backup and restore. Application Administrators can perform backup and restore operations of domain scoped backup of root domain only, not other sub-domains.

Note

Domain scoped backups can be restored on any clustered node.

Note

Domain scoped backups cannot be tied to an HSM.

Backup contents

Domain scoped backups include the following:

- Keys, key attributes, key links
- CTE policies
- Local users and groups
- Customer fragments
- CCKM resources: Projects, Google EKM Endpoints and Cryptospaces

Note

- The CipherTrust Manager allows you to take backup of the individual keys.
- Trusted Partner Cloud Cryptospace type is not included in the domain backup.

Typical backup and restore operations

The following are typical backup operations using the **ksctl**. If you are using the API, refer to the API playground for details.

1. [Create the backup key](#)
2. [Create the backup](#)
3. [Check the backup status](#)
4. [Download the backup key](#)
5. [Download the backup](#)
6. [Transfer the Backup using Secure Copy Protocol](#)
7. [Inspect the backup key file](#)
8. [Upload the backup key](#)
9. [Inspect the backup file](#)
10. [Upload the backup](#)
11. [Restore the backup](#)
12. [Check the restore status](#)

Create the backup key

Backups are automatically encrypted by a backup key at the time of their creation. If no key is specified, the default backup key is used. If needed, a new backup key can be created and optionally set as the default backup key.

Backup keys can be of two types:

- System scoped keys to encrypt only system backups
- Domain scoped keys to encrypt only domain scoped backups

To create a system backup key

To create a system backup key, run `ksctl backupkeys create`. If not specified, the default scope is `system`.

Example request:

```
ksctl backupkeys create
```

Example response:

```
{
  "scope": "system",
  "account": "kylo:kylo:admin:accounts:kylo",
  "version": "2",
  "id": "5dac1d33-9c2d-4fc1-9793-961052d73a29",
  "createdAt": "2020-04-07T08:17:54.600032599Z",
  "isDefault": false
}
```

To create a system backup key and set it as the *default* key, run `ksctl backupkeys create --scope system --default`.

Example request:

```
ksctl backupkeys create --scope system --default
```

Example response:

```
{
  "scope": "system",
  "account": "kylo:kylo:admin:accounts:kylo",
  "version": "2",
  "id": "c05b638d-2eea-4651-9937-7bee49a95f6b",
  "createdAt": "2020-04-07T08:21:59.139786694Z",
  "isDefault": true
}
```

To create a domain scoped backup key

To create a domain scoped backup key, run `ksctl backupkeys create --scope domain`. Here, `--scope domain` specifies that the key is domain scoped. The scope flag must be used for creating domain scoped backup keys.

Example request:

```
ksctl backupkeys create --scope domain
```

Example response:

```
{
  "scope": "domain",
  "account": "kylo:kylo:admin:accounts:kylo",
  "domainID": "00000000-0000-0000-0000-000000000000",
  "version": "2",
  "id": "88f47a71-a1d7-41ae-af49-736862f4f625",
  "createdAt": "2020-04-07T08:31:29.314569058Z",
  "isDefault": false
}
```

To create a domain scoped backup key and set it as the *default* key, run `ksctl backupkeys create --scope domain --default`.

Example request:

```
ksctl backupkeys create --scope domain --default
```

Example response:

```
{
  "scope": "domain",
  "account": "kylo:kylo:admin:accounts:kylo",
  "domainID": "00000000-0000-0000-0000-000000000000",
  "version": "2",
  "id": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
  "createdAt": "2020-04-07T08:24:13.648719582Z",
  "isDefault": true
}
```

To view the list of available backup keys

Run the command `ksctl backupkeys list`.

Example request:

```
ksctl backupkeys list
```

Example response:

```
{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "scope": "system",
      "account": "kylo:kylo:admin:accounts:kylo",
      "version": "2",
      "id": "c05b638d-2eea-4651-9937-7bee49a95f6b",
      "createdAt": "2020-04-07T08:21:59.139786694Z",
      "isDefault": true
    },
  ],
}
```

```
{
  "scope": "domain",
  "account": "kylo:kylo:admin:accounts:kylo",
  "domainID": "00000000-0000-0000-0000-000000000000",
  "version": "2",
  "id": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
  "createdAt": "2020-04-07T08:24:13.648719582Z",
  "isDefault": true
}
```

Create a backup

It allows you to create system and domain scoped backup using ksctl.

To create a system backup

To create a system backup, run `ksctl backup create`. If an encryption key is not specified when creating a backup, the backup is encrypted by the default backup key. You can check the status of the backup by running the `ksctl backup status` command. Refer to [Check the backup status](#) for details.

Example request:

```
ksctl backup create
```

Example response:

```
{
  "scope": "system",
  "account": "kylo:kylo:admin:accounts:kylo",
  "version": "v1.1.4",
  "id": "245d3336-21fa-4a34-9498-c0b64fa4d322",
  "createdAt": "2020-04-07T08:33:44.687970269Z",
  "status": "In Progress",
  "tiedToHSM": false,
  "backupKey": "c05b638d-2eea-4651-9937-7bee49a95f6b",
  "productVersion": "v1.10.0-beta4.4388"
}
```

When a backup is initiated, its status becomes `"In Progress"`. It may take a while until the status becomes `"Completed"`. When a backup operation is `"In Progress"`, another backup or restore operation cannot be initiated.

To create a system backup by specifying the scope as `system`, run `ksctl backup create --scope system`.

Example request:

```
ksctl backup create --scope system
```

Example response:

```
{
  "scope": "system",
  "account": "kylo:kylo:admin:accounts:kylo",
  "version": "v1.1.4",
  "id": "06c7d57d-2749-4285-8c60-c2778c911cd2",
  "createdAt": "2020-04-07T08:34:49.914616905Z",
  "status": "In Progress",
  "tiedToHSM": false,
  "backupKey": "c05b638d-2eea-4651-9937-7bee49a95f6b",
  "productVersion": "v1.10.0-beta4.4388"
}
```

Warning

A system level backup can optionally be tied to the HSM during creation by passing the flag `--tied-to-hsm`. This option is invalid for domain scoped backups. This feature requires careful management of the HSM RoT Key and can result in loss of data; see [Tying the System Backup to an HSM](#) for important details. To initiate a system level backup tied to the HSM, run the command:

```
$ ksctl backup create --scope system --tied-to-hsm --description
<description>
```

To create a domain scoped backup

When creating a domain scoped backup, specify the scope as `domain`. To create a domain scoped backup, run `ksctl backup create --scope domain`. You can check the status of the backup by running the `ksctl backup status` command. Refer to [check the backup status](#) for details.

Here, `--scope domain` specifies that the backup is domain scoped. The scope flag must be used for creating domain scoped backups.

Note

To perform the backup operation in a sub-domain, you need to log on to the domain or use the `--domain` flag.

Example request:

```
ksctl backup create --scope domain
```

Example response:

```
{
  "scope": "domain",
  "account": "kylo:kylo:admin:accounts:kylo",
  "domainID": "00000000-0000-0000-0000-000000000000",
  "version": "v1.1.4",
}
```

```
{
  "id": "762d15f3-4eba-4cd7-960c-c50a84a52b48",
  "createdAt": "2020-04-07T08:36:34.667280981Z",
  "status": "In Progress",
  "tiedToHSM": false,
  "backupKey": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
  "productVersion": "v1.10.0-beta4.4388"
}
```

Note

When creating a **domain** scoped backup, use filters to select resources. Filters specify selection criteria to include only certain resources in the backup from the requester's domain. Filter criteria include the resource type and its matching attribute values in JSON format for querying the resources.

Note

Currently, partial domain backup and restore of CCKM resources (Projects, Cryptospaces, EKM Endpoint) are *not* supported.

Filters apply to domain scoped backups only.

To use filters, run either of the following commands:

- `ksctl backup create --scope domain --filters <filters-json>`
- `ksctl backup create --scope domain --filters-jsonfile <file-name-containing-filters-json>`

Examples (Windows):

Note

The examples are provided for the Windows platform. To run these examples on Linux, see [CLI usage with JSON parameters](#).

- Example 1:

To create a domain scoped backup of all CTE keys and CTE policies, run the following command:

```
ksctl backup create --scope domain --filters "[{\"resourceType\": \"Keys\"},{\"resourceType\": \"cte_policies\"}]"
```

- Example 2:

To create a domain scoped backup for all CTE policies (without keys), run the following command:

```
ksctl backup create --scope domain --filters "[{\"resourceType\": \"cte_policies\"}] "
```

- Example 3:

To create a domain scoped backup for only STANDARD CTE policies (without keys), run the following command:

```
ksctl backup create --scope domain --filters "[{\"resourceType\": \"cte_policies\", \"resourceQuery\": {\"policy_type\": \"STANDARD\"}}] "
```

- Example 4:

To create a domain scoped backup for all users and groups, run the following command:

```
ksctl backup create --scope domain --filters "[{\"resourceType\": \"users_groups\"}] "
```

For users, the valid resourceQuery parameter values are the same as query parameters of the list '/v1/usermgmt/users' endpoint. For example, to back up all users with name "frank" and email id "frank@local", use:

```
ksctl backup create --scope domain --filters "[{\"resourceType\": \"users\", \"resourceQuery\": {\"name\": \"frank\", \"email\": \"frank@local\"}}] "
```

- Example 5:

To create a domain scoped backup for individual keys using key names, run the following command:

```
ksctl backup create --scope domain -k f52e26f8-ba8e-4f30-878d-ce88a118091f --filters "[{\"resourceType\": \"Keys\", \"resourceQuery\": { \"names\": [\"key1\", \"key2\"]}}] "
```

Note

The backup and restore of users, groups, and 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.

Note

- The assigned users from the root domain cannot be backed up.
- If the same user exists in the source domain (backup file) and the target domain, the user from the source domain is ignored. However, the corresponding groups of that user are restored. The same user of the target domain will not be part of any restored groups; but, can be added manually, if required.
- If the same group exists in the source domain (backup file) and the target domain, the group from the source domain is not restored. However, the corresponding users of that group are restored. The restored users will not be part of the same group; but can be added manually, if required.

Refer to the API playground for details on filters.

To view the list of available backups

To view the list of available backups, run `ksctl backup list`.

Example request:

```
ksctl backup list
```

Example response:

```
{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "scope": "system",
      "account": "kylo:kylo:admin:accounts:kylo",
      "version": "v1.1.4",
      "id": "06c7d57d-2749-4285-8c60-c2778c911cd2",
      "createdAt": "2020-04-07T08:34:49.914616905Z",
      "status": "Completed",
      "tiedToHSM": false,
      "backupKey": "c05b638d-2eea-4651-9937-7bee49a95f6b",
      "productVersion": "v1.10.0-beta4.4388"
    },
  ],
}
```

```

    "scope": "domain",
    "account": "kylo:kylo:admin:accounts:kylo",
    "domainID": "00000000-0000-0000-0000-000000000000",
    "version": "v1.1.4",
    "id": "762d15f3-4eba-4cd7-960c-c50a84a52b48",
    "createdAt": "2020-04-07T08:36:34.667280981Z",
    "status": "Completed",
    "tiedToHSM": false,
    "backupKey": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
    "productVersion": "v1.10.0-beta4.4388"
  }
]
}

```

To view the details of a backup

To view the details of a backup, run `ksctl backup get --id "backupID"`. The status of the backup is given in the response. A status of `"Completed"` indicates that the backup is successful.

Example request:

```
ksctl backup get --id "762d15f3-4eba-4cd7-960c-c50a84a52b48"
```

Example response:

```

{
  "scope": "domain",
  "account": "kylo:kylo:admin:accounts:kylo",
  "domainID": "00000000-0000-0000-0000-000000000000",
  "version": "v1.1.4",
  "id": "762d15f3-4eba-4cd7-960c-c50a84a52b48",
  "createdAt": "2020-04-07T08:36:34.667280981Z",
  "status": "Completed",
  "tiedToHSM": false,
  "backupKey": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
  "productVersion": "v1.10.0-beta4.4388"
}

```

Check the backup status

Initiating a backup begins an asynchronous operation on the appliance. Only one backup or restore operation can be performed at a time. If you try to perform another operation until the previous is complete, the operation will fail.

To view the status of the last backup, run the command `ksctl backup status`. The status of the backup is given in the response.

Example request:

```
ksctl backup status
```

Example response:

```
{
  "scope": "system",
  "account": "kylo:kylo:admin:accounts:kylo",
  "operation": "Backup create",
  "id": "f6b55640-ddec-49d0-893b-4f33f0809bf0",
  "started": "2020-04-10T00:27:01.078766633Z",
  "status": "In Progress"
}
```

In the output above, the status of the backup is `In Progress`.

Download the backup key

The key with which the backup is encrypted must be transferred to the appliance where the backup will be restored. The backup key is required for the restore. To transfer a backup key, it must be downloaded using the command `ksctl backupkeys download`. After the key is downloaded, it can be uploaded to the appliance to restore the backup.

To download the backup key

Run the command `ksctl backupkeys download --id "<backupkeyID>" --file "<file-to-store-key>"`. The same command is used to the download system scoped and domain scoped backup keys.

Here:

- `"<backupkeyID>"`: ID of the backup key to download.
- `"<file-to-store-key>"`: Name of the file to store the backup key.

Specify a password for the backup key when prompted.

Example - download system backup key:

```
ksctl backupkeys download --id "c05b638d-2eea-4651-9937-7bee49a95f6b" --file "sys
bkkey.txt" Password is required for backupkey. Please provide the backupkey passw
ord:
```

Download the backup

The backup must be transferred to the appliance where it will be restored. To transfer a backup, it must be downloaded by using the command `ksctl backup download`. After the backup is downloaded, it can be uploaded to the appliance for restore.

To download the backup

To download a backup, run the command `ksctl backup download --id "<backupID>" --file "<backup-file>"`.

Here,

- `"<backupID>"`: ID of the backup to download.

- "`<backup-file>`": Name of the file to store the backup.

Example - download system backup:

```
ksctl backup download --id "245d3336-21fa-4a34-9498-c0b64fa4d322" --file "backup-file.txt"
```

Transfer the backup using Secure Copy Protocols (SCP/SFTP)

SCP/SFTP allows you to securely transfer **only** the system backups from the CipherTrust Manager to the external servers.

The following operations can be performed:

- Transfer backup to the destination host machine
- Get backup transfer status

To transfer the backup to the destination host machine

To transfer the backup to the destination host machine, first, you need to retrieve and then rotate the public key of the CipherTrust Manager.

Retrieve the public key

Syntax

```
ksctl scp public-key get
```

Example request

```
ksctl scp public-key get
```

Example response:

```
{
  "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQDHMDnKwJUTBwZsdrF11hni2aCnPob78Wwq3ix8WYI4
  TwAP6DWi0zy2aDlEry62DIVGOMJ1z4hdugfgTfzY45IVf/vG0lMsK+eOtvGErA6A6dCwXvxhX068MnytZ
  Hqt2tR6KV8smHMWec62hAubtQs1wZSAz80JvWDMOSks3OtTRILLtgyaO7n1omAHZLxc6XZ70e2b0sAB03
  oV9PCzI2QJb/KBEdpcA1IEB0Ts4QrXlp3rEzCSYfAU2SW5pIrisSLKyBdu5MQ7esvf2SDcrCL3O2t0FXv
  8A/r0k63tmQNFRfLHOEQ1BjWF/jHntZ4T7Wf5pH0JspSPARVAKLQTP9vTGM0bR28/oisWqC9Dbky5Ks5T
  fCfQKZBQ57mN3b4Z4aU1mb6TQqTCSps8INq9Bk7wzPXBa3mPDpdBmIrsofiDuMBem+PRtuW5RwU28dvt7
  D5HqjD2V8pW8YMKugl0MZwCe6pcPiiH8vaKe0EI/HAKRPOoKBYmLHO6RzVtMUHxe0jUsD1zOZNviiB/S4
  j3JxE190FJCHEb4Ann3Tc1QFCMUMKnmGRfO4WNrgIWrULxFJSsyZVPk+E1g4RDFORwPbIwkEY0jhZaO/s
  W8ptriKNGVZgxU2G7RJOo0s4RTCUEA7wvcxHZFiYAu0jUGWhgN/YEYyztOXiWde/EFxd6o7WbaQ=="
}
```

Rotate the public key

It rotates the key used in the key-based authentication for securely copying the backups. The key rotation breaks the backup transfer operation in the existing backup scheduler, if any. After rotating the key, add the new

CipherTrust Manager public key to `~/.ssh/authorized_keys` file at the destination host machine.

Syntax

```
ksctl scp public-key rotate
```

Example request

```
ksctl scp public-key rotate
```

Example response:

```
{
  "success": true
}
```

To transfer the backup to the destination host machine, run:

Syntax

```
ksctl backup scp --id <backup-id> --host <hostname> port <port> --username <scp-connection-username> --auth-method <key or password> --protocol <scp-or-sftp> --password <scp-connection-password> --public-key <key> --path-to <destination-machine-path>
```

Example request:

```
ksctl backup scp --id fe70d13e-9598-48b8-8b49-d37d2591d7b4 --host 1.2.3.4 port 22 --username ksadmin --auth-method key --protocol sftp --public-key "ssh-rsa AAAAB3NzaCgQD1+kVUr4xNVA9Yhk=" --path-to /home/ksadmin
```

Example response:

```
{
  "id": "5d14bd52-a8fd-4410-bd31-850bfd7a67c0",
  "host": "1.2.3.4",
  "connection_status": "in_progress",
  "createdAt": "2024-07-02T11:19:10.283401126Z",
  "updatedAt": "2024-07-02T11:19:10.283401182Z"
}
```

To get the secure copy backup transfer status

To get the status of the backup transferred through SCP/SFTP, run:

Syntax

```
ksctl backup scp status --id <backup-id> --scpid <scp-id>
```

Example request:

```
ksctl backup scp status --id 7589f772-2e7c-40e3-bc2e-83047de4a93a --scpid 403b4cb1-dbe4-4f17-84e8-4bb81e349707
```

Example response:

```
{
  "id": "403b4cb1-dbe4-4f17-84e8-4bb81e349707",
  "host": "hostname",
  "connection_status": "completed",
  "createdAt": "2021-06-07T06:36:33.139235834Z",
  "updatedAt": "2021-06-07T06:36:43.811369505Z"
}
```

Note

The SCP/SFTP status is maintained for four (4) days, afterward, you can not access it through Status API. The status older than four days can be retrieved from AuditMaps (Server Records).

To get the status of all secure copy backup requests

To get the status of all SCP/SFTP requests for a given backup, run:

Syntax

```
ksctl backup scp status --id <backup-id>
```

Example request:

```
ksctl backup scp status --id 7589f772-2e7c-40e3-bc2e-83047de4a93a
```

Example response:

```
[
  {
    "id": "403b4cb1-dbe4-4f17-84e8-4bb81e349707",
    "host": "hostname",
    "connection_status": "completed",
    "createdAt": "2021-06-07T06:36:33.139235834Z",
    "updatedAt": "2021-06-07T06:36:43.811369505Z"
  },
  {
    "id": "e16d23e8-c34c-4ed3-b704-61227a1c359d",
    "host": "hostname",
    "connection_status": "completed",
    "createdAt": "2021-06-07T06:39:16.983115218Z",
    "updatedAt": "2021-06-07T06:39:16.98311528Z"
  }
]
```

Inspect the backup key file

Before uploading the backup key file, inspect it by reviewing its metadata. To inspect the backup key file, run the command `ksctl backupkeys inspect --file "<backup-key-file>"`.

Example - system backup key file:

```
ksctl backupkeys inspect --file "sysbkkey.txt"
```

Response:

```
{
  "version": 2,
  "keyid": "c05b638d-2eea-4651-9937-7bee49a95f6b",
  "createdAt": "2020-04-07T08:21:59.139786694Z"
}
```

Example - domain scoped backup key file:

```
ksctl backupkeys inspect --file "domainbkkey.txt"
```

Response:

```
{
  "version": 2,
  "keyid": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
  "createdAt": "2020-04-07T08:24:13.648719582Z"
}
```

Upload the backup key

Upload the backup key to the appliance where the backup will be restored.

To upload the system backup key

Run the command `ksctl backupkeys upload --file "<backup-key-file>"`. Here, `<backup-key-file>` represents the name of the system backup key file.

Specify the password of the backup key when prompted.

Example request:

```
ksctl backupkeys upload --file "sysbkkey.txt"
```

Example response:

```
Please provide the backupkey password:
{
  "scope": "system",
```

```
{
  "account": "kylo:kylo:admin:accounts:kylo",
  "version": "2",
  "id": "c05b638d-2eea-4651-9937-7bee49a95f6b",
  "createdAt": "2020-04-07T08:21:59.139786694Z",
  "isDefault": false
}
```

To upload the domain scoped backup key

Run the command `ksctl backupkeys upload --scope domain --file "<backup-key-file>"`.

Here,

- `--scope domain`: Scope of the backup key is domain. The scope flag must be used for uploading domain scoped backup keys.
- `<backup-key-file>`: Name of the domain scoped backup key file.

Specify the password of the domain scoped backup key when prompted.

Example request:

```
ksctl backupkeys upload --scope domain --file "domainbkkey.txt"
```

Example response:

```
Please provide the backupkey password:
{
  "scope": "domain",
  "account": "kylo:kylo:admin:accounts:kylo",
  "domainID": "00000000-0000-0000-0000-000000000000",
  "version": "2",
  "id": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
  "createdAt": "2020-04-07T08:24:13.648719582Z",
  "isDefault": false
}
```

Inspect the backup file

Before uploading the backup file, inspect the file by reviewing its metadata. To inspect the backup file, run the command `ksctl backup inspect --file "<backup-file>"`.

Example - system backup file:

```
ksctl backup inspect --file "backup-file.txt"
```

Response:

```
{
  "version": "v1.1.4",
```



```

    "createdAt": "2020-04-07T08:33:44.687970269Z",
    "scope": "system",
    "domainID": "",
    "backupKeyId": "c05b638d-2eea-4651-9937-7bee49a95f6b",
    "isEncrypted": true,
    "creatorAccount": "kylo:kylo:admin:accounts:kylo",
    "isHSMBacked": false,
    "description": "",
    "checksum": "9f19ab202f860cfa5de7fcbe70872a0aea1ee91af9092c80cb89939c35aea994",
    "mkekChecksum": "dlfe06afe54bfe5f242323b42c640b218a76fae91e7bb0baf10720b47a39be
20",
    "productVersion": "v1.10.0-beta4.4388",
    "id": "245d3336-21fa-4a34-9498-c0b64fa4d322"
  }

```

Example - domain scoped backup file:

```
ksctl backup inspect --file "domain-backup-file.txt"
```

Response:

```

{
  "version": "v1.1.4",
  "createdAt": "2020-04-07T08:36:34.667280981Z",
  "scope": "domain",
  "domainID": "00000000-0000-0000-0000-000000000000",
  "backupKeyId": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
  "isEncrypted": true,
  "creatorAccount": "kylo:kylo:admin:accounts:kylo",
  "isHSMBacked": false,
  "description": "",
  "checksum": "1c1a58b2ead4f09fceb67297603dbf01e40a8bec6d812e78a64287afa9619f8f",
  "mkekChecksum": "12f700ed4f872bd93864cfd7f9d162574025d3ee30ec2c08ccf13362c899a8
37",
  "productVersion": "v1.10.0-beta4.4388",
  "id": "762d15f3-4eba-4cd7-960c-c50a84a52b48"
}

```

Upload the backup

Upload the backup to the appliance where it will be restored.

To upload the system backup

Run the command `ksctl backup upload --file "<backup-file>"`. Here, `<backup-file>` represents the name of the system backup file.

Example request:

```
ksctl backup upload --file "backup-file.txt"
```

Example response:

```
{
  "scope": "system",
  "account": "kylo:kylo:admin:accounts:kylo",
  "version": "v1.1.4",
  "id": "245d3336-21fa-4a34-9498-c0b64fa4d322",
  "createdAt": "2020-04-07T08:33:44.687970269Z",
  "status": "Completed",
  "tiedToHSM": false,
  "backupKey": "c05b638d-2eea-4651-9937-7bee49a95f6b",
  "productVersion": "v1.10.0-beta4.4388"
}
```

To upload the domain scoped backup

Run the command `ksctl backup upload --scope domain --file "<domain-backup-file>"`.

Here,

- `--scope domain`: Scope of the backup is domain. The scope flag must be used for uploading domain scoped backups.
- `<domain-backup-file>`: Name of the domain scoped backup file.

Example request:

```
ksctl backup upload --scope domain --file "domain-backup-file.txt"
```

Example response:

```
{
  "scope": "domain",
  "account": "kylo:kylo:admin:accounts:kylo",
  "domainID": "00000000-0000-0000-0000-000000000000",
  "version": "2",
  "id": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
  "createdAt": "2020-04-07T08:24:13.648719582Z",
  "isDefault": false
}
```

Restore the backup

After the backup and the backup key are uploaded to the appliance, you can restore the backup on the appliance. Check the restore status by running the `ksctl backup status` command. Refer to [check the restore status](#).

Caution

Disable all quorum policies before performing backup restore.

To restore the backup

Run the command `ksctl backup restore --id "<backupID>"`. Here, `<backupID>` specifies the ID of the backup to restore.

System services will restart after the system backup is restored. Restore of the domain scoped backup does not restart the system services.

For domain-scoped backups, login to the desired destination domain where the backup objects will be restored.

Caution

Domain scoped backup fails to restore on another domain when a key with the same name and version, but a different ID, already exists. The error "Error copying data from backup database" is displayed.

To handle this issue, try either of the following:

- Retain both keys.
 - a. Take the backup without the conflicting key with filters.
 - b. Export/import the key material and import it separately.
- Retain only the backup key.
 - a. Delete the key with duplicate name on the restore system.
 - b. Restore the domain scoped backup.

Example request:

```
ksctl backup restore --id "245d3336-21fa-4a34-9498-c0b64fa4d322"
```

Example response:

```
{
  "version": "v1.1.4",
  "id": "245d3336-21fa-4a34-9498-c0b64fa4d322",
  "createdAt": "2020-04-07T08:33:44.687970269Z",
  "status": "Completed",
  "tiedToHSM": false
}
```

Note

Restoring a backup can often change the Local CA associated with an interface, which generates the

server certificate. If that occurs, you must [manually re-issue the server certificate](#) so that clients can reconnect to the interfaces. We recommend running `ksctl interfaces list` after the restore to view if the `auto_gen_ca_id` value has changed.

Strategies for restoring domain backup

If there are conflicting resources in the backup and restore domains, the following strategies are used by the CipherTrust Manager during backup restore:

Strategy	Description	Resource Type
Error	An error is thrown on restore.	• Keys
Skip	Skips the conflicting resource of the backup domain and keeps the restore domain one.	• Users and Groups • CTE Policies • CCKM Resources (Projects, Cryptospaces, EKM Endpoint) Refer to the Backup and Restore section in CCKM documentation for behavior of CCKM resources.

Check the restore status

Warning

When restoring the backup, if the WEB port is changed in the backup, the GUI becomes unresponsive for an indefinite time and does not show the restore status. To know the exact restore status, call the `/v1/backupStatus` API and when it starts showing the status as "Completed", reload the GUI with the updated port.

After performing the backup restore, you can check the status of the operation. To view the status of the restore, run the command `ksctl backup status`. The status of the restore is given in the response.

Example request:

```
ksctl backup status
```

Example response:

```
{
  "account": "kylo:kylo:admin:accounts:kylo",
  "id": "f6b55640-ddec-49d0-893b-4f33f0809bf0",
  "createdAt": "0001-01-01T00:00:00Z",
  "status": "In Progress",
  "tiedToHSM": false
}
```

In the output above, the status of the restore is `In Progress`.

Note

It is normal to get an error like the following for status when the system services are restarting after the system backup is restored:

Error: Server error. HTTP status code: 500

Filter domain-scoped backup to restore selective keys

This feature adds filtering capabilities when restoring keys from a domain-scoped CipherTrust Manager backup, allowing you to specify which keys to restore. The below tabs explain how to do this through the CLI and UI.

Note

The Browse preparation may take several minutes to open the backup for further restoration process.

CLI

To do so, use the following commands:

1. `browse-prepare`
2. `browse`
3. `restore`
4. `browse-cleanup`

Browse-prepare domain-scoped backup

The `browse-prepare` command creates a temporary copy of domain-scoped backup based on the ID of the domain backup. This backup can later be used by the `browse` command to inspect its content.

Syntax

```
ksctl backup browse-prepare --id <backup-id-of-domain-scoped-backup>
```

Request

```
ksctl backup browse-prepare --id ed1c3b32-7074-4d94-b2f6-ebbd1c5df0ed
```

Response

You can check the `browse-prepare` status using the command below:

Syntax

```
ksctl backup status
```

Response

```
{
  "scope": "domain",
  "account": "kylo:kylo:admin:accounts:kylo",
  "operation": "Browse prepare",
  "id": "f6ae27d7-9809-4eb6-9055-ecce85665398",
  "started": "2025-05-13T10:56:13.459513102Z",
  "finished": "2025-05-13T10:56:20.684139223Z",
  "status": "Completed"
}
```

Browse domain-scoped backup

The `browse` command examines the backup generated using `browse-prepare` before restoring it. You can use this command to filter the resources based on the filter parameters.

Note

To use `browse`, ensure that the `browse-prepare` operation is successful.

Syntax

```
ksctl backup browse --id <backup-id> --filter <keys-filter>
```

Request

```
ksctl backup browse --id ed1c3b32-7074-4d94-b2f6-ebbd1c5df0ed --filter '{"resourceType": "Keys","resourceQuery": {"algorithm": "RSA"}}'
```

Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resourceType": "keys",
  "resources": [
    {
      "account": "",
      "activationDate": "2025-04-25T06:53:17.478634Z",
      "algorithm": "AES",
      "application": "ncryptify:gemalto:admin:apps:kylo",
      "createdAt": "2025-04-25T06:53:17.480667Z",

```

```

    "defaultIV": "acb653e45bacde0a2e8eebd5853eec04",
    "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
    "emptyMaterial": false,
    "format": "raw",
    "id": "2b1a70d4807a4241b8c51cbbb2a963c2a66bb8dca7db41d89f5134b1bbfc2eb9",
    "keyCheckValue": "0b7b6c",
    "meta": {
      "ownerId": "local|2732f65d-153a-426e-8f68-f9701eb47299"
    },
    "muid": "63418a6d-0a24-4dc8-bf75-d88936c53db9fbec4fbd-93ab-44b9-ac5c-6f9b2e5da70d",
    "name": "key500",
    "neverExportable": false,
    "neverExported": true,
    "objectType": "Symmetric Key",
    "sha1Fingerprint": "4bc68f48420339cf",
    "sha256Fingerprint": "0baa8e473c38edc6bd78bf188426693a34ac73869b160dea0271e6390252cabb",
    "size": 256,
    "state": "Active",
    "undeletable": false,
    "unexportable": false,
    "updatedAt": "2025-04-25T06:53:17.480667Z",
    "uri": "kylo:kylo:vault:keys:key500-v0",
    "usage": "blob",
    "usageMask": 12,
    "uuid": "63418a6d-0a24-4dc8-bf75-d88936c53db9",
    "version": 0
  }
]
}

```

Selective Restore for domain-scoped backup

After the backup is ready and examined using `browse-prepare` and `browse` commands. You can restore the selective keys using filters.

Note

If a single key from a key pair is restored, the other part will not be automatically restored. You need to explicitly select and restore the other part if needed.

Syntax

```
ksctl backup restore --id <id> --filters <keys-filters>
```

OR

```
ksctl backup restore --id <id> --filters-jsonfile <filename-containing-filters-json>
```

Request

```
ksctl backup restore --id ed1c3b32-7074-4d94-b2f6-ebbd1c5df0ed --filters-jsonfile restore.json
```

OR

```
ksctl backup restore --id ed1c3b32-7074-4d94-b2f6-ebbd1c5df0ed --filters '[{"resourceType": "Keys","resourceQuery": {"version": 0}}]'
```

Response

```
{
  "scope": "domain",
  "account": "kylo:kylo:admin:accounts:kylo",
  "id": "a1839ce6-3993-4d27-ad33-793d1334287e",
  "createdAt": "0001-01-01T00:00:00Z",
  "status": "In Progress",
  "tiedToHSM": false
}
```

Browse-cleanup domain scoped backup

The `browse-cleanup` command removes the temporary backup created by `browse-prepare`.

Syntax

```
ksctl backup browse-cleanup --id <backup-id-of-domain-scopep-backup>
```

Request

```
ksctl backup browse-cleanup --id ed1c3b32-7074-4d94-b2f6-ebbd1c5df0ed
```

Response

You can check the browse-cleanup status using the command below:

Syntax

```
ksctl backup status
```

Response

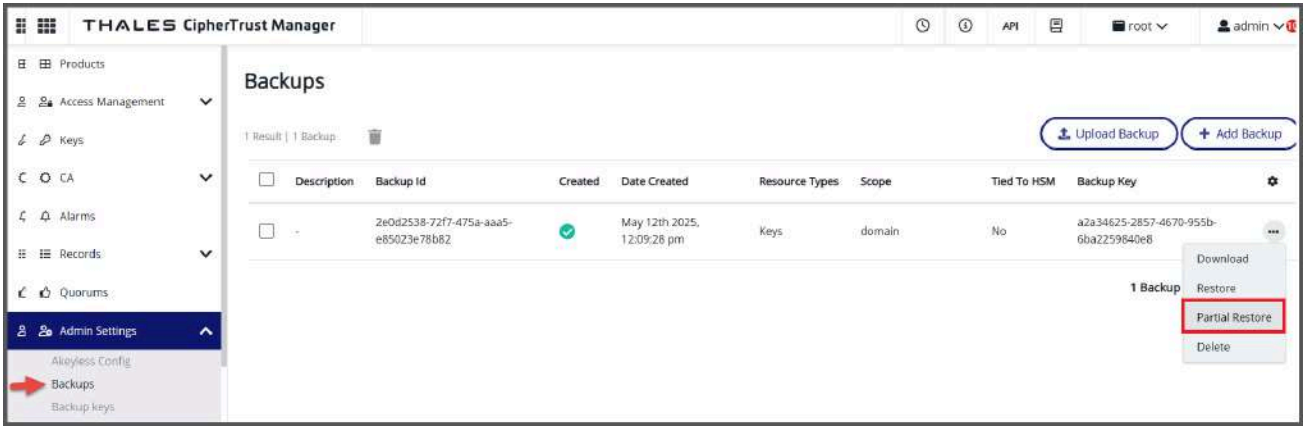
```
{
  "scope": "domain",
  "account": "kylo:kylo-576b32eb-47b3-4452-94e6-fdc7be55d1cb:admin:accounts:kyl",
  "operation": "Browse cleanup",
  "id": "45250993-01b7-44cd-bb9f-c62bbad95ddb",
  "started": "2025-05-14T09:09:33.493845988Z",
  "finished": "2025-05-14T09:09:33.667646238Z",
  "status": "Completed"
}
```


Note

The scheduled clean up occurs every 24 hours to remove domain-scoped backups created using browse-prepare.

UI

- 1. Log on to the CipherTrust Manager GUI as administrator.
- 2. Navigate to **Admin Settings > Backups**.
- 3. Navigate to the desired **Domain Backup**. Click the ellipsis (...) icon on the right and select **Partial Restore**.



The Prepare for partial restore screen appears.

- 4. Select the **Confirm open backup** checkbox and click **Open Backup**. It prepares and open the backup file for browsing.

The Partial Restore Flow screen appears.

- 5. You can select keys either by name or using Advanced filtering options.

By name

The "By name" tab allows to select individual keys.

- a. Navigate to **Find keys**. This tab appears by default.

The table below lists the keys available for restoration. You can also use the search bar to locate keys within this backup file.

- b. Select the desired keys using the plus (+) button in the right column. All versions of the selected keys will be restored.

Partial Restore Flow

1 Select Keys

2 Restore Keys

☒ By name
Select individual keys

☐ Advanced filtering
Define criteria for keys to restore using advanced filtering controls

The table below shows keys that are available to choose for restoration. You can use the search bar to find keys within this backup file, and then select them by using the plus button in the right side column. All versions of selected keys will be restored.

Find keys

Selected key list

Key Name

Search by Key Name

2 Results | 2 Keys

Name	Owner Name	Type	Alg	Size	
anuj-test	local admin	Symmetric Key	AES	256	+
citrus-cd309595-f61...	--	Symmetric Key	AES	256	+

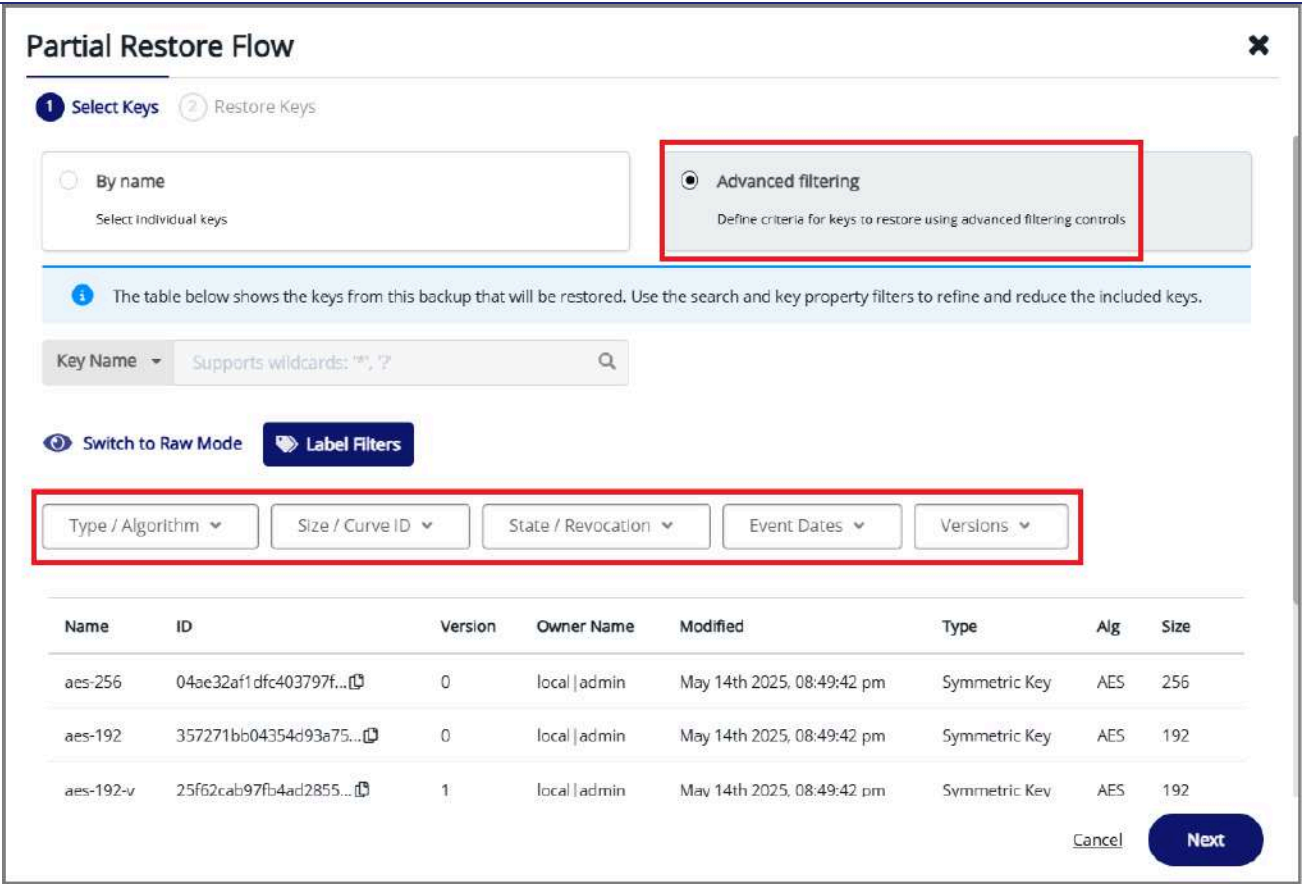
Cancel

Next

You can see the selected keys in the **Selected key list** tab.

Advanced filtering

The table shown in the screenshot below lists all the keys from this backup that will be restored. You can refine and narrow down the keys by using search and key property filters such as Type/Algorithm, Size/Curve ID, State/Revocation, Event Dates, or Versions.



The "Advanced Filtering" tab provides multiple options to filter the keys:

- Choose a single key property filter or combine multiple filters based on your requirements.
- Use **Switch to Raw Mode** to write custom queries for filtering the keys that need to be restored.



- Use the Search bar as shown in the screenshot below to filter the keys.

The screenshot shows the 'Key Name' search bar with a dropdown menu open. The dropdown menu lists the following options: ID, Alias, Owner ID, Owner Name, CTE Key Hash, and Key Check Value. Below the dropdown, there is a table with columns: ID, Version, and Owner Name. The table contains one row with the following data: ID: f8ae661305f24952ae6..., Version: 0, Owner Name: --.

6. Click **Next**.
7. Select **Confirm Restore** checkbox and click **Restore**.

The 'Partial Restore Flow' dialog box is shown. It has a progress bar with two steps: '1 Select Keys' and '2 Restore Keys'. A message box says: 'You have selected 1 keys for restoration from this backup file. To start restoration, click "Restore" below. Restoration may take some time depending on how many keys have been selected.' Below the message, the 'Confirm Restore' checkbox is checked. At the bottom right, there are 'Back' and 'Restore' buttons.

8. Check the restoration status on the next screen and click **Done**.

The 'Partial Restore Flow' dialog box is shown again, but now the restoration is complete. The progress bar shows '1 Select Keys' and '2 Restore Keys'. A message box says: 'Restore keys completed.' At the bottom right, there are 'Back' and 'Done' buttons.

If an API call fails during the restore attempt, the below screen will be displayed.



Backup/restore of customer fragments

Prerequisites

- CipherTrust Manager is up and running. Refer to [CipherTrust Manager Deployment](#) for details.
- Make sure to enable the `secrets-manager` service (**Admin Settings > Services**) before backup and restore operations. Otherwise, domain backup/partial domain backup will not include customer fragments into it.

Important notes

For the root-domain users

- Backup and restore of customer fragments can only be performed in the `root` domain. Let's understand different scenarios when the user is in the root domain and the `secrets-manager` service is enabled.

Backup Type	CSM Service State While Backup	Backup Result	CSM Service State While Restore	Restore Result
Full Domain Backup	Enabled	Backup will include customer fragments and all other resource types.	Enabled	Customer fragments and all other resources will be successfully restored along with the keys associated with the customer fragments.
Full Domain Backup	Enabled	Backup will include customer fragments and all other resource types.	Disabled	All other resources except customer fragments will be restored.
Partial Domain Backup	Enabled	Backup will include customer fragments and all other resource types if selected.	Enabled	All selected resources will be restored including the customer fragments with the

Backup Type	CSM Service State While Backup	Backup Result	CSM Service State While Restore	Restore Result
				corresponding keys.
Partial Domain Backup	Enabled	Backup will include customer fragments and all other resource types if selected.	Disabled	All selected resources will be restored, excluding the customer fragments.

For the non-root domain users

- CipherTrust Manager doesn't support customer fragments backup and restore in the non-root domain. Let's understand different scenarios if backup or restore occurs in non-root domain.

Backup or Restore	Backup Type	Outcome
Create Backup	Full Domain Backup	Backup will include all other resource types excluding customer fragments.
Create Backup	Partial Domain Backup	If customer fragments are included explicitly during backup. The following error will be thrown "Resource type ("customer_fragments") is not supported for the current domain".
Restore Backup	Full Domain Backup or Partial Domain Backup with customer fragments into it	The restore process will complete successfully, although customer fragments will be ignored and logged in the Audit records with the message "Ignoring customer_fragments restoration as it is only supported on the root domain.". All other resources will be restored as expected.
Restore Backup	Full Domain Backup or Partial Domain Backup without customer fragments into it	Backup will get restored successfully in the non-root domain as customer fragments are not there in the backup.

Note

If users of CSM Admins group want to backup or restore a customer fragment, they must also be part of the Domain Backup Admins and Domain Restore Admins groups.

Creating backup

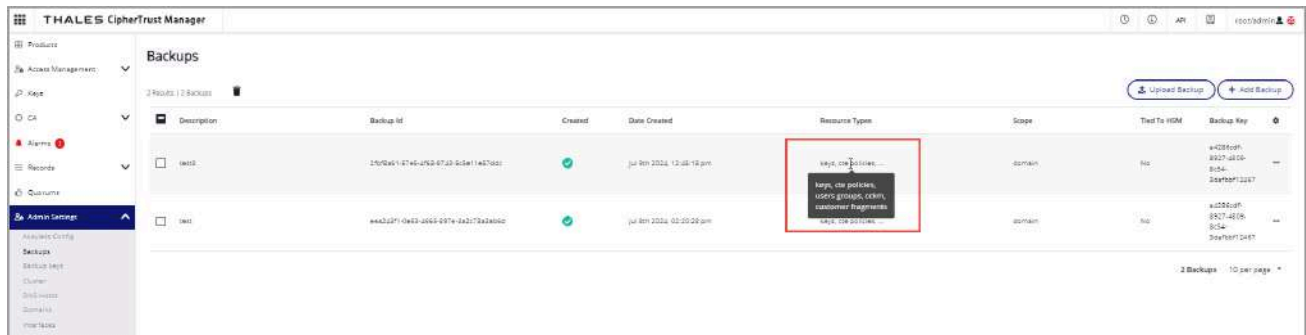
GUI

This section describes the GUI steps for creating a domain backup.

Domain Backup

1. Log on to the CipherTrust Manager GUI as administrator.
2. Navigate to **Admin Settings > Backups**.
3. Click **Add Backup**. The Add Backup wizard is displayed.
4. Select **Domain Backup** radio button.
5. Click **Next**.
6. Provide description and select domain backup key from the list.
7. Click **Add Backup**. The backup will include all resources types including customer fragments.

You can see the list of created backups on the dashboard.



Partial Domain Backup

1. Log on to the CipherTrust Manager GUI as administrator.
2. Navigate to **Admin Settings > Backups**.
3. Click **Add Backup**. The Add Backup wizard is displayed.
4. Select **Backup Customer Fragments** check box. Select other resource types if needed.
5. Click **Next**.
6. In the **Select Customer Fragments** section, select any one of the three (3) options:
 - **Select All** - Selects all customer fragments at once.
 - **Manual Select** - Click the plus (+) button next to the fragments you want to select. *Select at least one customer fragment.*

- **None** - No customer fragment will be selected.

You can also use the search bar to find the customer fragments.

Add Backup

1 Select Backup Scope
2 Select Keys, CTE Policies, Users/Groups, or Customer Fragments
3 **Select Customer Fragments**
4 Backup Properties

☒ **Select All**
Selects all Customer fragments.

☐ **Manual Select**
Select at least one Customer fragment.

☐ **None**
No Customer fragment will be selected.

i The table below shows Customer fragments that are available to choose for backup. You can use the search bar to find Customer fragments

Selected Customer fragment list

Name

5 Results | 11 Customer fragments

Name	Id	Description	Date Added
CipherTrust Manager's default customer ...	6222f9aa-60ae-44fc-...	CipherTrust Manager generated fragment.	25 Jul 2024, 10:50
cf1	0f2b33c9-4338-41c8-...	-	25 Jul 2024, 09:46
cf2	6770c1d3-e47c-4b10-...	-	25 Jul 2024, 09:46
cf3	959a4e61-b324-421a-...	-	25 Jul 2024, 09:46
cf4	a817983b-c478-49ad-...	-	25 Jul 2024, 09:46

[Back](#)
[Next](#)

- Click **Next**.
- Provide description and select domain backup key from the list.
- Click **Add Backup**. The backup will include the customer fragments and the associated keys.

You can see the list of created backups on the dashboard.

Name	ID	Status	Date	Scope	ID
test-backup	04056621-9365-4816-8423-07ae13fed3c3	✓	Jul 26th 2024, 01:09:09 pm	customer fragments ...	7ae471db2-037e-4391-ae45-1824ca252003
-	05ce990d-48a3-44c3-9aac-0710792d9f90	✓	Jul 26th 2024, 01:01:57 pm	keys, cte policies ...	7ae471db2-037e-4391-ae45-1824ca252003
-	6ae9040d-c306-44a6-af64-10f1fa28d777	✓	Jul 26th 2024, 10:11:31 am	keys, cte policies ...	7ae471db2-037e-4391-ae45-1824ca252003
-	00a89017-caea-48ce-ec0d-cfcb9ae67126	✓	Jul 26th 2024, 10:10:47 am	customer fragments ...	7ae471db2-037e-4391-ae45-1824ca252003
-	01424190-c078-4073-b9c7-e15ae42e3d0c	✓	Jul 26th 2024, 10:09:05 am	customer fragments ...	7ae471db2-037e-4391-ae45-1824ca252003
-	588bdc8e-ec0f-4786-eb06-3c33a46b7498	✓	Jul 26th 2024, 05:22:14 am	keys, cte policies ...	7ae471db2-037e-4391-ae45-1824ca252003
-	8a70f74d-a716-4b7a-a788-c4d01a0efb01	✓	Jul 26th 2024, 12:45:09 am	Keys	7ae471db2-037e-4391-ae45-1824ca252003

CLI

This section describes the ksctl commands for creating a domain backup.

Domain Backup

For domain backup

```
ksctl backup create --scope domain
```

The above command will create a full domain backup.

Partial Domain Backup

For partial-domain backup

```
ksctl backup create --scope domain --filters <filters-json>

or

ksctl backup create --scope domain --filters-jsonfile <file-name-containing-filters-json>
```

Here,

Filters are a set of selection criteria to specify what resources to include in the backup. Only applicable to domain-scoped backups. For example, to backup CSM customer fragments, set the filters to `[{"resourceType": "customer_fragments", "resourceQuery":{"names":["CipherTrust Manager's default customer fragment"]}]`.

Example 1

```
ksctl backup create --scope domain --filters "[{"resourceType": "customer_fragments", "resourceQuery":{"ids":["47f03ddd-c92b-4834-a4d3-dfe60b5fcf6a","fd9cf9b5-179a-4eba-a58b-dd321484e474"]}]"]
```

In `resourceQuery` you pass send either ids or names as parameters. In the above example, we are passing ids of customer fragments.

Example 2

```
]]]]"
```

In the above example, we are passing names of the customer fragments.

Example 3

```
ksctl backup create --scope domain --filters-jsonfile test.json
```

test.json

```
[{"resourceType": "customer_fragments", "resourceQuery":{"names":["CipherTrust Manager's default customer fragment","CF 1"]}}]
```

In the above example, we are using JSON file to pass the parameters.

Restoring backup

Caution

- Domain scoped backup fails to restore when a customer fragment with the same name but a different ID, already exists. The error "Error copying data from backup database" is displayed.

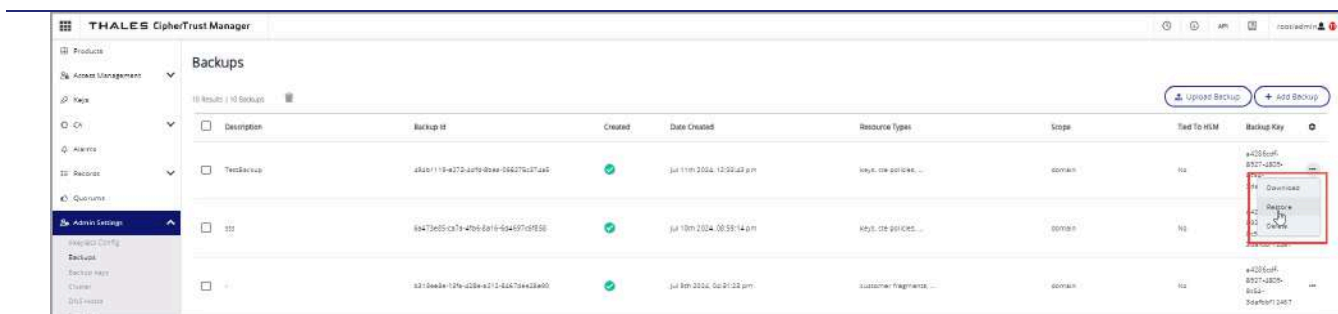
To handle this issue, try either of the following:

- Take backup without the conflicting customer fragments.
- Resolve the conflict before restore.
- After successful restore of customer fragments, the Akeyless gateway restarts across all the nodes of a cluster.

GUI

This section describes the GUI steps for creating a domain backup.

1. Log on to the CipherTrust Manager GUI as administrator.
2. Navigate to **Admin Settings > Backups**.
3. Go to the desired backup. Click the ellipsis on the right side of a backup and click **Restore** as shown below.



CLI

To restore the backup:

```
ksctl backup restore --id <backupID>
```

For details, refer to [Restore the backup](#).

Scheduling Backups

Note

This feature is not included in Community Edition and requires a valid Virtual CipherTrust Manager license. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

The CipherTrust Manager can periodically run database backup jobs in the background. This can be achieved by creating [scheduler](#) job configurations for database backups. Scheduler configurations are supported for both system and domain scoped backups. Refer to [Backups](#) for details on backups and their types.

You must be in the Backup Admins or Domain Backup admins [group](#), and logged into the root [domain](#) to manage backup job configurations.

In a clustered environment, by default the scheduled backup runs on the same cluster node (this node) where it was created. There is an option in the API and CLI to restrict which nodes can run the scheduled backup.

Caution

By default, no limit is set on the number of backup files retained on CipherTrust Manager. To avoid disk space issues, we strongly recommend using the `--retention-count` flag to set a limit on the number of backup files to retain.

As well, disk space issues can also occur if you do not periodically transfer or manually delete backup files. We recommend using the `--do-scp true` and `--conn "<scp_connection_name>"` flags to send backup files to an external server [registered as an external connection](#).

This section provides examples of backup scheduler configurations using the `ksctl` utility. Refer to the `ksctl` or API

documentation for details.

Note

Scheduler itself is a generic service, so it provides granularity of a minute. A job configuration should take the nature of operation information into consideration. For example, backing up a database every minute is a highly unlikely use case.

Scheduling system scoped backup jobs

To schedule a system scoped backup job, run the command:

```
ksctl scheduler configs create backup --name <config_name> --run-at "<cron_expression_format>" --description "<config-description>" --backup-key-id "<backup_key_ID>" --retention-count <number_of_backups_to_retain> --do-scp --conn <scp-connection>
```

Here,

- `<config_name>`: Name for the job configuration.
- `<cron_expression_format>`: Time when the job runs. The format must have *five* fields. If the number of fields is not equal to five, the format becomes invalid. Refer to [Time Specification](#) for details.
- `<config_description>`: Description of the job configuration.
- `<backup_key_ID>`: ID of the backup key. If a key ID is not specified, the default backup key will be used.

Note

If you are operating a cluster and specify a backup key, be sure to [upload that backup key to every node](#). Otherwise, the scheduled backup will fail on nodes that do not have the backup key.

- `<number_of_backups_to_retain>`: An integer value indicating how many backup files to store on the CipherTrust Manager.
- `<scp-connection>`: Name or ID of the SCP connection to external SCP server.

Example

The following command schedules a system scoped database backup job using the specified backup key everyday at 9 AM.

```
ksctl scheduler configs create backup --name daily\_backup --run-at "0 9 * * *"
```

```
--description "Daily system backup" --backup-key-id "e817b3ba-1787-496a-b5ac-27756e9efb1b" --retention-count 5
```

Scheduling domain scoped backup jobs

To schedule a domain scoped backup job, run the command:

```
ksctl scheduler configs create backup --scope "domain" --name <config_name> --run-at "<cron_expression_format>" --description "<config_description>" --retention-count <number_of_backups_to_retain>
```

Here,

- `scope`: Scope of the backup is domain. The scope flag must be used for scheduling domain scoped backup jobs.
- `<config_name>`: Name for the job configuration.
- `<cron_expression_format>`: Time when the job runs. The format must have *five* fields. If the number of fields is not equal to five, the format becomes invalid. Refer to [Time Specification](#) for details.
- `<config_description>`: Description of the job configuration.
- `<number_of_backups_to_retain>`: An integer value indicating how many backup files to store on the CipherTrust Manager.

Example:

The following command schedules a domain scoped database backup job every Monday at 9 AM.

```
ksctl scheduler configs create backup --scope "domain" --name daily_backup --run-at "0 9 * * 1" --description "Weekly domain backup" --retention-count 5
```

Scheduling backup for customer fragments

This section describes how to schedule backup for customer fragments.

Prerequisites

- CipherTrust Manager is up and running. Refer to [CipherTrust Manager Deployment](#) for details.
- Make sure that `secrets-manager` service (**Admin Settings > Services**) is **enabled**:
 - Before creating a scheduler for backup.
 - While running the scheduler to create backup.

Otherwise, domain backup/partial domain backup will not include customer fragments into it.

Important notes

- You can schedule backup of customer fragments in the `root` domain **only**.
- For the non-root domain users, backup will include all other resource types excluding customer fragments.
- If users of CSM Admins group want to schedule backup for customer fragments, they must also be part of the Domain Backup Admins and Domain Restore Admins groups.

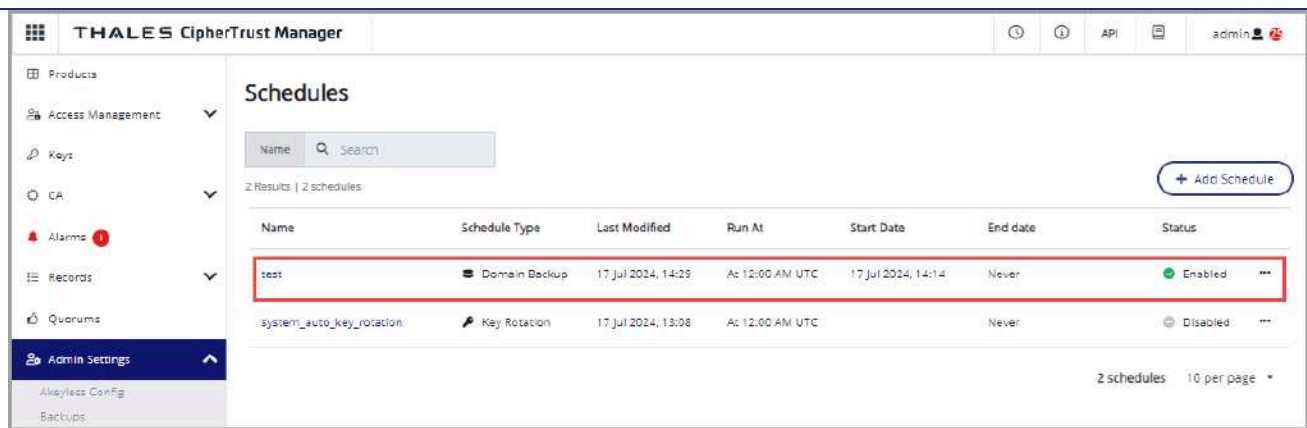
GUI

This section describes the GUI steps for scheduling a domain backup.

Domain Backup

1. Log on to the CipherTrust Manager GUI as administrator.
2. Navigate to **Admin Settings > Schedules**.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. Select **Domain Backup** radio button. Click **Next**.
5. Provide Schedule Name and Description.
6. Select **Enable Schedule** check box so that backup can be scheduled. Click **Next**.
7. Add Schedule Config. Refer to [Time Specification](#) for details.
8. Select **Number of backups to retain (leave blank for unlimited)** from spinbox. This value indicates how many backup files to store on the CipherTrust Manager. Provide description and select domain backup key from the list.
9. Click **Add Schedule**. The backup will include all resources types including customer fragments.

You can see the list of created backup schedules on the dashboard.



Partial Domain Backup

1. Log on to the CipherTrust Manager GUI as administrator.
2. Navigate to **Admin Settings > Schedules**.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. Select **Partial Domain Backup** radio button. Click **Next**.
5. Provide Schedule Name and Description.
6. Select **Enable Schedule** check box so that backup can be scheduled. Click **Next**.
7. Add Schedule Config. Refer to [Time Specification](#) for details.
8. Choose Keys, if needed. Click **Next**.
9. Choose CTE policies, if needed. Click **Next**.
10. Choose user/email/group, if needed. Click **Next**.
11. In the **Choose Customer Fragments** section, select any one of the three (3) options:
 - **Select All** - Selects all customer fragments at once. **If new customer fragments are added to the system after the backup is scheduled, the new customer fragments will also be added to the backup.**
 - **Manual Select** - Click the plus (+) button next to the fragments you want to select. *Select at least one customer fragment.*
 - **None** - No customer fragment will be selected.

You can also use the search bar to find the customer fragments.

Add Partial Domain Backup Schedule

1 Select Schedule Type

2 General Info

3 Schedule Config

4 Choose Keys

5 CTE Policies

6 Users/Groups

7 Choose Customer Frag

☒ Select All
Selects all Customer fragments.

☐ Manual Select
Select at least one Customer fragment.

☐ None
No Customer fragment will be selected.

The table below shows Customer fragments that are available to choose for backup. You can use the search bar to find Customer fragment. If new Customer fragments are added to the system after the backup is scheduled, the new Customer fragments will also be added to the backup.

Selected Customer fragment list

Name

Supports wildcards: *?*

5 Results | 11 Customer fragments

Name	Id	Description	Date Added
CipherTrust Manager's default customer ...	6222f8aa-60ae-44fc-...	CipherTrust Manager generated fragment.	25 Jul 2024, 10:50
cf1	0f2b33c9-4338-41c8-...		26 Jul 2024, 09:46

Back

Next

12. Select **Number of backups to retain (leave blank for unlimited)** from spinbox. This value indicates how many backup files to store on the CipherTrust Manager. Provide description and select domain backup key from the list.

13. Click **Add Schedule**. The backup will include the customer fragments and the associated keys.

You can see the list of created backup schedules on the dashboard.

Schedules

Name

Search

8 Results | 8 schedules

+ Add Schedule

Name	Schedule Type	Last Modified	Run At	Start Date	End date	Status
test-schedule	Partial Domain Backup	26 Jul 2024, 12:43	At 12:00 AM UTC	26 Jul 2024, 12:19	Never	Enabled
test2-manual	Partial Domain Backup	26 Jul 2024, 10:21	At 12:00 AM UTC	26 Jul 2024, 10:08	Never	Enabled
test-none	Partial Domain Backup	26 Jul 2024, 10:20	At 12:00 AM UTC	26 Jul 2024, 10:08	Never	Enabled
test1	Partial Domain Backup	26 Jul 2024, 10:15	At 12:00 AM UTC	26 Jul 2024, 10:08	Never	Enabled
sb	System Backup	26 Jul 2024, 00:23	At 12:00 AM UTC	25 Jul 2024, 23:44	Never	Enabled
db	Domain Backup	26 Jul 2024, 00:23	At 12:00 AM UTC	25 Jul 2024, 23:44	Never	Enabled
kr	Key Rotation	26 Jul 2024, 00:13	At 12:00 AM UTC	25 Jul 2024, 23:44	Never	Enabled

Thales Cybersecurity Docs Hub © 2019–2026, Thales Group. All rights reserved.
CipherTrust Manager | Last Updated: January 29, 2026

471

To view/edit the partial domain backup schedule:

1. Go to the desired backup schedule.
2. Click the ellipsis on the right side of a schedule and click **View/Edit** as shown below.

Schedules

Name

Search

8 Results | 8 schedules

+ Add Schedule

Name	Schedule Type	Last Modified	Run At	Start Date	End date	Status
test-schedule	Partial Domain Backup	26 Jul 2024, 12:43	At 12:00 AM UTC	26 Jul 2024, 12:19	Never	Enabled View/Edit
test2-manual	Partial Domain Backup	26 Jul 2024, 10:21	At 12:00 AM UTC	26 Jul 2024, 10:08	Never	Disable
test-none	Partial Domain Backup	26 Jul 2024, 10:20	At 12:00 AM UTC	26 Jul 2024, 10:08	Never	Run Now
test1	Partial Domain Backup	26 Jul 2024, 10:15	At 12:00 AM UTC	26 Jul 2024, 10:08	Never	Delete
sb	System Backup	26 Jul 2024, 00:23	At 12:00 AM UTC	25 Jul 2024, 23:44	Never	Enabled
db	Domain Backup	26 Jul 2024, 00:23	At 12:00 AM UTC	25 Jul 2024, 23:44	Never	Enabled
kr	Key Rotation	26 Jul 2024, 00:13	At 12:00 AM UTC	25 Jul 2024, 23:44	Never	Enabled

3. Go to CUSTOMER FRAGMENTS section to view/edit the details.

Trust Manager

USERS / GROUPS

CUSTOMER FRAGMENTS

Select All

Manual Select

None

The table below shows Customer fragments that are available to choose for backup. You can use the search bar to find Customer fragment. If new Customer fragments are added to the system after the backup is scheduled, the new Customer fragments will also be added to the backup.

Selected Customer fragment list

Name

Search

5 Results | 11 Customer fragments

Name	id	Description	Date Added
CipherTrust Manager's default customer ...	6222f8aa-60ae-44fc-...	CipherTrust Manager generated fragment.	25 Jul 2024, 10:50
cf1	0f2b53c9-4358-41c8-...	-	26 Jul 2024, 09:46
cf2	6770c1d3-e47c-4010-...	-	26 Jul 2024, 09:46
cf3	959e4e61-b324-421a-...	-	26 Jul 2024, 09:46
cf4	a817983b-c478-49ad-...	-	26 Jul 2024, 09:46

1 to 5 of 11 Customer fragments

CLI

Domain Backup

For full domain backup:

Thales Cybersecurity Docs Hub © 2019–2026, Thales Group. All rights reserved.
CipherTrust Manager | Last Updated: January 29, 2026

472

```
ksctl scheduler configs create backup --scope "domain" --name <config_name> --run-at "<cron_expression_format>" --description "<config_description>" --retention-count <number_of_backups_to_retain>
```

For details, refer to [Scheduling domain scoped backup jobs](#)

Partial Domain Backup

For partial domain backup:

Note

The examples are provided for Windows platform. To run these examples on Linux, see [CLI usage with JSON parameters](#).

Example 1

```
ksctl scheduler configs create backup --name "cf-backup-scd" --run-at "*/5 * * * *" --scope "domain" -description "test cf backup scheduler" --filters "[{"resourceType": "\"customer_fragments\"", \"resourceQuery\": {\"names\": [\"CipherTrust Manager's default customer fragment\"]}}]"
```

Here,

Filters are a set of selection criteria to specify what resources to include in the backup. Only applicable to domain-scoped backups. For example, to backup CSM customer fragments, set the filter to [{"resourceType": "customer_fragments", "resourceQuery": {"names": ["CipherTrust Manager's default customer fragment"]}}].

In resourceQuery you can pass either "ids" or "names" as parameters. In the above example, we are passing "names" of the customer fragments.

Example 2

```
ksctl scheduler configs create backup --name "cf-backup-scd" --run-at "*/5 * * * *" --scope domain -description "test cf backup scheduler" --filters "[{"resourceType": "\"customer_fragments\"", \"resourceQuery\": {\"ids\": [\"47f03ddd-c92b-4834-a4d3-dfe60b5fcf6a\", \"fd9cf9b5-179a-4eba-a58b-dd321484e474\"]}}]"
```

In the above example, we are passing "ids" of the customer fragments.

Example 3

```
ksctl scheduler configs create backup --name "cf-backup-scd-1" --run-at "*/5 * * * *" --scope domain -description "test cf backup scheduler" --filters-jsonfile filter.json
```

filter.json

```
[{"resourceType": "customer_fragments", "resourceQuery":{"names":["CipherTrust Manager's default customer fragment","CF 1"]}}]
```

In the above example, we are using JSON file to pass the parameters.

Disk Encryption After Initial Deployment

For added security, the disk of CipherTrust Manager can be fully encrypted with the public SSH key. For public cloud deployments this SSH key was provided during first launch. For physical appliance and private cloud deployments, the [SSH key is provided](#) after initial deployment.

Encryption can be initiated when an instance is first launched for Virtual CipherTrust Manager, or after deployment for both physical appliance and virtual instances. [Cloud-init configuration](#) with a **user-data** file is used for encryption on first launch.

Because installation specific secrets are generated the first time a Virtual CipherTrust Manager instance is launched, it is recommended that the instance be encrypted at launch time when possible to ensure these secrets are never exposed.

After encrypting the disk, you will need to unlock the encrypted instance on every boot using the 'ksctl diskenc secureboot' command and the private SSH key associated with the instance. See [to unlock an encrypted instance](#). Disk encryption is always applied, and this behavior cannot be disabled. If you wish to store your keys on an unencrypted instance, you can launch a new Virtual CipherTrust Manager and then use [backup and restore](#) to transfer keys and other data.

Security Best Practices

- We always recommend enabling disk encryption.
- Because installation specific secrets are generated the first time a Virtual CipherTrust Manager instance is launched, it is recommended that the instance be encrypted at launch time when possible to ensure these secrets are never exposed.
- Provide an externally signed certificate to the preboot interface if possible. As you cannot provide this certificate with cloud-init configuration, this is the preferred deployment sequence for Virtual CipherTrust Manager deployments:
 - a. [Enable disk encryption with cloud-init](#).
 - b. Launch the Virtual CipherTrust Manager instance.
 - c. [Upload an externally signed certificate](#).
 - d. Reboot the Virtual instance.
 - e. Verify the preboot certificate when unlocking the disk.

Encrypting an already launched instance

The following are examples for encrypting an already launched instance and for checking on its encryption status.

Also provided is a CLI example for unlocking the instance at boot time. These operations are only available through CLI or REST API.

To encrypt the instance

1. Run the following command:

```
ksctl diskenc cryptsetup
```

CipherTrust Manager creates the preboot interface and dynamically generates a self-signed certificate with the default server name `preboot.ciphertrustmanager.local`.

2. (Optional, but recommended) Upload an externally signed certificate file to the preboot interface. This certificate is used to protect ksctl and REST TLS communications with CipherTrust Manager while the disk is locked, in the preboot stage.

```
ksctl interfaces certificate modify --name preboot --file <certificate_file>
--format <pem_or_pkcs12>
```

CipherTrust Manager supports PEM or base64-encoded PKCS#12 format certificates. The private key algorithm must be RSA-2048 or RSA-4096. The certificate file must include the private key. You can include a chain of certificates, in ascending order from server certificate to root certificate authority (CA).

3. Save the preboot interface certificate file to later compare the fingerprint with the presented preboot interface certificate before [unlocking the instance](#).
 - For an externally signed certificate, convert a copy of the certificate you uploaded in the previous step to DER format. Provide this DER certificate to anyone unlocking the instance.
 - For the default self-signed certificate, download the self-signed certificate to a PEM file and then convert it to DER format. Provide this file to anyone unlocking the instance.

```
ksctl interfaces certificate get --name preboot --icertfile verifypreboot.pem
openssl x509 -in verifypreboot.pem -outform der -out verifypreboot.der
```

4. Reboot the instance.

To check encryption status

To check the encryption progress, you can run following CLI command:

```
$ ksctl diskenc status -p
```

This command might time out during system restart or due to a slow connection. As an alternative, you can view the [Console](#) for the instance to see disk encryption progress.

Example:

```
$ ksctl diskenc status -p
```

This returns the following response:

```
Encrypting...
14.81 GiB / 15.52 GiB [=====>-----] 95.44% 11s
```

The instance starts up after the encryption has finished. You do not need to unlock the disk on this startup after the initial encryption.

To unlock an encrypted instance

Every time an encrypted instance boots, you must unlock the instance to allow admins and users access to most Virtual CipherTrust Manager interfaces. To unlock the instance, you can provide the private key in OpenSSH, PKCS1, and PKCS8 format. We recommend the following practices when you unlock the instance for increased security:

- Independently obtain the preboot interface certificate out-of-band before rebooting. In general, the person who first **encrypted the disk** can provide the certificate file. Compare the fingerprint of the independently provided certificate with the returned preboot interface certificate fingerprint before unlocking.
- Enable TLS verification of the presented preboot interface certificate.

The following examples show unlocking examples, presented from most secure unlocking to least secure.

- Run the following commands to unlock the disk while verifying an externally signed preboot interface certificate.
 - a. Obtain the preboot interface certificate available on port 443 in PEM format.

```
echo | openssl s_client -connect <ciphertrust_manager_ip>:443 | sed -n '/BEGIN CERTIFICATE/,/END CERTIFICATE/p' > <certificate-file>.pem
```

- b. Convert the downloaded interface certificate to DER format.

```
openssl x509 -in <certificate_file>.pem -outform der -out preboot.der
```

- c. View the fingerprint for the downloaded preboot interface certificate.

```
openssl dgst -sha256 -c preboot.der
```

- d. Inspect the fingerprint for the independently obtained certificate file. Make sure the two certificate fingerprints match. The example command works for a DER encoded certificate file.

```
openssl dgst -sha256 -c <certificate-name>.der
```

Tip

If you have SSH or console access to the CipherTrust Manager instance, you can also view the expected preboot certificate fingerprint there. The fingerprint is displayed during boot up.

- e. Unlock the disk.

```
ksctl diskenc secureboot -i <private ssh key for the instance> --trusted-ca-file <filename_for_the_trusted_CA_that_issued_the_preboot_certificate> --server-name <hostname_on_preboot_certificate>
```

Tip

Alternatives to including the --server-name flag are modifying the `/etc/hosts` file on the client workstation to include the hostname or mapping this hostname on your network's DNS server.

- Run the following commands to unlock the disk while verifying a self-signed preboot interface certificate.
 - a. Obtain the preboot interface certificate available on port in PEM format.

```
echo | openssl s_client -connect <ciphertrust_manager_ip>:443 | sed -n '/BEGIN CERTIFICATE/,/END CERTIFICATE/p' > preboot.pem
```

- b. Convert the downloaded interface certificate to DER format.

```
openssl x509 -in preboot.pem -outform der -out preboot.der
```

- c. View the fingerprint for the downloaded preboot interface certificate.

```
openssl dgst -sha256 -c preboot.der
```

- d. Inspect the fingerprint for the independently obtained certificate file. Make sure the two certificate fingerprints match.

```
openssl dgst -sha256 -c verifypreboot.der
```

Tip

If you have SSH or console access to the CipherTrust Manager instance, you can also view the expected preboot certificate fingerprint there. The fingerprint is displayed during boot up.

- e. Unlock the disk.

```
ksctl diskenc secureboot -i <private ssh key for the instance> --server-name  
preboot.ciphertrustmanager.local
```

Tip

Alternatives to including the `--server-name` flag are modifying the `/etc/hosts` file on the client workstation to include `preboot.ciphertrustmanager.local` or mapping this hostname on your network's DNS server.

- Run the following command to unlock the disk without verifying the preboot interface certificate.

```
ksctl diskenc secureboot -i <private ssh key for the instance> --noSSLverify
```

Scheduler

Note

This feature is not included in Community Edition and requires a valid Virtual CipherTrust Manager license. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

The CipherTrust Manager can periodically run key rotation, database backup, and user password expiry notification jobs in the background. These jobs can be:

- Configured using the scheduler
- Scheduled by creating a job configuration

A job configuration defines what job is run and when. Job configurations can be read, listed, updated, and deleted. Jobs are run after job configurations are created. The jobs that were run can also be listed.

In a clustered environment, by default the scheduled job can run on any cluster node. There is an option in the API and CLI to restrict which nodes can run the scheduled job.

This section provides examples of scheduler configurations using the ksctl utility. Refer to the CLI or API documentation for details.

Default Schedulers

There are two system-generated schedulers, `system_auto_key_rotation` and `system_auto_user_password_expiry_notification`.

Key Rotation Scheduler

The system-generated scheduler `system_auto_key_rotation` is disabled by default. When this scheduler is enabled, it runs every day at midnight, and rotates the keys with a rotation date less than the current time. To enable the scheduler, run:

```
ksctl scheduler configs modify key-rotation --id <ID> --disabled=false
```

Example

```
ksctl scheduler configs modify key-rotation --id e08708b9-c6f0-49f2-bbea-64bd6ee80779 --disabled=false
```

Note

The scheduler that rotates the key (`system_auto_key_rotation`) based on `scheduledRotationDate` is one per domain and it comes disabled by default. Therefore, you need to enable this scheduler in relevant domains after migration.

Keys are rotated based on the value configured in the `scheduledRotationDate` parameter. You can configure this value by setting the `rotation-frequency-days` parameter during key create or key modify request.

The value of the `rotation-frequency-days` parameter is defined in number of days, and should be *greater than or equal to 0*. The value of `scheduledRotationDate` is calculated based on key creation time + `rotation-frequency-days`.

When the scheduler runs, it only rotates the keys with a `scheduledRotationDate` before the current time.

For example, if `rotation-frequency-days` is set to 5 days for any key, the scheduler rotates that key every 5 days.

If `rotation-frequency-days` is set to "0", the key is not rotated. For more details, refer to [Configuring Key Rotation Frequency Days](#).

Syntax (Key create)

```
ksctl keys create --name <key-name> --rotation-frequency-days <number-of-days>
```

Syntax (Key modify)


```
ksctl keys modify --name <key-name> --rotation-frequency-days <number-of-days>
```

User Password Expiry Notification Scheduler

`system_user_password_expiry_notification` is enabled by default. It runs every day at midnight, and sends password expiry reminder emails if other conditions are met. Refer to the [password policy](#) page for details on these conditions.

Schedule Types

Schedules are used to execute [key rotation](#), [CRL synchronization](#), and [backup](#) jobs periodically.

Synchronize CRLs Scheduler

CRL is a list of certificates that have been revoked by the CA before their scheduled expiration date and should no longer be trusted.

In the previous releases, the CipherTrust Manager would only update the CRL when it reached its expiration. The most recent CRL file downloaded by the CipherTrust Manager is stored in the cache till the time specified in the *Next Update* field for that CRL.

If a certificate that was initially valid and not included in the cached CRL is revoked, you may need to wait until the end of the time specified in the *Next Update* field to obtain the updated CRL. For more details, refer to [OCSP and CRL caching](#).

However, starting from this release, the CipherTrust Manager allows you to fetch the most recent list of revoked certificates (CRL) from issuing authorities by using the Synchronize CRLs Scheduler.

Note

The Synchronize CRLs Scheduler functionality is currently available only for *WEB* and *REST* interfaces and *does not* apply to NAE and KMIP.

Using GUI

To add CRL schedule:

1. Log on to the CipherTrust Manager GUI.
2. Go to **Admin Settings > Schedules**.
3. Click **Add Schedule**. The Add New Schedule wizard is displayed.
4. Select **Synchronize CRLs** radio button. Click **Next**.
5. Provide **Schedule Name** and **Description**.
6. Select **Enable Schedule** check box so that backup can be scheduled. Click **Next**.

7. Add Schedule Configuration. Refer to [Time Specification](#) for details.

8. Click **Add Schedule**.

You can see the created CRL schedule on the dashboard.

Schedules

Name

Search

2 Results | 2 schedules

Add Schedule

Name	Schedule Type	Last Modified	Run At	Start Date	End date	Status
refresh-crl	Synchronize CRLs	14 Jan 2025, 16:41	At 12:00 AM UTC		14 Jan 2025, 16:26	Never Enabled
system_auto_key_rotation	Key Rotation	14 Jan 2025, 14:43	At 12:00 AM UTC			Never Disabled

2 schedules10 per page

Using CLI

To create CRL schedule:

```
ksctl scheduler configs create sync-crl --name <id_of_job_config> --run-at <Job-execution-details> --description <new_description>
```

Example

To create a schedule that renews all the cached CRLs every Monday at 9 AM:

```
ksctl scheduler configs create sync-crl --name refresh-crl --run-at '0 9 * * 1'
```

To modify CRL schedule:

```
ksctl scheduler configs modify sync-crl --id <id_of_job_config> --description <new_description> --run-at <job-execution-details>
```

Example

```
ksctl scheduler configs modify --id e8a3bb32-dcbe-42a8-ad28-8674df108281 --description 'sync crl job at 9 AM'
```

Time Specification

The time that a job runs is described in the `run-at` parameter using the cron expression format: `"* * * * *"`

These five values indicate when the job should be executed. They are in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

Field Name	Mandatory	Allowed Values	Allowed Special Characters
Minute	Yes	0-59	* / , -
Hour	Yes	0-23	* / , -
Day of month	Yes	1-31	* / , -?
Month	Yes	1-12 or JAN-DEC	* / , -
Day of week	Yes	0-6 or SUN-SAT	* / , -?

Note

Month and **Day of week** field values are case *insensitive*. For example, "SUN", "Sun", and "sun" are equally accepted.

Note

Scheduler itself is a generic service, so it provides granularity of a minute. A job configuration should take the nature of operation information into consideration. For example, rotating the same set of multiple keys and backing up a database every minute are highly unlikely use cases.

Allowed Special Characters

The following table describes the use of special characters in the cron expression format:

Special Character	Description
Asterisk *	Indicate that the cron expression will match for all values of the field. For example, using an asterisk in the 5th field (month) would indicate every month.
Slash /	Describe increments of ranges. For example, 3-59/15 in the 1st field (minutes) would indicate the 3rd minute of the hour and every 15 minutes thereafter.
Comma ,	Separate items of a list. For example, using "MON,WED,FRI" in the 5th field (day of week) would mean Mondays, Wednesdays, and Fridays.
Hyphen -	Define ranges. For example, 9-17 would indicate every hour between 9am and 5pm inclusive, MON-WED would indicate Monday through Wednesday, and JAN-JUN would indicate every month from January through June.
Question mark ?	Use instead of "*" for leaving either day-of-month or day-of-week blank. "?" means no specific value. For example, to run the job every 10th day of the month and not limit the job to a particular day of the week: "* * ?

Special Character	Description
	10 * ?".

Permissions

You require specific [group](#) memberships to manage particular job configuration types. As well, you need to be logged into the root domain.

Job Configuration	Required Group	Required Domain
Key rotation	Key Admin	
System Backup	Backup Admin	
Domain Backup	Domain Backup Admin	
User Password Expiry Notification	admin (application administrators)	

Creating Job Configurations

To create a job configuration, run the command `ksctl scheduler configs create <operation> --name <config-name> --run-at "<cron-expression-format>"`.

Here, required values are:

- `<operation>`: Operation to schedule. The options are:
 - `backup`: Create a job configuration for backup operation. Refer to [Scheduling Backups](#) for more details and options.
 - `key-rotation`: Create a job configuration for key rotation. Refer to [Scheduling Key Rotation](#) for more details and options.
- `<config-name>`: Name for the job configuration.
- `<cron-expression-format>`: Time when the job runs. The format must have *five* fields. If the number of fields is not equal to five, the format becomes invalid. Refer to [time specification](#) for details.

Additional required values are specific to the operation.

- Refer to [Scheduling Backups](#) for options specific to backups.
- Refer to [Scheduling Key Rotation](#) for options specific to key rotation.

Viewing Existing Job Configurations

To view the list of job configurations, run the command `ksctl scheduler configs list`.

Example:

```

ksctl scheduler configs list
{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "id": "20c5263d-e296-4994-9129-b71aff49f4b3",
      "uri": "kylo:kylo:scheduler:job_configs:20c5263d-e296-4994-9129-b71af
f49f4b3",
      "account": "kylo:kylo:admin:accounts:kylo",
      "application": "ncryptify:gemalto:admin:apps:kylo",
      "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
      "createdAt": "2020-04-12T06:07:40.054712Z",
      "name": "ab-key-rot-config",
      "updatedAt": "2020-04-12T06:07:40.054712Z",
      "description": "",
      "operation": "key_rotation",
      "run_at": "45 11 * * 0",
      "run_on": "any",
      "disabled": false
    },
    {
      "id": "fd0c27b8-a502-47cc-b6bc-4c1c6fbbe89a",
      "uri": "kylo:kylo:scheduler:job_configs:fd0c27b8-a502-47cc-b6bc-4c1c6
fbbe89a",
      "account": "kylo:kylo:admin:accounts:kylo",
      "application": "ncryptify:gemalto:admin:apps:kylo",
      "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
      "createdAt": "2020-04-12T06:06:15.110531Z",
      "name": "ab-backup-config",
      "updatedAt": "2020-04-12T06:06:15.110531Z",
      "description": "",
      "operation": "database_backup",
      "run_at": "0 12 * * 0",
      "run_on": "any",
      "disabled": false
    }
  ]
}

```

The output shows existing job configurations.

Viewing Details of Job Configurations

To view the details of a job configuration, run the command `ksctl scheduler configs get --id "configID"`. Here, `<configID>` represents the ID of the configuration.

Example

```

ksctl scheduler configs get --id "fd0c27b8-a502-47cc-b6bc-4c1c6fbbe89a"
{
  "id": "fd0c27b8-a502-47cc-b6bc-4c1c6fbbe89a",
  "uri": "kylo:kylo:scheduler:job_configs:fd0c27b8-a502-47cc-b6bc-4c1c6fbbe89
a",
  "account": "kylo:kylo:admin:accounts:kylo",

```

```

    "application": "ncryptify:gemalto:admin:apps:kylo",
    "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
    "createdAt": "2020-04-12T06:06:15.110531Z",
    "name": "ab-backup-config",
    "updatedAt": "2020-04-12T06:06:15.110531Z",
    "description": "",
    "operation": "database_backup",
    "run_at": "0 12 * * 0",
    "run_on": "any",
    "disabled": false
  }

```

Modifying Job Configurations

Existing job configurations can be modified when needed. To modify a job configuration, run the command:

```
ksctl scheduler configs modify <operation> --id "<configID>" <args>
```

Here,

- **<operation>**: Operation can be `backup` or `key-rotation`.
- **<configID>**: ID of the job configuration to be modified.
- **<args>**: Parameters to modify.

Example Request 1

```
ksctl scheduler configs modify key-rotation --id "20c5263d-e296-4994-9129-b71aff49f4b3" --run-at "50 11 * * 0"
```

Example Response 1

```

{
  "id": "20c5263d-e296-4994-9129-b71aff49f4b3",
  "uri": "kylo:kylo:scheduler:job_configs:20c5263d-e296-4994-9129-b71aff49f4b3",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2020-04-12T06:07:40.054712Z",
  "name": "ab-key-rot-config",
  "updatedAt": "2020-04-12T06:14:57.23658004Z",
  "description": "",
  "operation": "key_rotation",
  "run_at": "50 11 * * 0",
  "run_on": "any",
  "disabled": false
}

```

Example Request 2 (update `replaced-key-state` value from `Deactivated` to `ProtectStop`)

```
ksctl scheduler configs modify key-rotation --id d0c48426-f5cc-416b-b935-e91b5872
```

```
db8a --replaced-key-state "ProtectStop"
```

Example Response 2

```
{
  "id": "d0c48426-f5cc-416b-b935-e91b5872db8a",
  "uri": "kylo:kylo:scheduler:job_configs:d0c48426-f5cc-416b-b935-e91b5872db8a",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-04-27T04:39:57.84822Z",
  "name": "test2",
  "updatedAt": "2023-04-27T04:52:36.182154615Z",
  "description": "",
  "operation": "key_rotation",
  "run_at": "0 1 * * *",
  "run_on": "any",
  "disabled": false,
  "job_config_params": {
    "change_state_after_time": 3,
    "replaced_key_state": "ProtectStop"
  }
}
```

Example Request 3 (update `change-state-after-time` to some valid value(>=0))

```
ksctl scheduler configs modify key-rotation --id d0c48426-f5cc-416b-b935-e91b5872db8a --change-state-after-time 10
```

Example Response 3

```
{
  "id": "d0c48426-f5cc-416b-b935-e91b5872db8a",
  "uri": "kylo:kylo:scheduler:job_configs:d0c48426-f5cc-416b-b935-e91b5872db8a",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-04-27T04:39:57.84822Z",
  "name": "test2",
  "updatedAt": "2023-04-27T04:58:36.855601961Z",
  "description": "",
  "operation": "key_rotation",
  "run_at": "0 1 * * *",
  "run_on": "any",
  "disabled": false,
  "job_config_params": {
    "change_state_after_time": 10,
    "replaced_key_state": "Deactivated"
  }
}
```

Manually Running Job Configurations

A scheduled configuration can be run manually when needed. To manually run a job configuration, run the command: `ksctl scheduler configs run-now --id "<configID>"`

Here, `<configID>` represents the ID of the configuration. The command creates a new job and waits for the job to be posted on the job queue. The `"job_id"` is returned if the job is posted within about 30 seconds.

Example:

```
ksctl scheduler configs run-now --id "e90cb2df-7df5-487a-ab5e-2c0ead7405c4"
{
  "job_id": "1e05dfb0-a5f9-48f6-afdf-406c368d2847",
  "job_config_id": "e90cb2df-7df5-487a-ab5e-2c0ead7405c4"
}
```

Disabling Job Configurations

You can disable an existing job configuration to temporarily prevent future job runs. To disable job configuration, run the command: `ksctl scheduler configs modify <operation> --id "<configID>" --disabled`

Here,

- `<operation>`: Operation can be `backup`, `key-rotation`, or `user-password-expiry-notification`.
- `<configID>`: ID of the job configuration to be modified.
- `disabled`: A boolean value indicated whether the job configuration is disabled, and will not run in the future.

Example Request

```
ksctl scheduler configs modify user-expiry-notification --id "20c5263d-e296-4994-9129-b71aff49f4b3" --disabled
```

Example Response

```
{
  "id": "20c5263d-e296-4994-9129-b71aff49f4b3",
  "uri": "kylo:kylo:scheduler:job_configs:20c5263d-e296-4994-9129-b71aff49f4b3",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2020-04-12T06:07:40.054712Z",
  "name": "ab-key-rot-config",
  "updatedAt": "2020-04-12T06:14:57.23658004Z",
  "description": "",
  "operation": "key_rotation",
  "run_at": "50 11 * * 0",
  "run_on": "any",
}
```



```
    "disabled": true
  }
```

To enable a disabled job configuration, run the command: `ksctl scheduler configs modify <operation> --id "<configID>" --disabled=false`

Example Request

```
ksctl scheduler configs modify user-password-expiry-notification --id "20c5263d-e296-4994-9129-b71aff49f4b3" --disabled=false
```

Deleting Job Configurations

Caution

This is a permanent operation. If you wish to temporarily suspend a scheduled job, you can [disable](#) it instead.

To delete a job configuration, run the command `ksctl scheduler configs delete --id "<configID>"`. Here, `<configID>` is the ID of the job configuration to be deleted. The specified job configuration is deleted without any message.

After a configuration is deleted, it will no longer run jobs automatically.

Job Runs

A job is created and run automatically when a scheduler configuration is run manually (using `run-now`) or automatically at a schedule defined in the job configuration. You can view the results of jobs executed in the past, including status and error count.

Viewing Job Runs

To view the list of already run jobs for the logged in account, run the command `ksctl scheduler jobs list`.

Example:

```
ksctl scheduler jobs list
{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "id": "5ce37ebd-f43f-4a97-bbaf-d6a7416074a7",
      "uri": "kylo:kylo:scheduler:jobs:5ce37ebd-f43f-4a97-bbaf-d6a7416074a7",
      "account": "kylo:kylo:admin:accounts:kylo",
    }
  ]
}
```

```

    "application": "ncryptify:gemalto:admin:apps:kylo",
    "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
    "createdAt": "2020-04-15T02:25:01.749924Z",
    "name": "rotate AES keys weekly",
    "updatedAt": "2020-04-15T02:25:14.003393Z",
    "job_config_id": "cbf85c25-08e9-4932-93da-a1042aafafdd",
    "description": "This is to rotate AES keys every Sunday, at 02:00",
    "operation": "key_rotation",
    "job_config_params": {
      "meta": {
        "naeState": "Restricted"
      },
      "query": {
        "algorithm": "AES"
      },
      "deactivate_replaced_key": 30
    },
    "enqueued_at": "2020-04-15T02:25:01.749791Z",
    "status": "completed",
    "processing_node": "local-node",
    "details": {
      "job_id": "5ce37ebd-f43f-4a97-bbaf-d6a7416074a7",
      "job_config_id": "cbf85c25-08e9-4932-93da-a1042aafafdd",
      "name": "rotate AES keys weekly",
      "operation": "key_rotation",
      "status": "completed",
      "percent_done": 100,
      "details": {
        "num_rotated": 1
      }
    }
  },
  {
    "id": "5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",
    "uri": "kylo:kylo:scheduler:jobs:5f2cf5fb-94ef-419a-9cf6-4f4045d43dc
1",
    "account": "kylo:kylo:admin:accounts:kylo",
    "application": "ncryptify:gemalto:admin:apps:kylo",
    "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
    "createdAt": "2020-04-15T02:09:02.740656Z",
    "name": "rotate AES keys",
    "updatedAt": "2020-04-15T02:09:13.978049Z",
    "job_config_id": "61bf1c6f-flb7-4ef9-b097-af8304647ecf",
    "description": "This is to rotate AES keys",
    "operation": "key_rotation",
    "job_config_params": {
      "meta": {
        "naeState": "Restricted"
      },
      "query": {
        "algorithm": "AES"
      },
      "deactivate_replaced_key": 30
    },
    "enqueued_at": "2020-04-15T02:09:02.740554Z",
    "status": "completed",
    "processing_node": "local-node",
    "details": {
      "job_id": "5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",

```

```

        "job_config_id": "61bf1c6f-f1b7-4ef9-b097-af8304647ecf",
        "name": "rotate AES keys",
        "operation": "key_rotation",
        "status": "completed",
        "percent_done": 100,
        "details": {
            "num_rotated": 1
        }
    }
}
]
}

```

The output shows that two jobs have run successfully.

Viewing Details of a Particular Job Run

To view the details of a particular job run for the logged in account, run the command `ksctl scheduler jobs get --id`. Here, `--id` is the ID of the job run you are searching for.

Example:

```

ksctl scheduler jobs get --id "5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1"
{
  "id": "5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",
  "uri": "kylo:kylo:scheduler:jobs:5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2020-04-15T02:09:02.740656Z",
  "name": "rotate AES keys",
  "updatedAt": "2020-04-15T02:09:13.978049Z",
  "job_config_id": "61bf1c6f-f1b7-4ef9-b097-af8304647ecf",
  "description": "This is to rotate AES keys",
  "operation": "key_rotation",
  "job_config_params": {
    "meta": {
      "naeState": "Restricted"
    },
    "query": {
      "algorithm": "AES"
    },
    "deactivate_replaced_key": 30
  },
  "enqueued_at": "2020-04-15T02:09:02.740554Z",
  "status": "completed",
  "processing_node": "local-node",
  "details": {
    "job_id": "5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",
    "job_config_id": "61bf1c6f-f1b7-4ef9-b097-af8304647ecf",
    "name": "rotate AES keys",
    "operation": "key_rotation",
    "status": "completed",
    "percent_done": 100,
    "details": {
      "num_rotated": 1
    }
  }
}

```

```

    }
  }
}

```

Deleting a Job

To delete a job, run the command `ksctl scheduler jobs delete --id`. This command stops the job (if it is running) and deletes it without any confirmation message.

Connection Manager

The Connection Manager contains a list of all connections to the resources that are external to the CipherTrust Manager server. Any resource that you intend to use with the CipherTrust Manager and that resides outside of the CipherTrust Manager infrastructure has to be added using the Connection Manager.

Note

- The **Connection Admins** can add, edit, delete, or test a connection.
- The **CCKM Admins** can *only* test a connection.

Accessing the Connection Manager

To access the Connection Manager, log in to the CipherTrust Manager as administrator. Next, click **Access Management** on the main screen, and then select **Connections** from the sidebar on the left.

The **Connections Management** screen is displayed. It is divided into:

- **CONNECTIONS** - allows you to configure the connections
- **INTERNAL CONNECTIONS** - allows you to configure the servers required for a connection

Internal Connections

The tabular view lists all the currently configured servers. You can arrange the list in different orders, by clicking on the column headers to sort it by that column. The table has the following columns:

- **Hostname** - hostname of the server
- **Description** - description of the server
- **Created** - date when the server was created
- **Service** - type of service
- **Products** - name of the product that uses the server

The last column contains an ellipsis icon (...). When clicked, it displays a menu that allows you to perform the following operations on the existing servers:

- **Delete** - delete the server
- **Download Server Cert** - download the server certificate

Click the **Download Luna Client Cert** button to download the certificate of the Luna client registered with Luna HSM.

Use the filters in the column headers to filter through multiple servers and display only those that you wish to display.

Use the **Search** box to search for a specific server.

Refer to [Adding an Internal Connection \(Server\)](#) to add the servers.

Connection Manager Certificate Expiration Check

The CipherTrust Manager inspects the expiration date of the certificates used in the configured connections everyday, at a preset system time to log the record. This is done for the following connections:

- Microsoft Azure
- DSM Connection
- Hadoop
- Salesforce
- Luna Network HSM

The CipherTrust Manager then creates list of certificates based on their expiration date:

- Certificates whose expiration dates are within 91 days.

This list is logged in the **Records** section once every week.

- Certificates whose expiration dates are within 7 days.

This list is logged in the **Records** section once every day.

- Certificates that are already expired.

This list is logged in the **Records** section once every day.

You can also create alarm triggers for these records. For more details, go to [Creating Alarm Trigger for Connection Certificate Expiration](#).

Connections

Users in the [Connection Admins](#) and 'admins' group can add, edit, delete, or test a connection through Connection Manager menus.

The tabular view lists all the currently configured connections. You can arrange the list in different orders, by clicking on the column headers to sort it by that column. The table has the following columns:

- **Name** - name of the connection
- **Creation** - date when the connection was created
- **Type** - type of connection
- **Products** - name of the product that uses the connection
- **State** - state that the connection is in. It is one of these states:
 - Not tested - connection has not been tested
 - Fail - connection has been tested and the test failed. The date when the connection failed is displayed to the right
 - Ready - connection is properly configured and ready

The last column contains an ellipsis icon (...). When clicked, it displays a menu that allows you to perform the following operations on the existing connections:

- **View/Edit** - view and edit the connection
- **Test Connection** - test the connection
- **Delete** - delete the connection

Use the filters in the column headers to filter through multiple connections and display only those that you wish to display.

Use the **Search** box to search for a specific connection.

Refer to [Adding a New Connection](#) to add a New connection.

External Certificate for Azure and Salesforce Connections

If you want to add an Azure or SFDC connection using external certificate, you first need to create and sign a Connection CSR using the CSR generator in CA.

To create a valid external certificate for connections through CipherTrust Manager GUI:

1. Sign in as a user in the [CA Admin](#) or [admin group](#), such as the root `admin` account.
2. Navigate to **CA > CSR Generator**.
3. Select **Connection CSR**.
4. Enter a **Common Name** as required.
5. Select a **Size**.

Azure Connections support 2048 and 4096.

Salesforce connections support 1024, 2048, and 4096.

6. Enter in any desired optional settings:

- Display Names
- Subject Alternative Names
 - DNS Names (comma separated)
 - IP Addresses (comma separated)
 - Email Addresses (comma separated)

7. If desired, enable **Encrypt Private Key**.

- a. In **Private Key Encryption**, select a key algorithm, `AES256`, `AES192`, `AES128`, or `TDES`. CipherTrust Manager will generate a new key with this algorithm to encrypt the private key.
- b. In **Private Key Encryption Password**, enter a password.

8. Click **Generate CSR**.

The Certificate Signing Request is displayed at the bottom of the page.

9. Click **Download CSR** to export the CSR contents to a file `CSR.pem`.

10. Have a [Local CA](#) or well-known External CA sign the certificate.

Note

As the purpose of the external certificate is authentication to a cloud service and not to CipherTrust Manager itself, there is no need to add the external CA to CipherTrust Manager.

11. Upload the external certificate to the cloud portal, Azure or Salesforce portal.

12. Store the external certificate in an accessible place. You will have to upload the certificate file to CipherTrust Manager when you create the Azure or Salesforce connection.

Adding a New Connection

Click the **+ Add Connection** button to open the **Add Connection** screen and complete the following steps:

1. [Select Connection Type](#)

2. General Info
3. Configure Connection
4. Add Products

1. Select Connection Type

In the **Select Category** section, select the **Cloud**, **HSM**, **Key Manager**, **OIDC/LDAP**, **TDP**, **SFTP/SCP**, **Log Forwarder**, or **More** option. Next, select a desired connection type from the **Select <Category> Type** dropdown.

- **Cloud:** Akeyless Gateway, AWS, Azure, Salesforce, SAP Data Custodian, Oracle Cloud Infrastructure, or Google. AWS, Azure, Salesforce, Google, and Oracle Cloud Infrastructure are cloud computing platforms and CipherTrust Cloud Key Manager (CCKM) manages cloud keys for these cloud services. The Akeyless Gateway is part of [Secrets Management](#), and connects to the Akeyless Vault Platform.
- **HSM:** Luna Network HSM. It allows the CCKM to manage and perform operations on the keys stored on HSM.
- **Key Manager:** DSM or CM Connection. The DSM connection provides a single point of authentication and access for DSMs in a cluster. The CM connection allows the CipherTrust Manager to act as an external key source in the connection manager.
- **OIDC/LDAP:** OIDC or LDAP Connection. OIDC is used to configure external identity providers for CTE agents. LDAP is used to configure LDAP connections, which can be used for LDAP browsing.

Note

OIDC connections to authenticate CipherTrust Manager users are configured [outside of connection manager](#).

- **TDP:** Hadoop Knox. It provides a single point of authentication and access for Hadoop services in a cluster.
- **SFTP/SCP:** SFTP/SCP . It helps to securely transfer system backup from CipherTrust Manager to the external servers.
- **Log Forwarder:** Loki, Elasticsearch, or Syslog. It helps forward server and client audit records and KMIP and NAE activity logs to Elasticsearch, Loki, or Syslog server.
- **More:** Lists the following connection type:
 - **CIFS/SMB:** It provides access to the shared files available in the network for CipherTrust Transparent Encryption (CTE).
 - **Attestation Authority:** It is used to configure connections to perform confidential computing.

Click **Next** to move to the next step.

2. General Info

In this step, provide a **Name** and **Description** (optional) for the new connection.

Click **Next** to move to the next step.

3. Configure Connection

The following table lists the available connections. Click the connection to know the configuration details.

AWS	Google	SFTP/SCP
Azure	Hadoop Knox	Loki
Salesforce	Luna Network HSM	Elasticsearch
SAP Data Custodian	CIFS/SMB	Syslog
Oracle Cloud Infrastructure	DSM Connection	OIDC
LDAP	CM Connection	Akeyless Gateway Connection
Attestation Authority		

4. Add Products

Use the check boxes in the **Products** list to select a product associated with the connection.

Note

The Akeyless Gateway connection is not associated with a product, so there is no **Add Products** selection.

- Data Discovery
- CTE
- Cloud Key Manager
- Backup/Restore
- HSM-anchored Domains

Click **Add Connection** to save your connection. The new connection is now listed in the **CONNECTIONS** list.

Labels in Connection Manager

Labels are key-value pairs used to group resources such as keys and certificate. Labels apply to AWS, Azure, GCP, and SCP connections in Connection Manager.

Specify labels as shown below:

```
"labels": {
  "key1": "value1",
  "key2": "value2"
}
```

A label can be defined with:

- **Key-value** pair
- **Key** with no value

A label must begin and end with an alphanumeric character (a-z, A-Z, 0-9). The remaining characters may be alphanumeric, dots (.), dashes (-), and underscores (_).

A label can also be specified by only providing a key with no value.

```
{ "critical": "" }
```

The following example shows how to define labels.:

```
"labels": {
  "sales.widgets.com/region": "noram",
  "team": "sales",
  "critical": ""
}
```

To remove a key-value pair, specify `null` in the value field.

For Example :

```
{ "key1": null }
```

Using labels in Connection Manager:

```
{
  "name": "aws-connection",
  "products": [
    "cckm"
  ],
  "labels": {
    "type": "aws"
  },
  "access_key_id": "Sample ID",
  "secret_access_key": "Sample Secret",
  "cloud_name": "aws"
}
```

Note

CipherTrust Manager uses a set of predefined labels for resource management. These labels are reserved and should not be explicitly used or removed by users. Each reserved label is prefixed with `ncryptify-reserved`, indicating its exclusive purpose within the system.

AWS

AWS connections to the CipherTrust Manager can be configured using the following:

- [Managing AWS Connections using GUI](#)
- [Managing AWS Connections using ksctl](#)

Note

Before adding a New AWS connection, ensure that time on the CipherTrust Manager and AWS is in sync. To change the time on the CipherTrust Manager, use the [NTP server](#) or run the `date` or `timedatectl` command.

When creating an AWS connection, you can specify the AWS Security Token Service (AWS STS) endpoints. By default, the AWS STS is available as a global service, and all AWS STS requests go to a single endpoint at <https://sts.amazonaws.com>. The global requests map to the US East (N. Virginia) Region, `us-east-1`. The valid values are:

- **Legacy** (default): Uses the global AWS STS endpoint, sts.amazonaws.com.
- **Regional**: The SDK or tool always uses the AWS STS endpoint for the currently configured Region. This is the recommended setting to reduce latency, build in redundancy, and increase session token validity.

To know more about AWS STS, refer to [Managing AWS STS in an AWS Region](#).

Note

- When the CipherTrust Manager is in **AWS VPC** (that is, connectivity to the global AWS STS endpoint is unavailable), for creating IAM Roles Anywhere connections, you need to create:
 - Endpoints for STS, EC2, KMS, and IAM role.
 - Regional AWS connection with the same region from where the CipherTrust Manager instance is launched.
- When connectivity to the global AWS STS endpoint is unavailable, set `aws_sts_regional_endpoints` to a reachable regional AWS STS endpoint.

Managing AWS Connections using GUI

To manage AWS connections using GUI, perform the following steps:

1. Log on to CipherTrust Manager UI as an administrator.

-
2. Navigate to **Access Management > Connections**.
 3. Click **Add Connection**.
 4. On the Add Connection screen, select category as **Cloud**.
 5. Select Select Cloud Type as **AWS** and click **Next**.
 6. Specify connection **Name** and **Description** and click **Next**.
 7. Configure the parameters and follow the steps defined in the below tabs.

Long-term Credentials

- **Access Key ID** - an access key is a long-term credential for an AWS account root user. An access key consists of two parts: an access key ID and a secret access key (pretty much like a user name and password). This is the user name part.
- **Secret Access Key** – this is the password part of the access key. Select the **Show Secret Access Key** check box to view the password as open text.
- **Cloud Name** - the name of the AWS cloud to connect to. Currently, only the following options are available:
 - AWS
 - AWS China
 - AWS-US-GOV
- **AWS STS Regional Endpoints** - Specify the [AWS STS endpoints](#).
- **Assume Role** - the AWS Assume Role. For more details, refer to the [AWS documentation](#).
- **Assume Role External ID** - the external ID for an IAM role in AWS (Assume Role). This is something that you may want to use when you need to give access to your AWS resources to a third party. For more details, refer to the [AWS documentation](#).

Short-term Credentials (IAM Roles Anywhere)

For IAM Roles Anywhere, select the **IAM Roles Anywhere** checkbox and provide the following information:

- **Cloud Name** - the name of the AWS cloud to connect to. Currently, only the following options are available:
 - AWS
 - AWS China
- **AWS STS Regional Endpoints** - Specify the [AWS STS endpoints](#).
- **Assume Role** - the AWS Assume Role. For more details, refer to the [AWS documentation](#).

-
- **Assume Role External ID** - the external ID for an IAM role in AWS (Assume Role). This is something that you may want to use when you need to give access to your AWS resources to a third party. For details, refer to [AWS documentation](#).

Click **Next** to proceed to **IAM Configuration** tab and provide the following details:

- **Certificate** - either upload the external certificate or paste the certificate content.
 - **File Upload** - select and click **Certificate Upload** to upload the certificate from your machine.
 - **Text** - select and paste the certificate content in the text field.
 - **Private Key** - specify the private key associated with the certificate.
 - **Role** - specify AWS IAM Anywhere Role ARN.
 - **Profile** - specify AWS IAM Anywhere Profile ARN.
 - **Trust Anchor** - specify AWS IAM Anywhere Trust Anchor ARN.
8. Click **Test Credentials** to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.
9. Click **Next** to move to the [Add Products](#) screen of the Add Connection wizard.

Managing AWS Connections using ksctl

When you interact with AWS, you specify your AWS security credentials to verify who you are and whether you have permission to access the resources that you are requesting. AWS uses the security credentials to authenticate and authorize your requests.

There are different types of users in AWS. All AWS users have security credentials. There is the account owner (root user), users in AWS IAM Identity Center (successor to AWS Single Sign-On), federated users, and IAM users.

Users can have either *long-term* or *temporary* security credentials. For more details, refer to [AWS security credentials](#).

The following operations can be performed:

- Create/Get/Update/Delete an AWS connection
- List all AWS connections
- Test an existing AWS connection
- Test the new AWS connection parameters before establishing the connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Name of the connection.
products	Optional	List of products.
clientid	Optional	Key ID of the AWS user.
secret	Optional	AWS client secret.
iamroleanywhere	Optional	Parameters required to create an IAM Anywhere Role connection in a json file.
isroleanywhere	Optional	Set the parameter to "true" to create or list connections of type AWS IAM Anywhere with temporary credentials. These parameters are: • anywhere_role_arn - AWS IAM Anywhere Role ARN. • certificate - The external certificate for AWS IAM Anywhere Cloud connections. • profile_arn - AWS IAM Anywhere Profile ARN. • trust_anchor_arn - AWS IAM Anywhere Trust Anchor ARN. • private_key - The private key associated with the certificate.
assumerolearn	Optional	Assume role amazon resource name (ARN). For more details, refer to the AWS documentation .
assumeroleexternalid	Optional	The external ID for an IAM role in AWS (Assume Role) is used when granting access to your AWS resources for a third party. For more details, refer to the AWS documentation .
cloudname	Optional	Name of the AWS cloud to connect to.
description	Optional	Connection description.
meta	Optional	meta information in json format. This information is provided in pair. For example, --meta <code>{"color\":"blue\","foo\":"bar\}"</code> .
json-file	Optional	Connection information provided in a JSON file format. Command line parameters will take precedence over values specified in the JSON file.

Creating an AWS Connection

Long-term Credentials

In AWS, the Root user, IAM user, and access keys have long-term security credentials that do not expire. Long-term credentials are persistent and can be used to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs. It's important to note that these credentials have significant power and should be protected and managed carefully, as they provide full access to the associated AWS account.

When creating an AWS connection, you can specify the [AWS Security Token Service \(AWS STS\) endpoints](#).

Syntax

```
ksctl connectionmgmt aws create --name <connection-name> --clientid <access-key-id> --secret <aws-client-secret> --assumerolearn <assume-role-arn> --assumeroleexternalid <assume-role-external-id> --cloudname <cloud-name> --description <description> --meta <key:value> --products <products-names> --json-file <connection-information-in-JSON-format-via-file>
```

Example

```
ksctl connectionmgmt aws create --name aws-connection --clientid AKIAJBLNQGTQFOWDRSRQ --secret DHZc9bi/rrJZW3PRfWrkgn3KYVMDyI1bnedKPmlwU
```

Response

```
{
  "id": "7b736162-3330-49f3-b631-044e107df033",
  "uri": "kylo:kylo:connectionmgmt:connections:aws-connection-7b736162-3330-49f3-b631-044e107df033",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-08-17T08:46:58.29948682Z",
  "channel": "",
  "updatedAt": "2022-08-17T08:46:58.296588992Z",
  "service": "aws",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "aws-connection",
  "access_key_id": "AKIAJBLNQGTQFOWDRSRQ",
  "cloud_name": "aws",
  "aws_sts_regional_endpoints": "legacy",
  "aws_region": "us-east-1"
}
```

Short-term Credentials (IAM Roles Anywhere)

The AWS Identity and Access Management (IAM) Roles Anywhere enables obtaining temporary security credentials in IAM for AWS resources such as servers, containers, and applications. The temporary security credentials are short-term and not stored with the user but are generated dynamically and provided to the user when requested. This simplifies the signing process for the applications. For more details, refer to [What is AWS Identity and Access Management Roles Anywhere & Temporary security credentials in IAM](#). To create an AWS connection of type IAM Roles Anywhere, run:

Syntax

```
ksctl connectionmgmt aws create --name <connection-name> --iamroleanywhere <path-to-json-file> --isroleanywhere true
```

The `iamroleanywhere` field defines the parameters required to create a connection of type IAM Roles Anywhere. These parameters are:

- `anywhere_role_arn` - AWS IAM Anywhere Role ARN.
- `certificate` - The external certificate for AWS IAM Anywhere Cloud connections.
- `profile_arn` - AWS IAM Anywhere Profile ARN.
- `trust_anchor_arn` - AWS IAM Anywhere Trust Anchor ARN.
- `private_key` - The private key associated with the certificate.

Note

When the `trust_anchor_arn` is created using a certificate issued from the CA of the CipherTrust Manager, `private_key` is not required while creating an AWS connection on the same CipherTrust Manager.

The `iamroleanywhere` fields are defined in the below *"iamanywhere.json"* file.

Set the `isroleanywhere` parameter to `true` to create connections of type AWS IAM Anywhere with temporary credentials.

Example

```
ksctl connectionmgmt aws create --name testconnection --iamroleanywhere iamanywhere.json --isroleanywhere true
```

Response

```
{
  "id": "740cff5c-44a1-4006-9e1c-d51e9ae0f4de",
  "uri": "kylo:kylo:connectionmgmt:connections:testcon-740cff5c-44a1-4106-9e1c-d51e9ae0f4de",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-05-05T06:24:18.164586919Z",
  "updatedAt": "2023-05-05T06:24:18.155392895Z",
  "service": "aws",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "testcon",
```



```
{
  "anywhere_role_arn": "arn:aws:iam::301888698005:role/RolesAnywhere_Test",
  "profile_arn": "arn:aws:rolesanywhere:us-east-1:306778600005:profile/0d90fe3
6-1977-4ff8-886d-8c84b5a922ed",
  "trust_anchor_arn": "arn:aws:rolesanywhere:us-east-1:300008697945:trust-anch
or/988e70c0-a0b9-4cb9-97fd-0b8ff46d53ad",
  "certificate": "-----BEGIN CERTIFICATE-----\nMIICATCCAYagAwIBAgIUVMghsopD+cgubEYjpvB96oOeShwwCgYIKoZIZj0EAwQw\nxNTELMaKGA1UEBhMCVVMxMzCzAJBgNVBAGTAK5ZMREwDwYDVQQHEWh0ZXcgcWw9yazEP\nnMA0GA1UEChMGVGhbbGVzMQwwCgYDVQQLEwNESVMxDzANBgNVBAMTB1RoYWxlczAe\nnFw0YmZAxMDMxMDE1NTBaFw0yNDAxMDMxMDE1NTBaMECzAJBgNVBAYTAlVTMQ8w\nnDQYDVQQKDAZUaGFsZXNMcEzARBgNVBAMMClRoYWxlcy5jb20xMzANBgNVBASMBkRJ\n\nu19EUzB2MBAGBqGSM49AgEGBSuBBAAiA2IABIt5e3BVMVAdNfTGslhkhhLlqJiA\n\nvnO0ECgAlC1TwzILZalsczZ0256PVWD9+q4DXRXVzUSKMHTTvD2dfxzU3Ze9AP92h\n\n/nz7PXR+AiBcyoz03kcqd4eInt8cVbI8iA0Yk1xqMGMB4wDAYDVR0TAAQH/BAIwADA
O\n\nBgNVHQ8BAf8EBAMCB4AwCgYIKoZIZj0EAwQDAQAwZgIxAO3YZ5spj2Nnkb5yycuIO\n\nnsz4CYbt5/Akw6XCikj676ExyhrLbHs3MnYocD1dqJIHbcgIxAPbMcFe0ntWfe+hm\n\n\nsGYqabQmzwJaewriq3qmmT\n\nhs8MPx8k7F9N6sKdiC7fI8d2XAsA==\n\n-----END CERTIFICATE-----",
  "private_key": "-----BEGIN EC PARAMETERS-----\nBgUrgQQAIG==\n\n-----END EC PARAMETERS-----\n\n-----BEGIN EC PRIVATEKEY-----\nMIGKAgeBBDBZ2diJpkmoQ5JGNjpHq0rdzKOFRDjP3nCSLhlZtrLDrhREY3CGiqX\n\n/nc/7hJFA49pugBwYFK4EEACKhZANiAASLeXtwVFQHTX0xrJYZIWY5aiYgLztBAOA\n\n\nnJQpU8MyC2WpbHM2dNuej1Vg/fquA10V1c1EijByU7w9nX8c1N2XvQD/doc+zl6/g\n\n\nnIgXMqM9N5HKneHiJ7fHFwYPIgNMpNcY=\n\n-----END EC PRIVATE KEY-----"
```

Syntax

```
ksctl connectionmgmt aws get --id <connection-name/Id>
```

Example

```
ksctl connectionmgmt aws get --id 5c7a0dab-6358-42dd-8809-257a96f0c010
```

Response

```
{
  "id": "5c7a0dab-6358-42dd-8809-257a96f0c010",
  "uri": "kylo:kylo:connectionmgmt:connections:aws-connection-5c7a0dab-6358-42d-8809-257a96f0c010",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-08-22T08:35:45.766632Z",
  "updatedAt": "2022-08-22T08:43:25.392906Z",
  "service": "aws",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "aws-connection",
  "access_key_id": "AKIAJBLNQGTQFOWDRSRQ",
  "cloud_name": "aws",
  "aws_sts_regional_endpoints": "legacy",
  "aws_region": "us-east-1"
}
```

Updating an AWS Connection

Note

When updating an AWS connection, you cannot update the connection type, that is, a connection generated with short-term credentials cannot be updated with the long-term credentials and vice versa.

Long-term Credentials

To update an AWS connection, run:

Syntax

```
ksctl connectionmgmt aws modify --id <connection-name/Id> --clientid <access-key-id> --secret <aws-client-secret> --assumerolearn <assume-role-arn> --assumeroleexternalid <assume-role-external-id> --cloudname <cloud-name> --description <description> --meta <key:value> --products <products-names> --json-file <connection-information-in-JSON-format-via-file>
```

Example

```
ksctl connectionmgmt aws modify --id 5c7a0dab-6358-42dd-8809-257a96f0c010 --clien
```

```
tid newclientid
```

Response

```
{
  "id": "5c7a0dab-6358-42dd-8809-257a96f0c010",
  "uri": "kylo:kylo:connectionmgmt:connections:aws-connection-5c7a0dab-6358-42d-8809-257a96f0c010",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-08-22T08:35:45.766632Z",
  "updatedAt": "2022-08-22T08:42:46.803919148Z",
  "service": "aws",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "aws-connection",
  "access_key_id": "newclientid",
  "cloud_name": "aws",
  "aws_sts_regional_endpoints": "legacy",
  "aws_region": "us-east-1"
}
```

Short-term Credentials (IAM Roles Anywhere)

To update an AWS connection of type IAM Roles Anywhere, run:

Syntax

```
ksctl connectionmgmt aws modify --id <connection-name/id> --iamroleanywhere <path-to-json-file>
```

Example

```
ksctl connectionmgmt aws modify --id 5da3c67e-8f28-4468-b733-709c0367cee2 --iamroleanywhere update.json
```

Response

```
{
  "id": "740cff5c-44a1-4006-9e1c-d51e9ae0f4de",
  "uri": "kylo:kylo:connectionmgmt:connections:testcon-740cff5c-44a1-4106-9e1c-d51e9ae0f4de",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-05-05T06:24:18.164586919Z",
  "updatedAt": "2023-05-05T06:24:18.155392895Z",
  "service": "aws",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "testcon",
  "cloud_name": "aws",
  "aws_sts_regional_endpoints": "legacy",
  "aws_region": "us-east-1",
}
```

```

    "is_role_anywhere": true,
    "iam_role_anywhere": {
      "certificate": "-----BEGIN CERTIFICATE-----\nMIICATCCAYagAwIBAgIUVMghsopD+cgu
bEYjpvB96oOeSHwwCgYIKoZIZj0EAwQw\nXTELMakGA1UEBhMCVVMxCzAJBgNVBAGTAk5ZMREwDwYDVQQ
HEwhOZXcgWW9yazEP\nMA0GA1UEChMGVzMqwwCgYDVQQLEwNESVMxDzANBgNVBAMTB1RoYWxlczAe\nnFw
0yMzAxMDMxMDE1NTBaFw0yNDExMDMxMDE1NTBaMEQxCzAJBgNVBAYTA1VTMQ8w\nnDQYDVQQKDAZUaGFsZ
XMxEzARBgNVBAMMC1RoYWxlcy5jb20xDzANBgNVBASMBkRJ\nnU19EUzB2MBAGByqGBSuBBAAiA2IABIt5
e3BVMVAdNfTGslhkhbLlqJiA\nnvO0ECgAlC1TwzILZalsczZ0256PVWD9+q4DXRXVzUSKMHJTvD2dfxzU
3Ze9AP92h\nnz7PXR+AiBcyoz03kcqd4eInt8cVbI8iA0yk1xqMgMB4wDAYDVR0TAQH/BAIwADA0\nnBgNV
HQ8BAf8EBAMCB4AwCgYIKoZIZj0EAwQDaQAwwZgIxAO3YZ5spj2Nnkb5ycuI0\nnsz4CYbt5/Akw6XCikj6
76ExyhRlBHs3MnYocDldqJIHbcgIxAPbMcFe0ntWfe+hm\nnsGYqabQmzwJaewriq3qmmThs8MPx8k7F9N
6sKdiC7fI8d2XAsA==\n-----END CERTIFICATE-----",
      "trust_anchor_arn": "arn:aws:rolesanywhere:us-east-1:306778697945:trust-anch
or/944e70c0-a0b9-4cb9-97fd-0b8ff46d43ad",
      "profile_arn": "arn:aws:rolesanywhere:us-east-1:306778697945:profile/0d90fe3
6-1977-4ff8-886d-8c84b5a922ed",
      "anywhere_role_arn": "arn:aws:iam::300000000000:role/RolesAnywhere"
    }
  }
}

```

update.json

```

{
  "anywhere_role_arn": "arn:aws:rolesanywhere:us-east-1:306778697945:profile/00
000000-0000-0000-0000-000000000000"
}

```

Deleting an AWS Connection

To delete an AWS connection, run:

Syntax

```
ksctl connectionmgmt aws get --id <connection-name/Id>
```

Example

```
ksctl connectionmgmt aws get --id 5c7a0dab-6358-42dd-8809-257a96f0c010
```

There will be no response if the AWS connection is deleted successfully.

Getting List of AWS Connections

Long-term Credentials

To list all the AWS connections, run:

Syntax

```
ksctl connectionmgmt aws list
```

Example

```
ksctl connectionmgmt aws list
```

Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "5c7a0dab-6358-42dd-8809-257a96f0c010",
      "uri": "kylo:kylo:connectionmgmt:connections:aws-connection-5c7a0dab-6358-42dd-8809-257a96f0c010",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2022-08-22T08:35:45.766632Z",
      "updatedAt": "2022-08-22T08:43:25.392906Z",
      "service": "aws",
      "category": "cloud",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "aws-connection",
      "access_key_id": "AKIAJBLfdgTQFOWDRSRQ",
      "cloud_name": "aws",
      "aws_sts_regional_endpoints": "legacy",
      "aws_region": "us-east-1"
    }
  ]
}
```

Short-term Credentials (IAM Roles Anywhere)

To list all the AWS connections of type IAM Anywhere, run:

Syntax

```
ksctl connectionmgmt aws list --isroleanywhere true
```

Example

```
ksctl connectionmgmt aws list --isroleanywhere true
```

Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "e0036191-7c11-4c18-9152-dcfc6ab423b7",
      "uri": "kylo:kylo:connectionmgmt:connections:demoiamanywhere-e0036191-7c11-4c18-9152-dcfc6ab423b7"
    }
  ]
}
```

```

1-7c11-4c18-9152-dcfc6ab423b7",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-05-08T08:49:23.134714Z",
  "updatedAt": "2023-05-08T08:49:23.133589Z",
  "service": "aws",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "testconnection",
  "meta": null,
  "cloud_name": "aws",
  "aws_sts_regional_endpoints": "legacy",
  "aws_region": "us-east-1",
  "is_role_anywhere": true,
  "iam_role_anywhere": {
    "certificate": "-----BEGIN CERTIFICATE-----\nMIICATCCAYagAwIBAgIU
VMghsopD+cgubEYjpvB96oOeSHwwCgYIKoZIzjEAAwQw\nXTELMakGA1UEBhMCVVMxMzA5BjBjNVBAMTAk5Z
MREwDwYDVQQHEwOZXcgWW9yazEP\nMA0GA1UEChMGVGVhbnGVzMQwwCgYDVQQLEwNESVMxMzA5BjBjNVBAM
TB1RoYWxlczAe\nFw0yMzAxMDMxMDE1NTBaFw0yNDAxMDMxMDE1NTBaMEQxCzBjNVBAYTA1VTMQ8w\nnDQ
YDVQQKDAZUaGFsZXNMcEzARBjNVBAMMClRoYWxlcy5jb20xMzA5BjBjNVBAMBAzBkRj\nU19EUzB2MBAGByqGS
M49AgEGBSuBBAAiA2IABIt5e3BVMVAdNfTGslhkhbLlqJiA\nnvO0ECgAlClTwzILZalsczZ0256PVWDq4
DXRXVzUSKMHJTvd2dfxzU3Ze9AP92h\nnz7PXR+AiBcyoz03kcqd4eInt8cVbI8ia0yk1xqMgMB4wDAYDV
R0TAQH/BAIwADAQ\nnBgNVHQ8BAf8EBAMCB4AwCgYIKoZIzj0EAwQDaQAwZgIxAO3YZ5spj2Nnkb5ycuI
0\nnsz4CYbt5/Akw6XCikj676ExyhRlBHs3MnYocD1dqJIHbcgIxAPbMcFe0ntWfe+hm\nnsGYqabQmzwJa
ewriq3qmmThs8MPx8k7F9N6sKdiC7fI8d2XAsA==\n-----END CERTIFICATE-----",
    "trust_anchor_arn": "arn:aws:rolesanywhere:us-east-1:30677869794
5:trust-anchor/944e70c0-a0b9-4cb9-97fd-0b8ff46d43ad",
    "profile_arn": "arn:aws:rolesanywhere:us-east-1:306778697945:prof
ile/0d90fe36-1977-4ff8-886d-8c84b5a922ed",
    "anywhere_role_arn": "arn:aws:iam::306778697945:role/RolesAnywher
e_Tester"
  }
}
]
}

```

Testing an Existing AWS Connection

To test an existing AWS connection, run:

Syntax

```
ksctl connectionmgmt aws test --id <connection-name/Id>
```

Example

```
ksctl connectionmgmt aws test --id 5c7a0dab-6358-42dd-8809-257a96f0c010
```

Response

```
{
  "connection_ok": true
}
```



```
KDAZUaGFsZXMxEzARBgNVBAMMC1RoYWxlcy5jb20xDzANBgNVBASMBkKJ\nU19EUzB2MBAGByqGSMAGEGB
SuBBAAiA2IABIt5e3BVMVAdNfTGslhkhbLlqJiA\nvO0ECgAlC1TwzILZalsczZ0256PVWD9+q4DXRXVz
USKMHJTvD2dfxzU3Ze9AP92h\nz7PXr+AiBcyoz03kcqd4eInt8cVbI8iA0yk1xqMgMB4wDAYDVR0TAQ
H/BAIwADA0\nBgNVHQ8BAf8EBAMCB4AwCgYIKoZIZj0EAQDaQAwZgIXAO3YZ5spj2Nnkb5ycuI0\nnsz4C
Ybt5/Akw6XCikj676ExyhRlBHs3MnYocD1dqJIHbcgIXAPbMcFe0ntWfe+hm\nsGYqabQmzwJaewriq3q
mmThs8MPx8k7F9N6sKdiC7fI8d2XAsA==\n-----END CERTIFICATE-----",
    "private_key": "-----BEGIN EC PARAMETERS-----\nBgUrgQQAig==\n-----END EC
PARAMETERS-----\n-----BEGIN EC PRIVATE KEY-----\nMIGkAgEBBDBZ2dIjpkmoQ5JGNjpHq0rd
z4KOFrbfbdfjP3nCSLhlZtrLDrhRCGiqX\nc/7hJFA49pugBwYFK4EEACKhZANiAASLeXtwVTFQHTX0xr
JYZIWY5aiYgLztBAoA\nJQpU8MyC2WpbHM2dNuej1Vg/fquA10V1c1EijByU7w9nX8c1N2XvQD/doc+z1
6/g\nIgxMQM9N5HKneHiJ7fHFWyPIgNMPNcY=\n-----END EC PRIVATE KEY-----"
  },
  "is_role_anywhere": true
}
```

Response

```
{
  "connection_ok": true
}
```

Google

Google connection to the CipherTrust Manager can be configured using the following:

- Managing Google Connections using GUI
- Managing Google Connections using ksctl

Warning

Thales strongly discourages creating a Google connection using a service account key file that grants permission to **root of trust keys**.

Managing Google Connections using GUI

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.
4. On the Add Connection screen, select category as **Cloud**.
5. Select Select Cloud Type as **Google** and click **Next**.
6. Specify connection **Name** and **Description** and click **Next**.

7. Configure the below parameters.
 - **Key File** - upload the key file (a JSON file) that you have got from the GCP console while creating the service account.
 - **Cloud Name** - select the **Google** from the drop-down list.
8. Click the **Test Credentials** button to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.
9. Click **Next** to move to the **Add Products** screen of the Add Connection wizard.

Note

Currently, the only product supported for Google connection is Cloud Key Manager.

Note

Service account keys are private keys that let you authenticate as a service account. To rotate a service account key, refer to [Service Account Key Rotation](#).

Managing Google Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete a Google connection
- List all Google connections
- Test an existing Google connection
- Test a New Google Connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Name of the connection.
products	Optional	List of products.
key-file	Mandatory	The content of a private key file of a Google service account. It is provided in a JSON format.

Parameter	Mandatory/Optional	Description
cloudname	Optional	Name of the Google cloud to connect to.
description	Optional	Connection description.
meta	Optional	meta information in json format. This information is provided in pair. For example, --meta <code>{"color\":"blue\","fo o\":"bar\}"</code> .
json-file	Optional	Connection information provided in a JSON file format. Command line parameters will take precedence over values specified in the JSON file.

Creating a Google Connection

To create a Google connection, run:

Syntax

```
ksctl connectionmgmt gcp create --name <Connection-Name> --key-file <Key-File-Path> --cloudname <Cloud-Name> --products <Product-Names> --meta <Key:Values>
```

Format of GCP Key File

```
{
  "type": "service_account",
  "project_id": "test",
  "private_key_id": "hbk0662522e157b8e39cc672108de25016d736y0",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDV7g0lBwL/XaBD\nbnbpKtMLzVb1NAHVcx3U\nnIUXcwn8mDT/aYWClnTDW7/ZwThnOsXS\nxbco68JdM2bpCS9nRqhYAlLb0eLM12pEU\nn59cqC1DjxsmVcmpabyi/726I\nn-----END PRIVATE KEY-----\n",
  "client_email": "test@some-project.iam.gserviceaccount.com",
  "client_id": "some-id",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://accounts.google.com/o/oauth2/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/test%40some-project.iam.gserviceaccount.com"
}
```

Example Request

```
ksctl connectionmgmt gcp create --name gcpConn --key-file gcp.json --products CCKM
```

Example Response

```
{
  "id": "047bcdcb-5bbe-4de8-85e2-1dc504d07c59",
  "uri": "kylo:kylo:connectionmgmt:connections:gcpconn-047bcdcb-5bbe-4de8-85e2-1dc504d07c59",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-04-01T04:56:28.5260642Z",
  "updatedAt": "2021-04-01T04:56:28.524593208Z",
  "service": "gcp",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "gcpConn",
  "products": [
    "CCKM"
  ],
  "cloud_name": "gcp",
  "client_email": "test@some-project.iam.gserviceaccount.com",
  "private_key_id": "y437c51g956b8ab4908yb41541262a2fa3b0f84f"
}
```

Getting Details of a Google Connection

To get details of a Google connection, run:

Syntax

```
ksctl connectionmgmt gcp get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt gcp get --id 047bcdcb-5bbe-4de8-85e2-1dc504d07c59
```

Example Response

```
{
  "id": "047bcdcb-5bbe-4de8-85e2-1dc504d07c59",
  "uri": "kylo:kylo:connectionmgmt:connections:gcpconn-047bcdcb-5bbe-4de8-85e2-1dc504d07c59",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-04-01T04:56:28.5260642Z",
  "updatedAt": "2021-04-01T04:56:28.5245932Z",
  "service": "gcp",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "gcpConn",
  "products": [
    "CCKM"
  ],
  "cloud_name": "gcp",
  "client_email": "test@some-project.iam.gserviceaccount.com",
  "private_key_id": "y437c51g956b8ab4908yb41541262a2fa3b0f84f"
}
```

Updating a Google Connection

To update a Google connection, run:

Syntax

```
ksctl connectionmgmt gcp modify --id <Connection-Name> --key-file <Key-File-Path> --cloudname <Cloud-Name> --products <Product-Names> --meta <Key:Values>
```

Example Request

```
ksctl connectionmgmt gcp modify --id 047bcdcb-5bbe-4de8-85e2-1dc504d07c59 --key-file gcp1.json
```

Example Response

```
{
  "id": "047bcdcb-5bbe-4de8-85e2-1dc504d07c59",
  "uri": "kylo:kylo:connectionmgmt:connections:gcpconn-047bcdcb-5bbe-4de8-85e2-1dc504d07c59",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-04-01T04:56:28.526064Z",
  "updatedAt": "2021-04-01T05:03:38.665326512Z",
  "service": "gcp",
  "category": "cloud",
  "last_connection_ok": true,
  "last_connection_at": "2021-04-01T05:00:03.806155Z",
  "name": "gcpConn",
  "products": [
    "CCKM"
  ],
  "meta": "",
  "cloud_name": "gcp",
  "client_email": "test@some-project.iam.gserviceaccount.com",
  "private_key_id": "y437c51g956b8ab4908yb41541262a2fa3b0f84f"
}
```

Deleting a Google Connection

To delete a Google connection, run:

Syntax

```
ksctl connectionmgmt gcp delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt gcp delete --id 047bcdcb-5bbe-4de8-85e2-1dc504d07c59
```

There will be no response if GCP connection is deleted successfully.

Getting List of Google Connections

To list all the Google connections, run:

Syntax

```
ksctl connectionmgmt gcp list
```

Example Request

```
ksctl connectionmgmt gcp list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "047bcdcb-5bbe-4de8-85e2-1dc504d07c59",
      "uri": "kylo:kylo:connectionmgmt:connections:gcpconn-047bcdcb-5bbe-4de8-85e2-1dc504d07c59",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2021-04-01T04:56:28.526696Z",
      "updatedAt": "2021-04-01T04:56:28.526696Z",
      "service": "gcp",
      "category": "cloud",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "gcpConn",
      "products": [
        "CCKM"
      ],
      "cloud_name": "gcp",
      "client_email": "test@some-project.iam.gserviceaccount.com",
      "private_key_id": "y437c51g956b8ab4908yb41541262a2fa3b0f84f"
    }
  ]
}
```

Testing an Existing Google Connection

To test an existing Google connection, run:

Syntax

```
ksctl connectionmgmt gcp test --id <Connection-Name/ID> --key-file <Key-File-Path>
```

Example Request

```
ksctl connectionmgmt gcp test --id 047bcdcb-5bbe-4de8-85e2-1dc504d07c59
```

Example Response

```
{
  "connection_ok": true
}
```

Testing a New Google Connection

To test a New Google connection, run:

Syntax

```
ksctl connectionmgmt gcp test --key-file <Key-File-Path>
```

Example Request

```
ksctl connectionmgmt gcp test --key-file gcp.json
```

Example Response

```
{
  "connection_ok": true
}
```

Service Account Key Rotation

Rotating service account keys can help reduce the risk posed by leaked or stolen keys. To rotate the service account keys, perform the following steps:

On Google

1. Identify the service account key that needs to be rotated.
2. Create a new key for the same service account handling the connection between CipherTrust Manager and Google.

At this stage, the Google cloud contains two keys: the new and the old one.

On the CipherTrust Manager

1. Replace the existing (old) service account key with the new key in the Google connection manager. To do so, either go to [GUI](#) and upload the new "Key File" or use `ksctl` to modify the `key-file` parameter value.
2. [Test the connection](#). The state of the connection should be "Ready".
3. [Refresh the key rings](#).

On Google

1. Disable the replaced key.

After disabling the key, verify that CCKM works as expected.

2. Delete the service account key that was replaced.

Secure Copy Protocols (SCP/SFTP)

Secure Copy (SCP/SFTP) connection to the CipherTrust Manager can be configured using the following:

- [Managing Secure Copy Connections using GUI](#)
- [Managing Secure Copy Connections using ksctl](#)

Managing Secure Copy Connections using GUI

Note

The Secure Copy connections are supported in the root domain only.

To manage SCP connections using GUI, perform the following steps:

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.
4. On the Add Connection screen, select category as **SFTP/SCP**.
5. Select Select SFTP/SCP Type as **SFTP/SCP** and click **Next**.
6. Specify connection **Name** and **Description** and click **Next**.
7. Configure the below parameters.
 - **Host** - IP/hostname of the Secure Copy server.
 - **Port** - port number of the Secure Copy server. Default port is 22.
 - **Username** - username of the Secure Copy server.
 - **Protocol** - Select either **SFTP** or **SCP** protocol.
 - **Auth Method** - you can use either **Password** or **Key** for authentication purpose.

Authentication Method	Description
Password	password to authenticate the Secure Copy server.
Key	public key used for authentication. Click the Download Public key for SSH authentication button. To upload the fetched key to the list of authorized keys on the Secure Copy server, refer to the Uploading Key to the List of Authorized Keys on the Secure Copy Server section.

- **Public key of SFTP Server** - (applicable only for SFTP server) public key of the SFTP server. It is used to verify the identity of the host through key fingerprint. It is available at the `/etc/ssh` location on the SFTP server. To find the public key of the SFTP server, refer to [Finding Public Key of the Secure Copy Server](#) section.
 - **Public key of SCP Server** - (applicable only for SCP server) public key of the SCP server. It is used to verify the identity of the host through key fingerprint. It is available at the `/etc/ssh` location on the SCP server. To find the public key of the SCP server, refer to [Finding Public Key of the Secure Copy Server](#) section.
 - **Path to** - path of the Secure Copy server where backup will be transferred.
8. Click the **Test Credentials** button to check whether the connection is configured correctly. If the test is successful, the status is `OK` else the status is `Fail`.

Note

While testing the Secure Copy connection, a file with the name `temp-cm-<protocol>-test-connection` is created at the specified path on the Secure Copy server.

9. Click **Next** to move to the [Add Products](#) screen of the Add Connection wizard.

Note

The only product supported for SCP/SFTP connection is Backup/Restore.

Uploading Key to the List of Authorized Keys on the Secure Copy Server

To upload the fetched key to the list of authorized keys on the Secure Copy server, perform the following steps:

1. Open the downloaded key and copy its content without quotes ("").
2. Append the content of this public key to the following file (authorized_keys) on the Secure Copy server. This file is available at: `/home/<SCP user>/.ssh/authorized_keys`.
3. Save the file and exit.

Example

Run the below command to get the content of the `authorized_keys` file:

```
ubuntu@ip:/etc/ssh$ cat /home/ubuntu/.ssh/authorized_keys
```

Output:

```
ssh-rsa.<content of authorized_keys>...+FFChClf1in1xnKG9UL/ Yaths-UbuntuDev
```

Append the content of public key of the CipherTrust Manager:

```
ssh-rsa.<content of authorized_keys>...+FFChClf1in1xnKG9UL/ Yaths-UbuntuDevssh-rsa...<content of public key of ${cm}>...P9+9JRqDINamNougibgw==
```

In this example, the highlighted code is the public key downloaded from Step 1.

Finding Public Key of the Secure Copy Server

Following example shows how to find the public key of the SCP/SFTP server.

Example

Copy the default SSH public key (`ssh_host_ecdsa_key.pub`) of the Secure Copy server. This key is available at: `/etc/ssh/`.

Run the command:

```
ubuntu@ip:/etc/ssh$ cat ssh_host_ecdsa_key.pub
```

Output:

```
ecdsa-sha2-nistp256.....YcS6IzvTZZ6tpL/F65f/M= root@ip
```

Note

By default, the `ssh_host_ecdsa_key` (private key) is used for SSH authentication. However, you can also use other keys for SSH authentication. To do so, uncomment the other option which you want to use and comment the unused ones in the `/etc/ssh/sshd_config` file referring to the `HostKey` in the target server.

Managing Secure Copy Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete an SCP/SFTP connection
- List all SCP/SFTP connections
- Test an existing SCP/SFTP connection
- Test a New SCP/SFTP connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Name of the connection.
description	Optional	Connection description.
username	Mandatory	Username for accessing SCP/SFTP server.
products	Optional	List of products.
protocol	Optional	Use 'sftp' or 'scp'. 'sftp' is the default value.
meta	Optional	meta information in json format. This information is provided in pair. For example, --meta <code>{"color":"blue","foo":"bar"}</code> .
json-file	Optional	Connection information provided in a JSON file format. Command line parameters will take precedence over values specified in the JSON file.
auth-method	Optional	Authentication type for SCP/SFTP server. Accepted values are "key" or "password".
conn-password	Optional	Password of the connection.
host	Mandatory	Hostname or FQDN of SCP/SFTP remote machine.
path-to	Mandatory	A path where the file to be copied via SCP/SFTP. For example, "/home/ubuntu/datafolder/".
public-key	Mandatory	Public key of destination host machine. It will be used to verify the host's identity by verifying key fingerprint. You can find it in <code>/etc/ssh/</code> at host machine.

Parameter	Mandatory/Optional	Description
scp-port	Optional	Port where SCP/SFTP service runs on host (usually 22).

Creating an Secure Copy Connection

To create an SCP/SFTP connection, run:

Syntax

```
ksctl connectionmgmt scp create --name <Connection-Name> --protocol <sftp-or-scp> --host <Hostname> --scp-port <Port> --username <SCP-Connection-Username> --auth-method <Key or Password> --conn-password <SCP-Connection-Password> --path-to <Destination-Machine-Path> --public-key <key> --products <Products-Names> --meta <Key:Value>
```

Example Request

```
ksctl connectionmgmt scp create --name test --protocol sftp --host 8.8.8.8 --scp-port 22 --username admin --auth-method password --conn-password paswd --public-key key --path-to "/home/ubuntu" --products "backup/restore"
```

Example Response

```
{
  "id": "55ffad19-8c7c-4a33-8fc6-d2d2bca5deb5",
  "uri": "kylo:kylo:connectionmgmt:connections:scp-conn-55ffad19-8c7c-4a33-8fc6-d2d2bca5deb5",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2024-07-02T10:28:05.641651524Z",
  "updatedAt": "2024-07-02T10:28:05.637630097Z",
  "service": "secure-copy",
  "category": "external-server",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "test",
  "products": [
    "backup/restore"
  ],
  "meta": null,
  "host": "8.8.8.8",
  "port": 22,
  "username": "admin",
  "auth_method": "password",
  "path_to": "/home/ubuntu",
  "protocol": "sftp"
}
```

Getting Details of an Secure Copy Connection

To get details of an SCP/SFTP connection, run:

Syntax

```
ksctl connectionmgmt scp get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt scp get --id test
```

Example Response

```
{
  "id": "55ffad19-8c7c-4a33-8fc6-d2d2bca5deb5",
  "uri": "kylo:kylo:connectionmgmt:connections:scp-conn-55ffad19-8c7c-4a33-8fc6-d2d2bca5deb5",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2024-07-02T10:28:05.641651524Z",
  "updatedAt": "2024-07-02T10:28:05.637630097Z",
  "service": "secure-copy",
  "category": "external-server",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "test",
  "products": [
    "backup/restore"
  ],
  "meta": null,
  "host": "8.8.8.8",
  "port": 22,
  "username": "admin",
  "auth_method": "password",
  "path_to": "/home/ubuntu",
  "protocol": "sftp"
}
```

Updating an Secure Copy Connection

To update an SCP/SFTP connection, run:

Syntax

```
ksctl connectionmgmt scp modify --name <Connection-Name> --protocol <sftp-or-scp> --host <hostname> --scp-port <Port> --username <SCP-Connection-Username> --auth-method <Key or Password> --conn-password <SCP-Connection-Password> --public-key <key> --path-to <Destination-Machine-Path> --products <Products-Names> --meta <Key:Value>
```

Example Request

```
ksctl connectionmgmt scp modify --id test --protocol scp --host 1.2.3.4 --scp-port 32
```

Example Response

```
{
  "id": "55ffad19-8c7c-4a33-8fc6-d2d2bca5deb5",
  "uri": "kylo:kylo:connectionmgmt:connections:scp-conn-55ffad19-8c7c-4a33-8fc6-d2d2bca5deb5",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2024-07-02T10:28:05.641652Z",
  "updatedAt": "2024-07-02T11:12:05.200414626Z",
  "service": "scp",
  "category": "external-server",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "test",
  "products": [
    "backup/restore"
  ],
  "meta": null,
  "host": "1.2.3.4",
  "port": 32,
  "username": "admin",
  "auth_method": "password",
  "path_to": "/home/ubuntu",
  "protocol": "scp"
}
```

Getting List of Secure Copy Connections

To list all the SCP/SFTP connections, run:

Syntax

```
ksctl connectionmgmt scp list --protocol <sftp-or-scp>
```

Example Request 1

```
ksctl connectionmgmt scp list
```

Example Response 1

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "name": "test",
      "id": "55ffad19-8c7c-4a33-8fc6-d2d2bca5deb5",
      "uri": "kylo:kylo:connectionmgmt:connections:scp-conn-55ffad19-8c7c-4a33-8fc6-d2d2bca5deb5",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2024-07-02T10:28:05.641651524Z",
      "updatedAt": "2024-07-02T10:28:05.637630097Z",
      "service": "scp",
      "category": "external-server",
      "products": [
```

```

        "backup/restore"
      ],
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "host": "8.8.8.8",
      "port": 22,
      "username": "admin",
      "auth_method": "password",
      "public_key": "public-key",
      "path_to": "/home/ubuntu",
      "protocol": "scp"
    }
  ]
}

```

Example Request 2 (filters result based on protocol name)

```
ksctl connectionmgmt scp list --protocol sftp
```

Example Response 2

```

{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "64d21caf-ab1c-4b4a-9156-c857739e4bed",
      "uri": "kylo:kylo:connectionmgmt:connections:test-64d21caf-ab1c-4b4a-9156-c857739e4bed",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2024-07-02T10:28:05.641652Z",
      "updatedAt": "2024-07-02T11:12:05.200415Z",
      "service": "secure-copy",
      "category": "external-server",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "test-demo",
      "host": "1.2.34.5",
      "port": 22,
      "username": "test",
      "auth_method": "key",
      "path_to": "/home/ubuntu",
      "protocol": "sftp"
    }
  ]
}

```

Deleting an Secure Copy Connection

To delete an SCP/SFTP connection, run:

Syntax

```
ksctl connectionmgmt scp delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt scp delete --id test
```

Example Response

There will be no response if SCP connection is deleted successfully.

Testing an Existing Secure Copy Connection

To test an existing SCP/SFTP connection, run:

Syntax

```
ksctl connectionmgmt scp test --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt scp test --id test
```

Example Response

```
{
  "connection_ok": true
}
```

Testing a New Secure Copy Connection

To test a new SCP/SFTP connection, run:

Syntax

```
ksctl connectionmgmt scp test --host <hostname> --protocol <sftp-or-scp> --scp-port <Port> --username <SCP-Connection-Username> --auth-method <Key or Password> --conn-password <SCP-Connection-Password> --path-to <Destination-Machine-Path> --public-key <key>
```

Example Request

```
ksctl connectionmgmt scp test --host 1.2.3.4 --protocol sftp --scp-port 22 --username admin --auth-method password --conn-password paswd --public-key key --path-to "/home/ubuntu"
```

Example Response

```
{
```

```

    "connection_ok": true
  }

```

Azure

Azure connections to the CipherTrust Manager can be configured using the following:

- [Managing Azure Connections using GUI](#)
- [Managing Azure Connections using ksctl](#)

Note

If you wish to use external certificate authentication for an Azure Cloud connection, you must first [create a valid external certificate](#).

Managing Azure Connections using GUI

To manage Azure connections using GUI, perform the following steps:

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.
4. On the Add Connection screen, select category as **Cloud**.
5. Select Select Cloud Type as **Azure** and click **Next**.
6. Specify connection **Name** and **Description** and click **Next**.
7. Configure the below parameters.
 - **Client ID** - this is an Application ID of the Azure application. It can be used either with Client Secret or Certificate to authenticate the application.
 - **Tenant ID** - this is the Office365 tenant ID. It is a globally unique identifier (GUID). For more details, refer to the Azure documentation.
 - **Cloud Name** - the name of the Azure cloud to connect to. Currently, only the following options are available:
 - Azure Cloud - For Azure Cloud configuration, refer to [Creating an Azure Cloud Connection](#).
 - Azure China Cloud

- Azure US Government
- Azure Stack - For Azure Stack configuration, refer to [Configure Azure Stack](#).
- **Authentication** - you can use either **Client Secret** or **Certificate** for authentication purpose.
 - **Client Secret** – this authentication method uses the application password of the Client ID to enable communication between Azure and CipherTrust Manager.
 - **Certificate** - this authentication method is used to enable password-less communication between Azure and CipherTrust Manager.

Note

Azure Stack does not support Certificate authentication.

- Select the **Certificate** radio button
- Select **Application** or **External** as the Certificate Type.

The **Application** certificate type is generated by CipherTrust Manager and self-signed.

The **External** certificate type is a pre-existing certificate generated on CipherTrust Manager and then signed by a CA local or external to the CipherTrust Manager. To use this option, you first need to [create a valid external certificate](#).

Configure Application Certificate

- Click the **Generate and Download** button.
- Upload the downloaded certificate on Salesforce for the provided Client ID.
- Once the upload is done, verify the **Thumbprint** on the CipherTrust Manager and Azure. Both the thumbprints must match.
- Specify **Certificate Duration in Days**, if desired. The default certificate duration is 730 days (2 years).

Configure External Certificate

Ensure that you have fulfilled the prerequisites to [create a valid external certificate](#).

Do one of the following:

- Select **File Upload** and click the **Upload Certificate** to upload the external certificate as a file.
- Select **Text** and paste the certificate contents in the text box.

Note

The CipherTrust Manager allows you to modify the external certificate in the existing connection. Any unused certificate will be automatically deleted after 24 hours.

8. Click the **Test Credentials** button to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.
9. Click **Next** to move to the **Add Products** screen of the Add Connection wizard.

- **Configure Azure Stack**

Note

- This configuration is applicable to Azure Stack only.
- Configuring an Azure Stack connection requires various URLs, described below. To get these URLs, run the command `Get AzureRmEnvironment` in your Azure AD VM. Refer to [Connect with Azure AD](#) for details.

- **Azure Stack Connection Type** - Azure stack supports two types backed by Active Directory as an identity provider:
 - AAD - Azure Active Directory
 - ADFS - Active Directory Federation Services
- **Active Directory Endpoint** - this is a URL at which the identity providers can be reached. For example, <https://login.microsoftonline.com/>
- **Key Vault DNS Suffix** - this is a DNS suffix for the key vault in the Azure Stack. For example, `vault.local.azurestack.external`.
- **Management URL** - this is the URL with a unique identifier for Azure Resource Manager registered with your identity provider.
- **Resource Manager URL** - this URL is the location of the Azure Resource Manager service. For example, `https://management.azure.com` or `https://management.local.azurestack.external`
- **Vault Resource URL** - this is the URL to access vault resources. For example, `https://vault.local.azurestack.external`
- **Azure Server Certificate** - this is the Server certificate used by HTTPS protocol for a secure

connection.

Managing Azure Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete an Azure Stack connection
- List all Azure Stack connections
- Test an existing Azure Stack connection
- Test parameters for a Azure Stack connection
- Create an Azure Cloud Connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Unique name of the connection.
description	Optional	Connection description.
products	Optional	List of products.
clientid	Mandatory	Unique Identifier (client ID) for the Azure application.
meta	Optional	meta information in json format. This information is provided in pair. For example, --meta <code>"{\"color\":\"blue\",\"foo\":\"bar\"}"</code> .
tenantid	Mandatory	Tenant ID of the Azure application.
cloudname	Optional	Name of the Azure cloud to connect to.
connection-type	Optional	Azure stack connection type (AAD or ADFS).
active-dir-endpoint	Optional	Azure stack active directory authority URL.
management-url	Optional	Azure stack management URL.
res-manager-url	Optional	Azure stack resource manager URL.
key-vault-dns-suffix	Optional	Azure stack key vault dns suffix.
vault-res-url	Optional	Azure stack vault service resource URL.
server-cert-file	Optional	Server certificate file path.

Note

Examples in this section are for ADFS connection type. Similarly, you can manage connections for AAD by changing the `connection-type` to AAD.

Creating an Azure Stack Connection

To create an Azure Stack connection, run:

Syntax

```
ksctl connectionmgmt azure create --name <Connection-Name> --products <Product-Names> --clientid <Azure-Key-ID> --meta <Key-Values> --tenantid <Tenant-ID> --cloudname <Cloud-Name> --connection-type <Connection-Type> --active-dir-endpoint <Active-Directory-Endpoint> --management-url <Management-URL> --res-manager-url <Resource-Manager-URL> --key-vault-dns-suffix <Keyvault-DNS-Suffix> --vault-res-url <Vault-Resource-URL> --server-cert-file <Server-Certificate-File>
```

Example Request

```
ksctl connectionmgmt azure create --name test-azs-adfs --products cckm --clientid client123 --secret secret123 --tenantid 123 --cloudname AzureStack --connection-type ADFS --active-dir-endpoint "https://adfs.local.azurestack.external/adfs" --management-url "https://management.adfs.azurestack.local/2aeeb93d-50a7-415e-8b217-01b5c5e2fasd" --res-manager-url "https://management.local.azurestack.external/" --key-vault-dns-suffix "vault.local.azurestack.external" --vault-res-url "https://vault.local.azurestack.external" --server-cert-file ~/server.pem
```

Example Response

```
{
  "id": "2cc2d7db-155c-472f-b248-4ca4072d1bb3",
  "uri": "kylo:kylo:connectionmgmt:connections:test-azs-adfs-2cc2d7db-155c-472f-b248-4ca4072d1bb3",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-24T11:06:31.917450971Z",
  "updatedAt": "2020-12-24T11:06:31.916445598Z",
  "service": "azure",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "test-azs-adfs",
  "products": [
    "cckm"
  ],
  "tenant_id": "123",
  "client_id": "client123",
  "cloud_name": "AzureStack",
  "active_directory_endpoint": "https://adfs.local.azurestack.external/adfs",
  "vault_resource_url": "https://vault.local.azurestack.external",
```

```

    "resource_manager_url": "https://management.local.azurestack.external/",
    "key_vault_dns_suffix": "vault.local.azurestack.external",
    "management_url": "https://management.adfs.azurestack.local/2aeeb93d-50a7-415e-8b217-01b5c5e2fasd",
    "azure_stack_server_cert": "-----BEGIN CERTIFICATE-----\nMIIEPDCCAiSgAwIBAgIRALJpeHdhAFCGctcAVJ1fpwMwDQYJKoZIhvcNAQELBQAw\nnWjELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAk1EMRAwDgYDVQQHEwdCZWxjYW1wMRAw\nnDgYDVQQKEwdHZW1hbHRvMyZp/w...+R5OmtS0p2wsRofbmY9in\n\nnoE4di6Pk83BMh2RpCDxDPb0UqTGlRlbPuew0mNfI2ePQLoFhyoTmwN1xEgUpexlu\n\nnQb9IovyN2/Bm1QNpt4wRwoDF4sGAgcEM6AAtMVe2uVQ=\n\n-----END CERTIFICATE-----\n",
    "azure_stack_connection_type": "ADFS"
}

```

Getting Details of an Azure Stack Connection

To get details of an Azure Stack connection, run:

Syntax

```
ksctl connectionmgmt azure get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt azure get --id 2cc2d7db-155c-472f-b248-4ca4072d1bb3
```

Example Response

```

{
  "id": "2cc2d7db-155c-472f-b248-4ca4072d1bb3",
  "uri": "kylo:kylo:connectionmgmt:connections:test-azs-adfs-2cc2d7db-155c-472f-b248-4ca4072d1bb3",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-24T11:06:31.917451Z",
  "updatedAt": "2020-12-24T11:06:31.916446Z",
  "service": "azure",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "test-azs-adfs",
  "products": [
    "cckm"
  ],
  "tenant_id": "123",
  "client_id": "client123",
  "cloud_name": "AzureStack",
  "active_directory_endpoint": "https://adfs.local.azurestack.external/adfs",
  "vault_resource_url": "https://vault.local.azurestack.external",
  "resource_manager_url": "https://management.local.azurestack.external/",
  "key_vault_dns_suffix": "vault.local.azurestack.external",
  "management_url": "https://management.adfs.azurestack.local/2aeeb93d-50a7-415e-8b217-01b5c5e2fasd",
  "azure_stack_server_cert": "-----BEGIN CERTIFICATE-----\nMIIEPDCCAiSgAwIBAgIRALJpeHdhAFCGctcAVJ1fpwMwDQYJKoZIhvcNAQELBQAw\nnWjELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAk1EMRAwDgYDVQQHEwdCZWxjYW1wMRAw\nnDgYDVQQKEwdHZW1hbHRvMRowGAYDVQQDExFkZX1lTZW1cmUgUm9vdCBDQTAeFw0y\n\nMDEyMDIwOTIzMTRaFw0yMDIwOTIzMTRaMCIXDjAMBgNVBAMTBWVkbWlucmRA

```

```
w...YHsN\nobEToCx8UNXoZlYUX2f8hE9ad/tGrpwqXUHkSWjnET2+R5OmtS0p2wsRofbmY9in\noE4di
6Pk83BMh2RpCDxDPb0UqTG1R1bPuew0mNfI2ePQLoFhyoTmwN1xEgUpex1u\nQb9IovyN2/Bm1QNpt4wR
woDF4sGAgcEM6AAAtMVe2uVQ=\n-----END CERTIFICATE-----\n",
  "azure_stack_connection_type": "ADFS"
}
```

Updating an Azure Stack Connection

To update an Azure Stack connection, run:

Syntax

```
ksctl connectionmgmt azure modify --id <Connection-Name/ID> --products <Product-N
ames> --secret <Azure-Client-Secret> --meta <Key-Values>
```

Example Request

```
ksctl connectionmgmt azure modify --id 2cc2d7db-155c-472f-b248-4ca4072d1bb3 --ten
antid 456
```

Example Response

```
{
  "id": "2cc2d7db-155c-472f-b248-4ca4072d1bb3",
  "uri": "kylo:kylo:connectionmgmt:connections:test-azs-adfs-2cc2d7db-155
c-472f-b248-4ca4072d1bb3",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-24T11:06:31.917451Z",
  "updatedAt": "2020-12-24T11:14:12.702605505Z",
  "service": "azure",
  "category": "cloud",
  "last_connection_ok": false,
  "last_connection_error": "Post \"https://adfs.local.azurestack.external/a
dfs/oauth2/token\": dial tcp: lookup adfs.local.azurestack.external on 127.0.0.1
1:53: no such host",
  "last_connection_at": "2020-12-24T11:12:48.403146Z",
  "name": "test-azs-adfs",
  "products": [
    "cckm"
  ],
  "meta": "",
  "tenant_id": "456",
  "client_id": "client123",
  "cloud_name": "AzureStack",
  "active_directory_endpoint": "https://adfs.local.azurestack.external/adf
s",
  "vault_resource_url": "https://vault.local.azurestack.external",
  "resource_manager_url": "https://management.local.azurestack.external/",
  "key_vault_dns_suffix": "vault.local.azurestack.external",
  "management_url": "https://management.adfs.azurestack.local/2aeeb93d-50a
7-415e-8b217-01b5c5e2fasd",
  "azure_stack_server_cert": "-----BEGIN CERTIFICATE-----\nMIIePDCCAiSgAwIB
AgIRALJpeHdhAFCGctcAVJ1fpwMwDQYJKoZIhvcNAQELBQAw\nWjEELMAkGA1UEBhMCVVMxMzA5BjBjNVBAg
TAk1EMRAwDgYDVQQHEwdCZWxjYW1wMRwA\nNDgYDVQQKEwdHZW1hbHRvMR0wGAYDVQQDExFLZX1TZ...k8
```

```
3BMh2RpCDxDPb0UqTGlRlbPuew0mNfI2ePQLoFhyoTmwN1xEgUpex1u\nQb9IovyN2/Bm1QNpt4wRwoDF
4sGAgcEM6AAAtMVe2uVQ=\n-----END CERTIFICATE-----\n",
    "azure_stack_connection_type": "ADFS"
}
```

Deleting an Azure Stack Connection

To delete an Azure Stack connection, run:

Syntax

```
ksctl connectionmgmt azure delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt azure delete --id 2cc2d7db-155c-472f-b248-4ca4072d1bb3
```

There will be no response if Azure Stack connection is deleted successfully.

Getting List of Azure Stack Connections

To list all the Azure Stack connections, run:

Syntax

```
ksctl connectionmgmt azure list
```

Example Request

```
ksctl connectionmgmt azure list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "2cc2d7db-155c-472f-b248-4ca4072d1bb3",
      "uri": "kylo:kylo:connectionmgmt:connections:test-azs-adfs-2c
c2d7db-155c-472f-b248-4ca4072d1bb3",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2020-12-24T11:06:31.917451Z",
      "updatedAt": "2020-12-24T11:06:31.916446Z",
      "service": "azure",
      "category": "cloud",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "test-azs-adfs",
    }
  ]
}
```

```

    "products": [
      "cckm"
    ],
    "tenant_id": "123",
    "client_id": "client123",
    "cloud_name": "AzureStack",
    "active_directory_endpoint": "https://adfs.local.azurestack.e
external/adfs",
    "vault_resource_url": "https://vault.local.azurestack.externa
1",
    "resource_manager_url": "https://management.local.azurestac
k.external/",
    "key_vault_dns_suffix": "vault.local.azurestack.external",
    "management_url": "https://management.adfs.azurestack.local/2
aeeb93d-50a7-415e-8b217-01b5c5e2fasd",
    "azure_stack_server_cert": "-----BEGIN CERTIFICATE-----\nMIIE
PDCCAiSgAwIBAgIRALJpeHdhAFCGctcAVJl1fpwMwDQYJKoZIhvcNAQELBQAw\nwWJELMAkGA1UEBhMCMVVM
xCZAJBgNVBAGTAK1EMRAwDgYDVQQHEwdCZWxjYW1wMRAw\nnDgYDVQQKEwdHZW1hbHRvMRowGAYDVQQDEx
FLZX1TZWN1cmUgUm9vdCBDQTAeFw0y\nnMDEyMDIwOTIzMTRaFw0yMjEyMDIwOTIzMTRaMCIxMjE2PQLoFh
AMTBWFkbWluMRAw\nnDgYKCCZImizPzPyLQGBARMAMIIBIjANBgkqhki...b0UqTGlRlbpuew0mNfiI2ePQLoFh
yoTmwN1xEgUpexlu\nnQb9IovyN2/Bm1QNpt4wRwoDF4sGAGcEM6AAAtMve2uVQ=\n-----END CERTIFIC
ATE-----\n",
    "azure_stack_connection_type": "ADFS"
  },
}

```

Testing an Existing Azure Stack Connection

To test an existing Azure Stack connection, run:

Syntax

```
ksctl connectionmgmt azure test --id <Connection-Name/ID> --clientid <Azure-Key-ID> --secret <Azure-Client-Secret> --tenantid <Tenant-ID>
```

Example Request

Example Response

```
{
  "connection_ok": true
}
```

Testing Parameters for an Azure Stack Connection

To test parameters for an Azure Stack connection, run:

Syntax


```
ksctl connectionmgmt azure test --clientid <Azure-Key-ID> --meta <Key-Values> --tenantid <Tenant-ID> --cloudname <Cloud-Name> --connection-type <Connection-Type> --active-dir-endpoint <Active-Directory-Endpoint> --management-url <Management-URL> --res-manager-url <Resource-Manager-URL> --key-vault-dns-suffix <Keyvault-DNS-Suffix> --vault-res-url <Vault-Resource-URL> --server-cert-file <Server-Certificate-File>
```

Example Request

```
ksctl connectionmgmt azure test --clientid client123 --secret secret123 --tenantid 123 --cloudname AzureStack --connection-type ADFS --active-dir-endpoint "https://adfs.local.azurestack.external/adfs" --management-url "https://management.adfs.azurestack.local/2aeeb93d-50a7-415e-8b217-01b5c5e2fasd" --res-manager-url "https://management.local.azurestack.external/" --key-vault-dns-suffix "vault.local.azurestack.external" --vault-res-url "https://vault.local.azurestack.external" --server-cert-file ~/server.pem
```

Example Response

```
{
  "connection_ok": true
}
```

Creating an Azure Cloud Connection

The Azure Cloud connection can be created using:

- Internal certificate
- External certificate

Creating an Azure Cloud Connection using internal certificate

To create an Azure Cloud connection using internally generated self-signed certificate, run:

Example

```
ksctl connectionmgmt azure create --name "azureconnection2" --clientid "a-client-id" --cloudname "AzureCloud" --use-certificate true
```

Response

```
{
  "id": "525d00e7-e677-4411-9f8c-0af01576d4c5",
  "uri": "kylo:kylo:connectionmgmt:connections:azureconnection2-525d00e7-e677-4411-9f8c-0af01576d4c5",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-08-23T08:28:00.109946977Z",
  "updatedAt": "2022-08-23T08:28:00.108830988Z",
  "service": "azure",
  "category": "cloud",
}
```

```

    "last_connection_ok": null,
    "last_connection_at": "0001-01-01T00:00:00Z",
    "name": "azureconnection2",
    "client_id": "a-client-id",
    "cloud_name": "AzureCloud",
    "certificate": "-----BEGIN CERTIFICATE-----\nMIIFvjCCA6agAwIBAgIRAIEusgD8lFVB
JoLiSXw7EBUwDQYJKoZIhvcNAQELBQAww\nfzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbgG1mb3JuaWE
xETAPBgNVBACTCFNh\nnbiBkb3NlMQ8wDQYDVQQKEwZUaGFsZXNMcXFDASBgNVBASTC0NpcGhlclRydXN0MS
Ew\nnHwYDVQQDEExhY2ttLnRoYWxlc2VzZW50eS5jb20wHhcNMjIwODIzMDgyODAw\nnWhcNMjIwODI
yMDgyODAwWjB/MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZv\nncm5pYTERMA8GA1UEBxMIU2FuIE
pvc2UxDzANBgNVBAoTB1RoY...MFeV/+CJH53cqBTBJ\nnJTv7aYJtV6vzW9tVaCb6nPnLtryc49ucjNeN
PFfCxtXrZog7fJeocFsdWimMwLXy\nnSqGYAaFdOJJZgAhv1Qusl4oJIVZ3Cw9OAK61whTjEMfXAYJHRuW
P/0uDZWN06/z7\nn8GmgLrPeEBuc8qyXy26ALoUm2rsDCSjo14qL1u29bVkeFP3ZdPBoapvyzCudmSx2\nn
2NuDQJO/TaREDGkvx27xyu8thIPRLCb4HuzlDhDi3Xg2tA==\nn-----END CERTIFICATE-----\nn",
    "certificate_thumbprint": "5BB5FC44C0CAFA417773CA4EC80A07232AC02499"
}

```

Creating an Azure Cloud Connection using external certificate

Note

The external certificate cannot be used with `use_certificate` and `client_secret` parameters.

To create an Azure cloud connection using an external certificate generated from the custom CSR signed by any internal/external CA:

1. Generate a new Certificate Signing Request (CSR). The Azure connections do not support RSA 1024-bit keys for creating CSRs. The supported RSA key strengths are 2048 and 4096 bits.

Syntax

```
ksctl connectionmgmt connections csr --cn <common-name> --csr-outfile <filename>
```

Example

```
ksctl connectionmgmt connections csr --cn "test" --csr-outfile "Azurecsr.pem"
```

Response

```

{
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIHIMHECAQAwDzENMA8GA1UEAxMEd
GVzdDBZMBMBGByqGSM49AgEGCCqGSM49AwEH\nnA0IABPkDWFDb/khM9xaRPAAnRKJ0nq7hfkQiX9UY8v
03zL/X9YybSB/L3W4CpI0o6\nnhLZQtoOjiv6ziRToKDFpq4K/WdegADAKBggqhkJOPQQDAgNHADBEA
iA2kC7YOUuq\nn0BtS+SDI/OuCd21JhkQoVX0ZcD/e/g5jtQIgTHE79SCJ/G/UXLNHjfmGZyP9zVmH\

```

```
nObA8stMQDpSMJhM=\n-----END CERTIFICATE REQUEST-----\n"
}
```

This CSR can only be used for one connection in the native domain. Also, this CSR can't be reused in other domains.

2. Sign the CSR with any local or external CA. It will generate an external certificate.
3. Upload the generated certificate to the Azure portal.
4. Create the Azure cloud connection using the external certificate generated above.

Example

```
ksctl connectionmgmt azure create --name "azureconnecnwithcert" --json-file ce
rtazure.json
```

Response

```
{
  "id": "5c440f1f-650c-497e-bd38-b7ebfe7e4e65",
  "uri": "kylo:kylo:connectionmgmt:connections:azure-connectio2n-5c440f1f-65
0c-497e-bd38-b7ebfe7e4e65",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-08-23T08:16:24.236837416Z",
  "updatedAt": "2022-08-23T08:16:24.23580786Z",
  "service": "azure",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "azure-connection",
  "products": [
    "cckm"
  ],
  "meta": {
    "color": "blue"
  },
  "tenant_id": "3bf0dbe6-a2c7-431d-9a6f-4843b74c71285nfjdu2",
  "client_id": "3bf0dbe6-a2c7-431d-9a6f-4843b74c7e12",
  "cloud_name": "AzureCloud",
  "certificate": "-----BEGIN CERTIFICATE-----\nMIIFUzCCAugAwIBAgIRAIzHRMIS7
tVGXVzIXlhGwCMwDQYJKoZIhvcNAQELBQAw\nWjELMAkGA1UEBhMCVVMxCzAJBgNVBAGTA1RYMQ8wD
QYDVQQHEwZBdXN0aW4xDzAN\nBgNVBAoTB1RoYXNlczEcMBoGA1UEAxMTQ21waGVyVHJlc3QgUm9vd
CBDQTAeFw0y\nMjA4MjIwODE0MzNaFw0yMzA4MjMwODE0MzNaMGQxCzAJBgNVBAYTA1VTMQswCQYD\
```

```
nVQQIEwJNRDEQMA4GA1UEBxMHQmVsY2FtcDEVMBMGA1UEChMMVGhhbGVzIECA...0tYrQf8Jtk9x
W+TqQfli1QZSfpK7vBypys87hFYRD7I82EA6zDLtIz16rjcFPUG\nitTI7OJsCVX8QhaLGqc3vahhE
sEfKhEEOcUwEc9oGAFOlsjrJvVM6/wwebvD0G3\nM+tG8aEYPLphmR4dD5Zp9mmlcVdpUk
M=\n-----END CERTIFICATE-----\n",
    "certificate_thumbprint": "9CECEBF8E89C12E201461200070376971B9678374"
}
```

JSON File

```
{
  "name": "azure-connection",
  "products": [
    "cckm"
  ],
  "meta": {
    "color": "blue"
  },

  "cloud_name": "AzureCloud",
  "client_id": "3bf0dbe6-a2c7-431d-9a6f-4843b74c7e12",
  "tenant_id": "3bf0dbe6-a2c7-431d-9a6f-4843b74c71285nfjdu2",
  "certificate": "-----BEGIN CERTIFICATE-----\nMIIFUzCCAzugAwIBAgIRAIzHRMI
S7tVGXVzIXlhGwCMwDQYJKoZIhvcNAQELBQAw\nWjELMAkGA1UEBhMCVVMxMzA4MjMwODE0MzNn..Qf8Jtk9xW+TqQfli1QZSfpK7v
wDQYDVQQHEwZBdXN0aW4xDzAN\nBgNVBAoTB1RoYWxlcEcMB0GA1UEAxMTQ21waGVyVHJ1c3QgUm9
vdCBDQTAeFw0y\nmMjA4MjIwODE0MzNaFw0yMzA4MjMwODE0MzNn..Qf8Jtk9xW+TqQfli1QZSfpK7v
Bypys87hFYRD7I82EA6zDLtIz16rjcFPUG\nitTI7OJsCVX8QhaLGqc3vahhEsEfKhEEOcUwEc9oG
AFOLsjrJvVM6/wwebvD0G3\nM+tG8aEYPLphmR4dD5Zp9mmlcVdpUkM=\n-----END CERTIFICAT
E-----\n"
}
```

The CipherTrust Manager allows you to modify the external certificate in the existing connection. Any unused certificate will be automatically deleted after 24 hours.

Hadoop Knox

Hadoop Knox connections to the CipherTrust Manager can be configured using the following:

- [Managing Hadoop Knox Connections using GUI](#)
- [Managing Hadoop Knox Connections using ksctl](#)

Managing Hadoop Knox Connections using GUI

Note

Only one TDP connection of a particular type (Hadoop Knox) can be created at a time on the CipherTrust Manager. The Hadoop Knox connections are supported in the root domain only.

Note

For Hadoop Knox, only HTTPS protocol is supported.

Note

While testing a Hadoop Knox connection, if the connectivity is working for any one of the nodes in the connection, the overall connection status will be true.

To manage Hadoop Knox connections using GUI, perform the following steps:

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.
4. On the Add Connection screen, select category as **TDP**.
5. Select TDP Type Type as **Hadoop Knox** and click **Next**.
6. Specify connection **Name** and **Description** and click **Next**.
7. Configure the below parameters.

Add Knox Node

- **Node Host** - provide hostname of a Hadoop Knox node. Use the hostname value as returned by the 'hostname' command on the node that is running Knox, not the Fully Qualified Domain Name (FQDN). For example, if the FQDN is "my-tdp-node1-269.sjinternal.com" and the 'hostname' command returns "my-tdp-node1-269", use "my-tdp-node1-269".
- **Port** - provide port number of the Hadoop Knox node.
- **Certificate** - upload the Hadoop Knox node certificate.

To add multiple nodes in a Hadoop Knox connection, click **+Add Knox Node**.

- **Topology** – provide the Knox topology. The default topology name is "default". If you are not using the default topology, name your own topology.
 - **Authentication** – provide a valid credential provisioned in the authentication service configured on Knox through in Ambari.
8. Click the **Test Credentials** button to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.
 9. Click **Next** to move to the **Add Products** screen of the Add Connection wizard.

Managing Hadoop Knox Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete a Hadoop Knox connection
- List all Hadoop Knox connections
- Test an existing Hadoop Knox connection
- Test parameters for a Hadoop Knox connection
- Add/Get/Update/Delete a node in an existing Hadoop Knox connection
- List all nodes in a Hadoop Knox connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Unique connection name.
description	Optional	Connection description.
nodes-json-file	Mandatory	List of server nodes in a JSON file (For example, 'ServerNodes.json' in case of Hadoop).
hostname	Mandatory	hostname for Hadoop Server.
port	Mandatory	Port for Hadoop Server. Possible values are 1-65535.
protocol	Mandatory	HTTP or HTTPS protocol to be used for communication with the Hadoop node (HTTPS required for hadoop-knox).
path	Optional	path for Hadoop Server.
server-certificate	Optional	SSL certificate for Hadoop Server

Parameter	Mandatory/Optional	Description
		TLS communication.
username	Mandatory	Username for accessing Hadoop server (required for Knox).
conn-password	Mandatory	Password for Hadoop server (required for Knox).
meta	Optional	Meta information in the json format. For example, --meta <code>{"color\":"blue\","fo o\":"bar\}"</code> .
service	Mandatory	Name of the third-party service associated with the resource. For example,hadoop-knox.
topology	Optional	Topology deployment of the Knox gateway.

Creating a Hadoop Knox Connection

To create a Hadoop Knox connection, run:

Syntax

```
ksctl connectionmgmt hadoop create --name <Connection-Name> --products <Products-Names> --username <Server-UserName> --conn-password <Server-Password> --meta <Key-Values> --service <Service-Name> --topology <Topology> --nodes-json-file <JSON-File>
```

Example Request

```
ksctl connectionmgmt hadoop create --name knox --service hadoop-knox --conn-password admin --username admin --nodes-json-file ../hadoop.json --topology default
```

Example Response

```
{
  "id": "d9fe26ca-0321-4cba-bc14-8334bfc6f9c",
  "uri": "kylo:kylo:connectionmgmt:connections:knox-d9fe26ca-0321-4cba-bc14-8334bfc6f9c",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-28T10:56:00.729395192Z",
  "updatedAt": "2020-12-28T10:56:00.728472048Z",
  "service": "hadoop-knox",
  "category": "system",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "knox",
```

To get details of a Hadoop Knox connection, run:

```
ksctl connectionmgmt hadoop get --id <Connection-Name/ID>
```

```
ksctl connectionmgmt hadoop get --id 604da7f5-0359-4b56-ae8d-7164857a2f2f
```



```
]
}
```

Updating a Hadoop Knox Connection

To update a Hadoop Knox connection, run:

Syntax

```
ksctl connectionmgmt hadoop modify --id <Connection-Name/ID> --products <Product
s-Names> --username <Server-Username> --conn-password <Server-Password> --meta <K
ey:Values> --topology <Topology>
```

Example Request

```
ksctl connectionmgmt hadoop modify --id 604da7f5-0359-4b56-ae8d-7164857a2f2f --us
ername admin
```

Example Response

```
{
  "id": "604da7f5-0359-4b56-ae8d-7164857a2f2f",
  "uri": "kylo:kylo:connectionmgmt:connections:hadoopconnection-604da7f5-0359-4
b56-ae8d-7164857a2f2f",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-28T04:48:36.061995Z",
  "updatedAt": "2020-12-28T09:34:20.197779489Z",
  "service": "hadoop-knox",
  "category": "system",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "hadoopConnection",
  "products": [
    "cte"
  ],
  "meta": "",
  "description": "test description",
  "username": "admin",
  "topology": "default"
}
```

Deleting a Hadoop Knox Connection

To delete a Hadoop Knox connection, run:

Syntax

```
ksctl connectionmgmt hadoop delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt hadoop delete --id 604da7f5-0359-4b56-ae8d-7164857a2f2f
```

There will be no response if connection is deleted successfully.

Getting List of Hadoop Knox Connections

To list all the Hadoop Knox connections, run:

Syntax

```
ksctl connectionmgmt hadoop list
```

Example Request

```
ksctl connectionmgmt hadoop list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "604da7f5-0359-4b56-ae8d-7164857a2f2f",
      "uri": "kylo:kylo:connectionmgmt:connections:hadoopconnection-604da7f5-0359-4b56-ae8d-7164857a2f2f",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2020-12-28T04:48:36.061995Z",
      "updatedAt": "2020-12-28T04:48:36.061306Z",
      "service": "hadoop-knox",
      "category": "system",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "hadoopConnection",
      "products": [
        "cte"
      ],
      "description": "test description",
      "username": "admin",
      "topology": "default",
      "nodes": [
        {
          "hostname": "host1",
```

```

        "port": "1234",
        "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIG1TCCB
b2gAwIBAgIQBVfICygm6F7ChFEkylreTANBgkqhkiG9w0BAQsFADBw\nMQswCQYDVQQGEwJVUzEVMBMG
A1UEChMRRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3\nd3cuZGlnaWNlcnQuY29tMS8wLQYDVQQDEyZEaWd
pQ2VydCBTSEEyIEhpZ2ggQXNz\ndXJhbmNlIFNlcnZlcjBQTAeFw0yMDA1MDUwMDAwMDBaFw0yMjA1MT
AxMjAwMDBa\nMGYxCzAJBgNVBAYTA1VTMRMwEQYDVQQIEWpDYWxpZm9ybm1hMRYwFAYDVQQHEw1T\nnYW4
gRnJhbmNpc2NvMURUeWYDVQQKEWxHaXRIdWIsIEluYy4xEzARBgNVBAMTCmdp\ndGh1Yi5jb20wgGEiMA
0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC7MrTQ2J6a\nnoX5KUwrgO9cQ9STO5R4/zBUxxvI5S8bm
c0QjWfIVAwHWuT0Bn/H1oS0LM0tTkQm\nnArqN77v9McVB8MWTGsmGQnS/1kQRFuKiYGUHf7iX5pfijbY
sOkfb4AiVKysKUNV\nnUtgvVpJoe5RWURjQp9XDWkeo2DzGHXLcBDadrM8VLC6H1/D9SXdVruxKqduLKR4
1\nnZ/6dlSDdeY1gCnhz3ChlpYbfmfsTCTamw+AtRtlwK3b2rfTHffhowjuzM15UKt+b\nnrr/cEB1AjQTv
a8rutYU9K9ONgl+pG2u7Bv516DwmNy8xz9wOjTeOpeh0M9N/ewq8\ncgbr87LFaxilAgMBAAGjggNzMI
DbzAfBgNVHSMEGDAWgBRRaP+QrwIHdTz2M2WVWk\nnYqISuFlyOzAdBgNVHQ4EFgQUYwLSXQJf943VWhKedh
E2loYsikgwJQYDVR0RB4w\nnHIIKZ210aHViLmNvbYI0d3d3LmdpdGh1Yi5jb20wDgYDVR0PAAQH/BAQDA
gWgMB0G\nnA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjB1BgNVHR8EbjBsMDSGMAwhi5o\ndHRwOi
8vY3J5SMy5kaWdpY2VydC5jb20vc2hhMi1oYS1zZXJ2ZXItZzYuY3J5SMDSg\nnMqAwhi5odHRwOi8vY3J5S
N5kaWdpY2VydC5jb20vc2hhMi1oYS1zZXJ2ZXItZzYuY3J5SMEwGA1UdIARFMEMwNwYJYIZIAy
b9bAEB
MCowKAYIKwYBBQUHAgEWHGh0dHBz\nnOi8vd3d3LmR5Z21jZXJ0LmNvbS9DUFMwCAYGZ4EMAQICMIGDBgg
rBgEFBQcBAQR3\nnMHUwJAYIKwYBBQUHMAAGGGGh0dHA6Ly9vY3NwLmR5Z21jZXJ0LmNvbTBnBggrBgEF\nn
BQcwAoZBAHR0cDovL2NhY2VydHMuZGlnaWNlcnQuY29tL0R5Z21jZXJ0U0hBMkhp\nnZ2hBc3N1cmFuY2V
TZXJ2ZXJ2ZDQ5jcnQwDAYDVR0TAQH/BAIwADCCAXwGCisGAQQB\nn1nkCBAIEggFsBIIbAFAfMAHUAKXm+8J
45OSHwVnOfY6V35b5XfZxgCvj5TV0mXCVd\nnx4QAAAFx51tprwAABAMARjBEAiAuWGCWxN/M0Ms3KOsqF
jDMHT8Aq0SlHfQ68KDg\nnrVU6AAIgDA+2EB0D5W5r0i4Nh1jx6AB1IByzrEdfcxiOD/o6//EAdQAiRUUH
WVUk\nnVpY/oS/x922G4CMmY63AS39dxoNcbuIPAgAAAXHmW2nTAAAEAwBGMEQCIBp+XQKa\nnUDiPHwjBx
dv5qvgYALKaysKqMF60gqem8iPRAiAk9Dp5+VBUXfShqyW+tVShUigh\nnndopccf8Gs21KJ4jXgB2AFGj
sPX9AXmcVm24N3iPDKR6zBsny/eeiEKaDf7UiwXl\nnAAABceZbabsAAAQDAEcwRQIgd/5HcxT4wfNV8za
vwXjYkw2TYBAuRccqplSjWKF\nn4EoCIQDHSTHxnbpxWFbP6v5Y6nGFZCDjaHgd9HrzUv2J/DaacDANBg
kqhkiG9w0B\nnAQsFAAOCAQEAhjKPNB4r+jR3gg6RA5xICTW/A5YMcyqtK0clQzFr8S7/l+skGpC\nnyCH
rJfFrLDeYKqgabvLRT6YvvM862MGfMMDsk+sKWtzLbDIcYG7sbviGpU+gtG1q\nnB0ohWNApfWWKyNpquq
vwdSEzAEBvhcUT5idzbK7q45bQU9vBIWgQz+PYULAU7KmY\nnz7jOYV09o22TNMQT+hFmo92+EBlwSeIET
YEsHy5ZxixTRTvu9hP00CyEbiht5OTK\nn5EiJG6vsIh/uEtRsdnMCxV06W2f20Af4iSfo0uk6clryHef
h08FcwA4pSNUaPyi\nnPb8YGQ6o/blejFzo/OSiUnDueafSJ0p6SQ==\nn-----END CERTIFICATE-----
",
        "path": "/gateway/default/webhdfs/v1",
        "protocol": "https"
    }
}
]
}
]
}

```

Testing an Existing Hadoop Knox Connection

To test an existing Hadoop Knox connection, run:

Syntax

```
ksctl connectionmgmt hadoop test --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt hadoop test --id 604da7f5-0359-4b56-ae8d-7164857a2f2ft;
```

Example Response

```
{
  "status": true,
  "nodes": [
    {
      "hostname": "1.2.3.4",
      "connection_ok": true,
    }
  ]
}
```

Testing Parameters for a Hadoop Knox Connection

To test parameters for a Hadoop Knox connection, run:

Syntax

```
ksctl connectionmgmt hadoop test --username <Server-Username> --conn-password <Server-Password> --service <Service-Name> --nodes-json-file <JSON-File>
```

Example Request

```
ksctl connectionmgmt hadoop test --username admin --conn-password test_password --nodes-json-file hadoop.json --service hadoop-knox
```

Example Response

```
{
  "status": true,
  "nodes": [
    {
      "hostname": "1.2.3.4",
      "connection_ok": true,
    }
  ]
}
```

Adding a Node in an Existing Hadoop Knox Connection

To add a node in an existing Hadoop Knox connection, run:

Syntax

```
ksctl connectionmgmt hadoop node add --id <Connection-Name/ID> --hostname <Server-Hostname> --port <Server-Port> --path <Server-Path> --server-cert-file <Server-
```

```
Certificate-File> --protocol <Communication-Protocol>
```

Example Request

```
ksctl connectionmgmt hadoop node add --hostname node2 --port 1234 --protocol http
s --server-cert-file ~/server.pem --id 4690c02c-769b-43c6-88a0-d58e3634027d
```

Example Response

```
{
  "id": "4690c02c-769b-43c6-88a0-d58e3634027d",
  "uri": "kylo:kylo:connectionmgmt:hadoop-knox-node:hadoopconnection-4690c02c-7
69b-43c6-88a0-d58e3634027d",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-28T10:48:36.907928008Z",
  "hostname": "node2",
  "port": "1234",
  "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIG1TCCBb2gAwIBAgIQBVfIC
ygm6F7ChFEkylreTANBgkqhkiG9w0BAQsFADBw\nnMQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNl
cnQgSW5jMRkwFwYDVQQLExB3\nd3cuZGlnaWNlcnQuY29tMS8wLQYDVQQDEyZEaWdpQ2VydCBTSEeYIEh
pZ2ggQXNz\ndXJhbmNlIFNlcnZlciBDQTAEFw0yMDA1MDUwMDAwMDBaFw0yMjA1MTAxMjAwMDBa\nnMGYx
CzAJBgNVBAYTA1VTMRMwEQYDVQQIEWpDYWxpZm9ybmlhMRYwFAYDVQQHEw1T\nnYw4gRnJhbmNpc2NvMRU
wEwYDVQQKEWxHaXRIdWIsIEluYy4xEzARBgNVBAMTCmdp\ndGh1Yi5jb20wgGElMA0GCSqGSIs3DQEBAAQ
UAA4IBDwAwggEKAoIBAQC7MrTQ2J6a\nnnox5KUwrgO9cQ9STO5R4/zBUxxvI5S8bmc0QjWfIVAwHWuT0B
n/H1oS0LM0tTkQm\nnArRqN77v9McVB8MWTGsmGQnS/1kQRFuKiYGUHf7iX5pfijbYsOkfb4AiVKysKUN
V\nnUtgVvpJoe5RWURjQp9XDWkeo2DzGHXLcBAdrM8VLC6H1/D9SXdVruxKqduLKR41\nnZ/6dlSDdeYlg
Cnhz3ChlpYbFMfsTCTamw+AtRtwlK3b2rfTHffhowjuzM15UKt+b\nnrr/cEB1AjQTva8rutYU9K9ONg
l+pG2u7Bv516DwmNy8xz9wOjTeOpeh0M9N/ewq8\ncgbr87LFaxilAgMBAAGjggNzMIIDbzAfBgNVHSME
GDAWgBRRaP+QrwIHDtZm2WVv\nnYqISuFlyOzAdBgNVHQ4EFgQUYwLSXQJf943VWhKedhE2loYsikgwJQY
DVR0RBB4w\nnHIIKZ210aHViLmNvbYIod3d3LmdpdGh1Yi5jb20wDgYDVROPAQH/BAQDAGwGMB0G\nnA1Ud
JQQWMBQGCSGAQUFBwMBBggrBgEFBQcDAjB1BgNVHR8EbjBsMDSGMAwhi5o\nndHRwOi8vY3JsMy5kaWd
pY2VydC5jb20vc2hhMi1oYS1zZXJ2ZXItZzYuY3JsMDSG\nnMqAwhi5odHRwOi8vY3JsNC5kaWdpY2VydC
5jb20vc2hhMi1oYS1zZXJ2ZXItZzYuY3JsMEwGA1UdIARFMEMwNwYJYIZIAyB9baEBMCowKAYIKwYBB
QUHAQEWHGh0dHBz\nnOi8vd3d3LmRpZ21jZXJ0LmNvbS9DUFMwCAYGZ4EMAQICMIGDBggrBgEFBQcBAQR
3\nnMHUwJAYIKwYBBQUHMAGGGh0dHA6Ly9vY3NwLmRpZ21jZXJ0LmNvbTBnbGgrBgEF\nnBQcwoAoZBaHR0
cDovL2NhY2VydHMuZGlnaWNlcnQuY29tL0RpZ21lDZXJ0U0hBMkhp\nnZ2hBc3N1cmFuY2VTZXJ2ZXJDQS5
jcnQwDAYDVROTAQH/BAIwADCCAXwGCisGAQQB\nn1nkCBAIEggFsBIIBaAFmAHUAKXm+8J45OSHwVnOfY6
V35b5XfZxgCvj5TV0mXCVd\nnx4QAAAFx5ltprwAABAMARjBEAiAuWGCWxN/M0Ms3K0sqFjDMHT8Aq0SlH
fQ68KDg\nnrVU6AAIgDA+2EB0D5W5r0i4Nhljx6ABLIByzrEdfcxiOD/o6//EAdQAIRUHVWUk\nnVpY/o
S/x922G4CMmY63AS39dxoNcbuIPAgAAAXHmW2nTAAAEAwBGMEQCIBp+XQKa\nnUDiPHwjBxdv5qvgYALKa
ysKqMF60gqem8iPRAiAk9Dp5+VBUXfSHqyW+tvShUigh\nnndopccf8Gs21KJ4jXgB2AFGjsPX9AXmcVm2
4N3iPDKR6zBsny/eeiEKaDf7UiwXl\nnAAABceZbabsAAAQDAEcwRQIgd/5HcxT4wfnV8zavwxjYkw2TYB
AuRCcqp1SjWKF\nn4EoCIQDHSTHxnbpxWFbP6v5Y6nGFZCDjaHgd9HrzUv2J/DaacDANBgkqhkiG9w0B\nnAQsFAAOCAQEAhjkPnBW4r+jR3gg6RA5xICTW/A5YMcyqtK0c1QzFr8S7/l+skGpC\nnyCHRjFfRlDeyKq
gabvLRT6YvvM862MGfMMDsk+sKWtzLbDIcYG7sbviGpU+gtG1q\nnB0ohWNApfWWKyNpquqvwdSezAEBvh
cUT5idzbK7q45bQU9vBIWgQz+PYULAU7KmY\nnz7jOYV09o22TNMQT+hFmo92+EBlwSeIETYESHy5ZxixT
RTvu9hP00CyEbiht5OTK\nn5EiJG6vsIh/uEtRsdnMCxV06W2f20Af4iSFo0uk6clryHefh08FcwA4pSN
UaPyi\nnPb8YGQ6o/blejFzo/OSiUnDueafSJ0p6SQ==\nn-----END CERTIFICATE-----",
  "protocol": "https"
}
```

Getting Node Details in a Hadoop Knox Connection

To get node details in a Hadoop Knox connection, run:

Syntax

```
ksctl connectionmgmt hadoop node get --id <Connection-Name/ID> --node-id <Node ID/Hostname>
```

Example Request

```
ksctl connectionmgmt hadoop node get --id 4690c02c-769b-43c6-88a0-d58e3634027d --node-id node2
```

Example Response

```
{
  "id": "4690c02c-769b-43c6-88a0-d58e3634027d",
  "uri": "kylo:kylo:connectionmgmt:hadoop-knox-node:hadoopconnection-4690c02c-769b-43c6-88a0-d58e3634027d",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-28T10:48:36.907928Z",
  "hostname": "node2",
  "port": "1234",
  "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIG1TCCBb2gAwIBAgIQBVfICygmG6F7ChFEkylreTANBgkqhkiG9w0BAQsFADBw\n/nMQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlNaUN1cnQgSW5jMRkwFwYDVQQLExB3\n/nd3cuZGlnaWNlcnQuY29tMS8wLQYDVQQDEyZEAwdpQ2VydCBTSEEyIEhpZ2ggQXNz\n/ndXJhbmNlIFNlcnZlciBDQTAEFw0yMDA1MDUwMDAwMDBaFw0yMjA1MTAxMjAwMDBa\n/nMGYxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEWpDYWxpZm9ybmlhMRYwFAYDVQQHEw1T\n/nYW4gRnJhbmNpc2NvMRUwEwYDVQQKEWxHaXRIdWIsIEluYy4xezARBgNVBAMTCmdp\n/ndGh1Yi5jb20wggeiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC7MrTQ2J6a\n/nnox5KUwrgO9cQ9STO5R4/zBUxxvI5S8bmc0QjWfIVAwHWuT0Bn/H1oS0LM0tTkQm\n/nARrqN77v9McVB8MWTGsmGQnS/1kQRFuKiYGUHf7iX5pfijbYsOkfb4AiVKysKUN\n/V\n/nUtgVvpJoe5RWURjQp9XDWkeo2DzGHXLcBdadrM8VLC6H1/D9SXdVruxKqduLKR41\n/nZ/6dlSDdeY1gCnhz3ChlpYbfMfsTCTamw+AtRtWlK3b2rfTHffhowjuzM15UKt+b\n/nrr/cEB1AjQTva8rutYU9K9ONg1+pG2u7Bv516DwmNy8xz9wOjTeOpeh0M9N/ewq8\ncgbr87LFaxilAgMBAAGjggNzMIIDbzAfBgNVHSMEGDAWgBRRAp+QrwIHDtZm2WV\nk/nYqISuFlyOzAdBgNVHQ4EFgQUYwLSXQJf943VWhKedhE2loYsikgwJQYDVR0RBB4w\n/nHIIKZ210aHViLmNvbYlOdz3d3LmdpdGh1Yi5jb20wDgYDVDR0PAQH/BAQDAgWgMB0G\n/nA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjB1BgNVHR8EbjBsMDSgMqAw\nhi5o/ndHRwOi8vY3JsMy5kaWdpY2VydC5jb20vc2hhMlloYS1zZXJ2ZXItZzYuY3JsMDSg\n/nMqAw\nhi5odHRwOi8vY3JsNC5kaWdpY2VydC5jb20vc2hhMlloYS1zZXJ2ZXItZzYuY3JsMEwGA1UdIARFMEMwNwYJYlZlAYb9bAEBMCowKAYIKwYBBQUHAgEWHGh0dHBz\n/nOi8vd3d3LmRpZ21jZXJ0LmNvbS9DUFMwCAYGZ4EMAQICMIGDBggrBgEFBQcBAQR3\n/nMHUwJAYIKwYBBQUHMAggGGGh0dHA6Ly9vY3NwLmRpZ21jZXJ0LmNvbTBnbGgrBgEF\n/nBQcwoAoZBaHR0cDovL2NhY2VydHMuZGlnaWNlcnQuY29tL0R3d3J0U0hBMkhp\n/nZ2hBc3N1cmFuY2VTZXJ2ZXJ0Q295cnQwDAYDVDR0TAQH/BAIwADCCAXwGCisGAQQB\n/n1nkCBAIEggFsBIIBaAFmAHUAKXm+8J45OSHwVnOfY6V35b5XfZxgcVj5TV0mXCVd\n/nx4QAAAFx5ltprwAABAMARjBEAiAuWGCWxN/M0Ms3KOsqFjDMHT8Aq0S1HfQ68KDg\n/nrVU6AAIgdA+2EB0D5W5r0i4Nhljx6AB1IByzrEdfcxiOD/o6//EAdQAiRUUHWVUK\n/nVpY/oS/x922G4CMmY63AS39dxoNcbuIPAgAAAXHmW2nTAAAEAwBGMEQCIBp+XQKa\n/nUDiPHwjBxdv5qvgvYALKaySkqMF60gqem8iPRAiAk9Dp5+VBUXfSHqyW+tvShUigh\n/nndopccf8Gs21KJ4jXgB2AFGjsPX9AXmcVm24N3iPDKR6zBsny/eeiEKaDf7UiwXl\n/nAAABceZbAhsAAQAQDAECwRQIGd/5HcxT4wfNV8zavwxjYkw2TYBAuRCcqp1SjWKFn\n/n4EoCIQDHSTHxnbpxWFbP6v5Y6nGFZCDjaHgd9HrzUv2J/DaacDANBgkqhkiG9w0B\n/nAQsFAAOCAQEAhjKPNBW4r+jR3gg6RA5xICTW/A5YMcyqtK0c1QzFr8S7/1+skGpC\n/nYCHRjFfRlDeyKqgabvLRT6YvvM862MGfMMDsk+sKWtzLbDIcYG7sbviGpU+gtG1q\n/nB0ohWNApfWWKYnpquqvwdSEzAEBvh
```

To update a node in a Hadoop Knox Connection, run:

```
ksctl connectionmgmt hadoop node modify --id <Connection-Name/ID> --node-id <Node ID/Hostname> --hostname <Server-Hostname> --port <Server-Port> --path <Sever-Path> --protocol <Communication-Protocol> --sever-cert-file <Server-Certificate-File>
```

```
ksctl connectionmgmt hadoop node modify --id 4690c02c-769b-43c6-88a0-d58e3634027d
--node-id node2 --port 5678
```

```
{
  "id": "4690c02c-769b-43c6-88a0-d58e3634027d",
  "account": "kylo:kylo:admin:accounts:kylo",
  "hostname": "node2",
  "port": "5678",
  "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIGlTCCBb2gAwIBAgIQBVfICygmG6F7ChFEkylreTANBgqhkiG9w0BAQsFADBW\n\\nMQswCQYDVQQGEwJVUzEVMBMGAUEChMMRGlnaUN1cnQgSW5jMRkwFwYDVQQLExB3\\nd3cuZGlnaWN1cnQuY29tMS8wLQYDVQQDEyZEaWdpQ2VydCBTSEEyIEhpZ2ggQXNz\\ndXJhbmlhbnRlcjEwLmF1dDQwMDA1MDUwMDAwMDBaFw0yMjAlMTAxMjAwMDBa\\nmGYxCzAJBgNVBAYTAhVMTCRMEwEQYDVQIEwPDIWxpZm9ybmlhMRYYFAIDVQqHEw1T\\nyW4gRnJhbmlhbnRlcjEwLmF1dDQwMDA1MDUwMDAwMDBaFw0yMjAlMTAxMjAwMDBa\\nwEwYDVQQKEwxHaXRIdWIsIEluYy4xEzARBGNVBAMTCmdp\\ndGhlYi5jb20wggeEiMAOGCSqGSIlb3DQEBAQUAA4IBDwAwggEKAOIBAQC7MrTQ2J6a\\nnox5KUwrqO9cQ9STO5R4/zBUxxvI5S8bmc0QjWfIVAWHwuT0Bn/Hl0S0LM0tTkQm\\nARrqN77v9MCvB8MWTGsmGqnS/lkQRfuKiYGUHF7ix5pfijbYsOkfb4AiVKysKUNV\\nUtgvVvpJoe5RWURjQp9XDWkeo2DzGHXLcBDadrM8VLC6H1/D9SXdvruXKqduLKR4l\\nz/6dlSDdeYlgCnhz3ChlpYbfmfsTCTamw+AtRtlwK3b2rfTHffhowjuzM15UKt+b\\nnrr/cEBlaJqTva8rutYU9K9ONG1+pG2u7Bv516DwmNy8xz9wojTeOpeh0M9N/ewq8\\ncgbr87LFaxi1AgMBAAGjggNzMIIDbzAfBgNVHSMEGDAWgBRRAp+QrwIHdTzM2VWk\\nyqISuflyOzAdBgNVHQ4EFggQUYwLSXQJf943VWhKedhE2loYsikgwJQYDVR0RBBAw\\nHIKZ210aHViLmNvbYIod3d3LmdpdGhlYi5jb20wdG9YDVR0PAQH/BAQDAgWgMB0G\\na1UdJQQWMBQGCGCsGAQUFBwMBBggrBgEFBQcDAjB1BgNVHR8EbjBSMDSGmMqAwhi5o\\ndhRWOi8vY3JsMy5kaWdpY2VydC5jb20vc2hhMi1oYS1zZXJ2ZXItZzYuY3JsMDSG\\nmMqAwhi5odHRWOi8vY3JsNC5kaWdpY2VydC5jb20vc2hhMi1oYS1zZXJ2ZXItZzYuY3JsMEwGA1UdIARFMEMwNwYJYIZIAyB9bAEBCowKAyIKwYBBQUHAQEWHGh0dBHz\\noI8vd3d3LmRpZ21ljZXJ0LmNvbS9DUFMwCAYGZ4EMAQICMIGDBggrBgEFBQcBAQR3\\nMHUwJAyIKwYBBQUHMAGGGGh0dHA6Ly9vY3NwLmRpZ21ljZXJ0LmNvbTBnbGgrBgEFF\\nbQcwAoZBaHR0cDovL2NhY2VydHMuZGlnaWN1cnQuY29tL0RpZ21lZDZlODU0bmhmKh\\nz2hBc3N1cmFuY2VTZXJ2ZXJDQS5jcjcnQwDAYDVR0TAQH/BAlwADCCAxcGcisGAQQB\\nl1nkCBAIEggFsBIIBAFmAUAkXm+8J45OSHwVnofY6V35b5XfxzgCvj5TV0mXCVD\\nx4QAAAFx5ltprwAABAMARjBEAiAuWGCwXN/MOMs3KOsfjDMHT8AQoS1hfQ68KDg\\nrVU6AAIgDA+2EB0D5W5r0i4Nh1jx6AbLIByzrEdfcxiOD/o6//EADQAIRUUHWVuk\\nVpy/o
```



```
S/x922G4CMmY63AS39dxoNcbuIPAgAAAXHmW2nTAAAEAwBGMEQCIBp+XQKa\nUDiPHwjBxdv5qvgyALKa
ysKqMF60gqem8iPRAiAk9Dp5+VBUXfSHqyW+tVShUigh\nndopccf8Gs21KJ4jXgB2AFGjsPX9AXmcVm2
4N3iPDKR6zBsny/eeiEKaDf7UiwXl\nAAABceZbahsAAAQDAEcwRQIgd/5HcxT4wfNV8zavwxjYkw2TYB
AuRCcqp1SjWKF\n4EoCIQDHSTHxnbpxWFbP6v5Y6nGFZCDjaHgd9HrzUv2J/DaacDANBgkqhkiG9w0B\
nAQsFAAOCAQEAAhjkPnBW4r+jR3gg6RA5xICTW/A5YMcyqtK0c1QzFr8S7/l+skGpC\nnyCHrJfFrLDeyKq
gabvLRT6YvVM862MGfMMDsk+sKWtzLbDIcYG7sbviGpU+gtG1q\nB0ohWNApfWWKyNpquqvwdSEzAEBvh
cUT5idzbK7q45bQU9vBIWgQz+PYULAU7KmY\nnz7jOYV09o22TNMQT+hFmo92+EBlwSeIETYESHy5ZxixT
RTvu9hP00CyEbiht5OTK\n5EiJG6vsIh/uEtRsdenMCxV06W2f20Af4iSFo0uk6c1ryHefh08FcwA4pSN
UaPyi\nPb8YGQ6o/blejFzo/OSiUnDueafSJ0p6SQ==\n-----END CERTIFICATE-----",
    "protocol": "https"
}
```

Deleting a Node in a Hadoop Knox Connection

To delete a node in a Hadoop Knox connection, run:

Syntax

```
ksctl connectionmgmt hadoop node delete --id <Connection-Name/ID> --node-id <Node
ID/Hostname>
```

Example Request

```
ksctl connectionmgmt hadoop node delete --id c4a11fc1-d226-4f19-8d03-5911198e89e5
--node-id e4a11fc1-d226-4f19-8d03-5911198e89e4
```

There will be no response if node is deleted successfully.

Getting List of all Nodes in a Hadoop Knox Connection

To list all the nodes in a Hadoop Knox connection, run:

Syntax

```
ksctl connectionmgmt hadoop node list --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt hadoop node list --id 4690c02c-769b-43c6-88a0-d58e3634027d
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
```

[illegible]

```

VKysKUNV\nUtgVvpJoe5RWURjQp9XDWkeo2DzGHXLcBDadrM8VLC6H1/D9SXdVruxKqduLKR41\nZ/6dl
SDdeY1gCnhz3ChlpYbFMfsTCTamw+AtRtwlK3b2rfTHffhowjuzM15UKt+b\nrr/cEB1AjQTva8rutYU9
K9ONgl+pG2u7Bv516DwmNy8xz9wOjTeOpeh0M9N/ewq8\ncgbr87LFaxilAgMBAAGjggNzMIIDbzAfBgN
VHSMEGDAWgBRRaP+QrwIHdTzM2WVvk\nYqISuFlyOzAdBgNVHQ4EFgQUYwLSXQJf943VWhKedhE2loYsik
gwJQYDVR0RBBA4w\nnHIIKZ2l0aHViLmNvbYlOdz3LmdpdGh1Yi5jb20wDgYDVR0PAQH/BAQDAgWgMB0G\
nA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjB1BgNVHR8EbjBsMDSgMqAwhi5o\ndHRwOi8vY3JsMy
5kaWdpY2VydC5jb20vc2hhMi1oYS1zZXJ2ZXItZzYuY3JsMDSg\nmMqAwhi5odHRwOi8vY3JsNC5kaWdpY
2VydC5jb20vc2hhMi1oYS1zZXJ2ZXItZzYuY3JsMDEwGA1UdIARFMEMwNwYJYIZIAyB9bAEBMCowKAYI
KwYBBQUHAgEWHGh0dHBz\nOi8vd3d3LmRpZ21jZXJ0LmNvbS9DUFMwCAYGZ4EMAQICMIGDBggrBgEFBQc
BAQR3\nnMHUwJAYIKwYBBQUHMAGGGGh0dHA6Ly9vY3NwLmRpZ21jZXJ0LmNvbTBnbGgrBgEF\nnBQcwAoZB
aHR0cDovL2NhY2VydHMuZGlnaWNlcnQuY29tL0RpZ21lDZXJ0U0hBMkhp\nnZ2hBc3N1cmFuY2VTZXJ2ZXJ
DQS5jcncwDAYDVR0TAQH/BAIwADCCAXwGCisGAQQB\nn1nkCBAIEggFsbIIBaAFmAHUAKXm+8J45OShwVn
OfY6V35b5XfZxgCvj5TV0mXCVd\nnx4QAAAFx5ltpwAABAMARjBEAiAuWGCWxN/M0Ms3K0sqFjDMHT8Aq
0SlHfQ68KDg\nnrVU6AAIgdA+2EB0D5W5r0i4Nhljx6ABlIByzrEdfcxiOD/o6//EAdQAIRUHHWVuk\nnVp
Y/oS/x922G4CMmY63AS39dxoNcbuIPagAAAXHmW2nTAAAEAwBGMEQCIBp+XQKa\nnUDiPHwjBxdv5qvgY
LKaysKqMF60gqem8iPRAiAk9Dp5+VBUXfSHqyW+tVShUigh\nnndopccf8Gs21KJ4jXgB2AFGjsPX9AXmc
Vm24N3iPDKR6zBsny/eeiEKaDf7UiwXl\nnAAABceZbabsAAAQDAECwRQIgd/5HcxT4wfNV8zavwxjYkw2
TYBAuRCCqp1SjWKF\nn4EoCIQDHSTHxnbpxWFbP6v5Y6nGFZCDjaHgd9HrzUv2J/DaacDANBgkqhkiG9w
0B\nnAQsFAAOCAQEAhjKPnBW4r+jR3gg6RA5xICTW/A5YMcyqtK0clQzFr8S7/l+skGpC\nnyCHRjFfrLDe
yKqgabvLRT6YvvM862MGfMMDsk+sKWtzLbDIcYG7sbviGpU+gtG1q\nnB0ohWNApfWWKyNpquqvwdSEzAE
BvhcUT5idzbK7q45bQU9vBIWgQz+PYULAU7KmY\nnz7jOYV09o22TNMQT+hFmo92+EBlwSeIETYESHy5Zx
ixTRTvu9hP00CyEbiht5OTK\nn5EiJG6vsIh/uEtRsdnMCxV06W2f20Af4iSFo0uk6clryHefh08FcwA4
pSNUaPyi\nnPb8YQG6o/blejFzo/OSiUnDueafSJ0p6SQ==\nn-----END CERTIFICATE-----",
    "path": "/gateway/default/webhdfs/v1",
    "protocol": "https"
  }
}
}

```

Loki

Loki connections to the CipherTrust Manager can be configured using the following:

- [Managing Loki Connections using GUI](#)
- [Managing Loki Connections using kscctl](#)

Note

CipherTrust Manager can have a total of 64 log forwarder connections. Each [Elasticsearch](#), [Loki](#), and [Syslog](#) connection is counted towards the 64 connection total.

Managing Loki Connections using GUI

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.

-
4. On the Add Connection screen, select category as **Log Forwarder**.
 5. Select Log Forwarder Type as **Loki** and click **Next**.
 6. Specify connection **Name** and **Description** and click **Next**.
 7. Configure the below parameters.

- **Host:** IP address or hostname of the Loki server.
- **Port:** port number for connecting to the Loki server.

Loki Grafana doesn't directly support authentication, but you can configure authentication values for requests between CipherTrust Manager and a reverse proxy, which forwards requests to Loki Grafana. Check the [Loki Grafana deployment for the authentication required](#) for incoming data.

- **HTTP User:** name of the HTTP user for basic authentication.
- **HTTP Password:** password of the HTTP user.
- **Transport:** protocol for transport. Either TCP or TLS. TLS requires additional settings.

Settings for TLS:

You must provide **CA Certificate** or select **Skip Certificate Verification**

- **CA Certificate:** either upload the CA certificate or paste the certificate content. This is a CA which directly or indirectly signs the Elasticsearch server certificate.
 - **File Upload:** select and click **Upload Certificate** to upload the trusted CA certificate from your machine. In TLS mode, the CA certificate must be in PEM format.
 - **Text:** select and paste the certificate content in the text field.
 - **Skip Certificate Verification:** select if the TLS verification of the CA certificate is not required.
8. Click **Test Credentials** to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.
 9. Click **Next** to move to the [Add Products](#) screen of the Add Connection wizard.

Managing Loki Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete a Loki connection
- List all Loki connections
- Test an existing Loki connection

- Test the new Loki connection parameters before establishing the connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Unique name of the connection.
description	Optional	Connection description.
products	Optional	List of products.
host	Mandatory	Host of the log-forwarder server.
port	Mandatory	Port of the log-forwarder server.
transport	Optional	Transport mode for sending data, supports "tls" and "tcp". The "tls" mode requires either a trusted CA certificate or insecure TLS skip verify to be set to true. The default transport mode is "tcp".
json-file	Optional	meta information in json format. This information is provided in pair. For example, <code>--meta "{\"color\":\"blue\", \"foo\":\"bar\"}"</code> .
http-user	Optional	HTTP basic auth username.
http-password	Optional	HTTP basic auth password.
ca_cert	Optional	CA certificate in PEM format.
insecure-skip-verify	Optional	In TLS mode, skips server certificate validation. This setting should only be used for testing.

The Loki connections can be authenticated or unauthenticated. Authentication can include username and password or TLS authentication or both. Check the [Loki Grafana deployment for the authentication required](#) for incoming data. Loki Grafana doesn't directly support authentication, so authentication takes place between CipherTrust Manager and a reverse proxy, which forwards requests to Loki Grafana.

The `tls` mode requires either a trusted CA certificate in the PEM format or `--insecure-skip-verify` set to `true`. By default, the CipherTrust Manager attempts to validate the reverse proxy's server certificate using the provided CA. The external CAs already present on the CipherTrust Manager are not used. Use `--ca-cert` to provide a CA which directly or indirectly signs the server certificate. Use the `--insecure-skip-verify` flag to indicate that CipherTrust Manager does not verify Elasticsearch's presented server certificate.

Creating a Loki Connection

To create a Loki connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder loki create --name <connection-name> --product
```

```
s <products-name> --description <description> --host <host> --port <port> --transport <transport-protocol> --http-user <username> --http-password <password> --ca-cert <ca-cert> --meta <Key:Value> --insecure-skip-verify
```

The supported transport mode for sending data are `tls` and `tcp`. The `tls` mode requires either a trusted CA certificate in the PEM format or `--insecure-skip-verify` set to `true`.

Example Request

```
ksctl connectionmgmt log-forwarder loki create --name loki-conn-1 --description conn-description --host 127.0.0.1 --port 1234 --transport tcp --http-user admin --http-password password
```

Example Response

```
{
  "id": "9d3af367-d4a3-4838-8663-ce07d3e88353",
  "uri": "kylo:kylo:connectionmgmt:connections:loki-conn-1-9d3af367-d4a3-4838-8663-ce07d3e88353",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-04T07:53:49.838640861Z",
  "updatedAt": "2022-05-04T07:53:49.837431591Z",
  "service": "loki",
  "category": "log-forwarders",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "loki-conn-1",
  "description": "conn-description",
  "host": "127.0.0.1",
  "port": 1234,
  "loki_params": {
    "insecure_tls_skip_verify": false,
    "transport": "tcp",
    "http_user": "admin"
  }
}
```

Getting Details of a Loki Connection

To get details of a Loki connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder loki get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt log-forwarder loki get --id 9d3af367-d4a3-4838-8663-ce07d3e88353
```

Example Response

```
{
  "id": "9d3af367-d4a3-4838-8663-ce07d3e88353",
  "uri": "kylo:kylo:connectionmgmt:connections:loki-conn-1-9d3af367-d4a3-4838-8663-ce07d3e88353",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-04T07:53:49.838641Z",
  "updatedAt": "2022-05-04T07:53:49.837432Z",
  "service": "loki",
  "category": "log-forwarders",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "loki-conn-1",
  "description": "conn-description",
  "host": "127.0.0.1",
  "port": 1234,
  "loki_params": {
    "insecure_tls_skip_verify": false,
    "transport": "tcp",
    "http_user": "admin"
  }
}
```

Updating a Loki Connection

To update a Loki connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder loki modify --id <Connection-Name/ID> --products <products-name> --description <description> --host <host> --port <port> --transport <transport-protocol> --http-user <username> --http-password <password> --ca-cert <ca-cert> --meta <Key:Value> --insecure-skip-verify
```

The supported transport mode for sending data are `tls` and `tcp`. The `tls` mode requires either a trusted CA certificate in the PEM format or `--insecure-skip-verify` set to `true`.

Example Request

```
ksctl connectionmgmt log-forwarder loki modify --id 9d3af367-d4a3-4838-8663-ce07d3e88353 --host 127.0.0.1
```

Example Response

```
{
  "id": "9d3af367-d4a3-4838-8663-ce07d3e88353",
  "uri": "kylo:kylo:connectionmgmt:connections:loki-conn-1-9d3af367-d4a3-4838-8663-ce07d3e88353",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-04T07:53:49.838641Z",
  "updatedAt": "2022-05-04T08:17:20.615258139Z",
  "service": "loki",
  "category": "log-forwarders",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
}
```

```

    "name": "loki-conn-1",
    "description": "conn-description",
    "host": "127.0.0.1",
    "port": 1234,
    "loki_params": {
      "insecure_tls_skip_verify": false,
      "transport": "tcp",
      "http_user": "admin"
    }
  }
}

```

Deleting a Loki Connection

To delete a Loki connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder loki delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt log-forwarder loki delete --id 9d3af367-d4a3-4838-8663-ce07d3e88353
```

Example Response

There will be no response if the Loki connection is deleted successfully.

Getting List of Loki Connections

To list all the Loki connections, run:

Syntax

```
ksctl connectionmgmt log-forwarder loki list
```

Example Request

```
ksctl connectionmgmt log-forwarder loki list
```

Example Response

```

{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "9d3af367-d4a3-4838-8663-ce07d3e88353",
      "uri": "kylo:kylo:connectionmgmt:connections:loki-conn-1-9d3af367-d4a

```



```

3-4838-8663-ce07d3e88353",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-04T07:53:49.839242Z",
  "updatedAt": "2022-05-04T07:53:49.839242Z",
  "service": "loki",
  "category": "log-forwarders",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "loki-conn-1",
  "description": "conn-description",
  "host": "127.0.0.1",
  "port": 1234
}
]
}

```

Testing an Existing Loki Connection

Note

Test Connection verifies the status of the server certificate if it is revoked or not by using the [Certificate Revocation Lists \(CRLs\)](#). If the certificate is found to be revoked, the test connection fails. If an error occurs during the evaluation of the certificate's revocation status, the connection remains in a soft fail state and test connection is allowed to continue.

To test an existing Loki connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder loki test --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt log-forwarder loki test --id 9d3af367-d4a3-4838-8663-ce07d3e88353
```

Example Response

```
{
  "connection_ok": true
}
```

Testing a New Loki Connection

Note

Test Connection verifies the status of the server certificate if it is revoked or not by using the [Certificate Revocation Lists \(CRLs\)](#). If the certificate is found to be revoked, the test connection fails. If an error occurs during the evaluation of the certificate's revocation status, the connection remains in a soft fail state and test connection is allowed to continue.

To test the parameters of a New Loki connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder loki test --host <host> --port <port> --transport <transport-protocol> --http-user <username> --http-password <password> --ca-cert <ca-cert> --insecure-skip-verify
```

Example Request

```
ksctl connectionmgmt log-forwarder loki test --host 127.0.0.1 --port 1234 --transport tcp --http-user admin --http-password password
```

Example Response

```
{
  "connection_ok": true
}
```

Salesforce

The Salesforce connections to the CipherTrust Manager can be configured using the following:

- [Managing Salesforce Connections using GUI](#)
- [Managing Salesforce Connections using ksctl](#)

Note

If you wish to use external certificate authentication, you must first [create a valid external certificate](#).

Managing Salesforce Connections using GUI

- **Client ID** - application ID of the Salesforce application. It can be used with Client Credential, Certificate, or

Password to authenticate the application.

Note

When creating multiple connections to the same Salesforce organization from different domains:

- **For certificate-based authentication**, you **must** create a new "Connected App" in Salesforce for each connection. This provides a unique Client ID, as Salesforce does not allow reusing a single Client ID with multiple certificates.
- **For client credential-based authentication**, you can reuse the same Client ID for multiple connections.

- **Cloud Name** - the name of the Salesforce cloud to connect to. Currently, only the following options are available:
 - Salesforce Sandbox Cloud
 - Salesforce Cloud
- **Hostname**- The hostname of the Salesforce instance to connect to. For example, test-site-aesf24.sf-onp-dc-test-02.ml.
- **Authentication** - you can use the following authentication methods.

- **Client Credential (My Domain)** - this authentication method uses Client Secret and Domain Name for the given Client ID to enable the communication between Salesforce and the CipherTrust Manager. The Domain Name is mapped to [My Domain](#) in Salesforce login and application URLs.

The client credentials don't require "username" and "password" for authentication. For details, refer to [Salesforce documentation](#).

- **Certificate** - this authentication method is used to enable password-less communication between Salesforce and the CipherTrust Manager. To do so:
 - Select the **Certificate** radio button.
 - Enter the **Username**.
 - Select **Application** or **External** as the Certificate Type.

The **Application** certificate type is generated by CipherTrust Manager and self-signed.

The **External** certificate type is a pre-existing certificate generated on CipherTrust Manager and then signed by a CA local or external to the CipherTrust Manager. To use this option, you first need to [create a valid external certificate](#).

Configure Application Certificate

- Enter the authorization server's URL for the **Audience** value for the Salesforce.com cloud: <https://login.salesforce.com>, <https://test.salesforce.com>. The audience identifies the authorization server as the intended audience.
- Click the **Generate and Download** button.
- Upload the downloaded certificate on Salesforce for the provided Client ID.
- Once the upload is done, verify the **Certificate Subject** on the CipherTrust Manager and Salesforce. Both the Certificate Subjects must match.
- (Optional) Specify **Certificate Duration in Days**. The default certificate duration is 730 days (2 years).

Configure External Certificate

- Enter the authorization server's URL for the **Audience** value for the Salesforce.com clouds: <https://login.salesforce.com>, <https://test.salesforce.com>. The audience identifies the authorization server as the intended audience.
- Provide **External Certificate**. Ensure that you have fulfilled the prerequisites to [create a valid external certificate](#). Use either of the methods:
- Select **File Upload** and click the **Upload Certificate** to upload the external certificate as a file.
- Select **Text** and paste the certificate contents in the text box.

Note

The CipherTrust Manager allows you to modify the external certificate in the existing connection. Any unused certificate will be automatically deleted after 24 hours.

- **Password** – this authentication method uses the Client Secret, Username, and Password for the given Client ID to enable the communication between Salesforce and the CipherTrust Manager.

Note

- If you have enabled the Multi-Factor authentication on the API, password authentication will not work.
- Before creating a Salesforce connection using "Password" based authentication, ensure that the "Allow OAuth Username-Password Flows" setting is enabled on the Salesforce cloud (**Salesforce**

> **Setup** > **Settings** > **Identity** > **OAuth and OpenID Connect Settings** > **Allow OAuth Username-Password Flows**).

- (Optional) **Enable Mutual TLS**.

Click **Next** to move to the [Add Products](#) screen of the Add Connection wizard.

Note

Currently, the only product supported for Salesforce connection is Cloud Key Manager.

Click **Next** to move to the next step.

Note

For Salesforce Organizations, you can provide the [Mutual TLS Certificate](#) details, you need to **Enable Mutual TLS**.

Mutual TLS Certificate

In the Mutual TLS Certificate section, provide the TLS Client Certificate details.

Select **File Upload** and click **Upload Certificate** to upload the certificate (in PEM format). You can also select **Text** and enter the certificate in the text area.

Enter the **Password of TLS Client Certificate**.

Click **Next** to move to the [Add Products](#) screen of the Add Connection wizard.

Managing Salesforce Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete an Salesforce connection
- List all Salesforce connections
- Test an existing Salesforce connection
- Test parameters for a Salesforce connection

Authentication methods

The Salesforce connection can be created using the following authentication methods:

- Password
- Certificate
- Client Credential (My Domain)

Creating a Salesforce Connection

Using Password

Note

Before creating a Salesforce connection using "Password" based authentication, ensure that the "Allow OAuth Username-Password Flows" setting is enabled on the Salesforce cloud (**Salesforce > Setup > Settings > Identity > OAuth and OpenID Connect Settings > Allow OAuth Username-Password Flows**).

It is consumer secret based authentication. Following fields are used for authentication:

- name - username to access the Salesforce server
- client_id - application ID of the Salesforce application
- client_secret - consumer secret for the Salesforce application
- password - password for the Salesforce account

To create a Salesforce connection, run:

Syntax

```
ksctl connectionmgmt salesforce create --name <connection-name> --products <products-names> --clientid <salesforce-client-id> --secret <salesforce-client-secret> --username <salesforce-client-username> --conn-password <salesforce-client-password> --cloudname <salesforce-cloud-name> --salesforce-hostname <salesforce-hostname>
```

Example Request 1

```
ksctl -v connectionmgmt salesforce create --name salesforce2 --products cckm --clientid "3MVG9c1ghSpUbLl_2HbT5XAlPgXdC.voJEM.D4jfqQipoLHa.8rtSE09Zw6qX0MhVdK3V7LrASTjEmBipQwkv" --username myuser --cloudname "Salesforce Sandbox Cloud" --conn-password mypassword--thaxxxxx --secret 8177EE1E5967F39381DED2D41B363910B865AD881587AA4DF62C5E813988322A
```

Example Response

```
{
  "id": "bad74caa-25e4-42b6-a348-99190db7edf1",
  "uri": "kylo:kylo:connectionmgmt:connections:salesforce2-bad74caa-25e4-42b6-a348-99190db7edf1",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-06-14T12:43:10.528951737Z",
  "updatedAt": "2023-06-14T12:43:10.527954022Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "salesforce2",
  "products": [
    "cckm"
  ],
  "cloud_name": "Salesforce Sandbox Cloud",
  "client_id": "3MVG9c1ghSpUbLl_2HbT5XAlPgXdC.voJEM.D4jfqQipoLHa.8rtSEO9Zw6qX0MhVdK3V7LrASTjEmBIpQwkv",
  "username": "myuser"
}
```

Example Request 2

```
ksctl -v connectionmgmt salesforce create --name salesforce2 --products cckm --clientid "123456" --username xyz@gmail.com --cloudname "Salesforce Sandbox Cloud" --conn-password mypassword--thaxxxxx --secret 8177EE1E5967F39381DED2D41B363910B865AD881587AA4DF62C5E813988322A --salesforce-hostname "xyz-org.my.salesforce.com"
```

Example Response

```
{
  "id": "bad74caa-25e4-42b6-a348-99190db7edf1",
  "uri": "kylo:kylo:connectionmgmt:connections:salesforce2-bad74caa-25e4-42b6-a348-99190db7edf1",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-06-14T12:43:10.528951737Z",
  "updatedAt": "2023-06-14T12:43:10.527954022Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "salesforce2",
  "products": [
    "cckm"
  ],
  "cloud_name": "Salesforce Sandbox Cloud",
  "client_id": "123456",
  "username": "xyz@gmail.com",
  "hostname": "xyz-org.my.salesforce.com"
}
```

Using Certificate

It is certificate-based authentication. Following fields are used for authentication:

- name - username to access the Salesforce server.
- client_id - application ID of the Salesforce application.
- certificate - represents the internal/external certificate.

The Salesforce connection can be created using:

- Internal certificate
- External certificate

You can also [enable mutual TLS \(mTLS\)](#) for a Salesforce connection.

Parameters Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Name of the connection.
products	Mandatory	List of products.
clientid	Mandatory	Application ID of the Salesforce application.
username	Mandatory	Username used to access the server.
cloudname	Mandatory	Name of the Salesforce cloud to connect to.
certificate	Mandatory	Represents the internal/external certificate.
certificate-duration	Optional	Certificate duration in days. The default certificate duration is 730 days (2 years).
salesforce-hostname	Optional	The hostname of the Salesforce instance to connect to.
audience	Optional	the authorization server's URL for the audience value for the Salesforce.com clouds: https://login.salesforce.com , https://test.salesforce.com . The audience identifies the authorization server as the intended audience.

Using Internal certificate

To create a Salesforce connection using internally generated self-signed certificate, run:

Syntax


```
ksctl connectionmgmt salesforce create --name <Connection-Name> --products <Products-Names> --clientid <Salesforce-Key-ID> --username <Salesforce-Client-Secret> --cloudname <Salesforce-Cloud-Name> --certificate <yes/y> --certificate-duration <Certificate-Duration> --salesforce-hostname <salesforce-hostname> --audience <audience>
```

Example Request 1

```
ksctl connectionmgmt salesforce create --name "salesforce-3" --products "cckm" --cloudname "Salesforce Sandbox Cloud" --clientid 123456 --username "xyz@gmail.com" --certificate yes --certificate-duration 730
```

Example Response

```
{
  "id": "93734270-78e0-4a80-b545-dd87d536e1d3",
  "uri": "kylo:kylo:connectionmgmt:connections:salesforce-3-93734270-78e0-4a80-b545-dd87d536e1d3",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-13T07:04:14.160133827Z",
  "updatedAt": "2022-05-13T07:04:14.158769531Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "salesforce-3",
  "products": [
    "cckm"
  ],
  "cloud_name": "Salesforce Sandbox Cloud",
  "client_id": "123456",
  "username": "xyz@gmail.com",
  "certificate": "-----BEGIN CERTIFICATE-----\nMIIFvTCCA6WgAwIBAgIQT3N0S1zNYmmO8lrBpKh9FzANBgkqhkiG9w0BAQsFAADB/\nMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcn5pYTERMA8GA1UEBxMIU2Fu\nnIEpvc2UxXzANBgNVBAoTQlEwPDIWwXpZm9y\nnbmlhMREwD...+3N94oXqI3UDJ7nq29Op1I9IcyzkiigNSZ7lboDGV\nn1TO/xYlVRyMxZyhI3R4a/WyWtvCcCXcnnhe3Z7iVSyBa\nn-----END CERTIFICATE-----\n",
  "certificate_subject": "CN=cckm.thalessecurity.com,OU=CipherTrust,O=Thales,L=San Jose,ST=California,C=US"
}
```

Example Request 2

```
ksctl connectionmgmt salesforce create --name "salesforce-conn" --products "cckm" --cloudname "Salesforce Sandbox Cloud" --clientid 123456 --username "xyz@gmail.com" --certificate yes --certificate-duration 730 --salesforce-hostname "xyz-org.my.salesforce.com"
```

Example Response

```
{
  "id": "93734270-78e0-4a80-b545-dd87d536e1d3",
  "uri": "kylo:kylo:connectionmgmt:connections:salesforce-3-93734270-78e0-4a80-b545-dd87d536e1d3"
```

```

0-b545-dd87d536e1d3",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-13T07:04:14.160133827Z",
  "updatedAt": "2022-05-13T07:04:14.158769531Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "salesforce-conn",
  "products": [
    "cckm"
  ],
  "cloud_name": "Salesforce Sandbox Cloud",
  "client_id": "123456",
  "username": "xyz@gmail.com",
  "certificate": "-----BEGIN CERTIFICATE-----\nMIIFvTCCA6WgAwIBAgIQT3N0S1zNYmmO
81rBpKh9FzANBgqhkiG9w0BAQsFADB/\nMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcn5pYTE
RMA8GA1UEBxMIU2Fu\nnIEpvc2UxDzANBgNVBAoTB1RoYWxlczEUMBIGA1UECzMlQ2lwaGVyVHJlc3QxIT
Af\nBgNVBAMTGGNja20f8fanBqfdIK3nzowvvdvAp7OTjuXO2eHKly321ScuLYVsh+eMx\nniArdzP3McH4
buvPoP/PGXUFTpcrz7rynmU-----CD5chpkAmVGrkjmgY1QWAgeoAEnhva\nnhmhkqsY2xsEQhYo
873gbb9CP+8yJOLgAGUJVRC+e6FVm40Svm7e9ffkaekBAyWRq\nnlPPqzrPzx1KILAW8uEaeFcg+3N94oX
qI3UDJ7nq29Op1I9IcyzkiigNSZ7lboDGV\nl1TO/xYlVRyMxZyhI3R4a/WyWtvCcCXcnnhe3Z7iVSyBa\
n-----END CERTIFICATE-----\n",
  "certificate_subject": "CN=cckm.thalessecurity.com,OU=CipherTrust,O=Thale
s,L=San Jose,ST=California,C=US",
  "hostname": "xyz-org.my.salesforce.com"
}

```

Example Request 3

```

ksctl connectionmgmt salesforce create --name "salesforce-3" --products "cckm" --
cloudname "Salesforce Sandbox Cloud" --clientid 123456 --username "xyz@gmail.co
m" --use-certificate yes --audience "https://test.salesforce.com" --cert-duratio
n 730

```

Example Response

```

{
  "id": "9be5fde1-4ac8-4ff4-95fd-e4fd3562c7a6",
  "uri": "kylo:kylo:connectionmgmt:connections:salesforce-1-9be5fde1-4ac8-4ff
4-95fd-e4fd3562c7a6",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2025-05-22T10:36:54.155739522Z",
  "updatedAt": "2025-05-22T10:36:54.153944745Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "salesforce-1",
  "products": [
    "cckm"
  ],
  "meta": null,
  "cloud_name": "Salesforce Sandbox Cloud",
  "client_id": "3bf0dbe6-a2c7-431d-9a6f-4843b74c7e12",
  "username": "xyz@abc.com",
}

```

```
{
  "certificate": "-----BEGIN CERTIFICATE----- ... -----END CERTIFICATE-----\n",
  "audience": "https://test.salesforce.com",
  "certificate_subject": "CN=cckm.example.com,OU=CipherTrust,O=example,L=San Jose,ST=California,C=US"
}
```

Using external certificate

Note

The external certificate cannot be used with `use_certificate` and `client_secret` parameters.

To create an Salesforce connection using external certificate generated from custom CSR signed by any internal or external CA:

1. Generate a new Certificate Signing Request (CSR). The Salesforce connections support RSA 1024, 2048, and 4096 bit keys for creating CSRs.

Syntax

```
ksctl connectionmgmt connections csr --cn <common-name> --csr-outfile <filename>
```

Example

```
ksctl connectionmgmt connections csr --cn "test" --csr-outfile "Salesforce.pem"
```

Response

```
{
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIHJMHECAQAwDzENMAsgA1UEAxMEDGVzdDBZMBMBGByqGSM49AgEGCCqGSM49AwEH\nnA0IABAjn2IrYo9SusCMXGwz0oEDcYCpb2U9F15BF3Y9myZQk40cNMjZ8XMwYYLOR\nnyDOMyg8N0XTGy3kg5wqnnSm5namgADAKBggqhkJOPQQDAgNIADBFAiBa7LjFF2D9\nniPKhU89pjndsgG9cx2M9JSC6ZzWo15Py0QIhAJUHgBeYKttzcpo87Ej3eQHRb8MT\nnJRVOqvM3mdbA/ToQ\n-----END CERTIFICATE REQUEST-----\n"
}
```

This CSR can only be used for one connection in the native domain. Also, this CSR can't be reused in other domains.

2. Sign the CSR with any local or external CA. It will generate an external certificate.

3. Upload the generated certificate to the Salesforce portal.
4. Create the Salesforce connection using the external certificate generated above.

Example

```
ksctl connectionmgmt salesforce create --json-file certificatesalesforcecreation.json --name "testsalesforce"
```

Response

```
{
  "id": "b1b23132-507e-4807-aef9-9af2a6c5c2fe",
  "uri": "kylo:kylo:connectionmgmt:connections:testsalesforce-b1b23132-507e-4807-aef9-9af2a6c5c2fe",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-08-23T07:18:33.757838337Z",
  "updatedAt": "2022-08-23T07:18:33.756451354Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "testsalesforce",
  "products": [
    "cckm"
  ],
  "cloud_name": "Salesforce Sandbox Cloud",
  "client_id": "3bf0dbe6-a2c7-431d-9a6f-4843b74c7e12",
  "username": "xyz@abc.com",
  "certificate": "-----BEGIN CERTIFICATE-----\nMIIFUjCCAzqgAwIBAgIQKPONHN6+j5mJCtT+MeR+BDANBgkqhkiG9w0BAQsFADBa\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCVFgx\nDzANBgNVBAcTBkF1c3RpbjEPMA0G...n7O9K/+bXkYc\nnOLvHfHcKHY2If84Y0ik8gtvM4MOCsooFQqzAP\nDMSdRgY8mIMcJYTzZruKqMt4Lh9\nnvOjrBGHmKstG7gIA74frT0ZelHfeZQ==\n-----END CERTIFICATE-----\n",
  "certificate_subject": "CN=kylo.com,OU=RnD,O=Thales Group,L=Belcamp,ST=MD,C=US"
}
```

JSON File

```
{
  "products": [
    "cckm"
  ],
}
```

```

    "cloud_name": "Salesforce Sandbox Cloud",
    "client_id": "3bf0dbe6-a2c7-431d-9a6f-4843b74c7e12",
    "username": "xyz@abc.com",
    "certificate": "-----BEGIN CERTIFICATE-----\nMIIFUjCCAzqgAwIBAgIQKPONHN
6+j5mJCtT+MeR+BDANBgkqhkiG9w0BAQsFADBBA\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCFgxDz
ANBgNVBAcTBkF1c3RDQEBECwUAA4ICAQALSJCy3AqJCmFb\n6m/b6Dj0kJqqUP/xttYGgN....04szy
ln7xj5XIW7SmcT5TvQn7O9K/+bXkYc\nOLvHFHcKHY2If84Y0ik8gtvM4MOCsooFQqzAPDMSdRgY8m
IMcJYTzZruKqMt4Lh9\nvOjrbGHmKstG7gIA74frT0ZelHfeZQ==\n-----END CERTIFICAT
E-----\n",
    "cert_duration": 730
  }

```

The CipherTrust Manager allows you to modify the external certificate in the existing connection. Any unused certificate will be automatically deleted after 24 hours.

Using Client Credential (My Domain)

It is domain name based authentication and uses "domain_name" to map [My Domain](#) in Salesforce login and application URLs.

The client credentials don't require "username" and "password" for authentication. For details, refer to [Salesforce documentation](#).

Following fields are used for authentication:

- client_id - application ID of the Salesforce application
- client_secret - consumer secret for the Salesforce application
- domain_name - represents [My Domain](#) used in the Salesforce account.

To create a Salesforce connection, run:

Syntax

```

ksctl connectionmgmt salesforce create --name <connection-name> --products <produ
cts-names> --clientid <salesforce-client-id> --secret <salesforce-client-secret>
--cloudname <salesforce-cloud-name> --domain-name <salesforce-org-domain-name> --
salesforce-hostname <salesforce-hostname>

```

Note

The domain-name parameter cannot be specified with the salesforce-hostname parameter.

Example Request 1

```

ksctl connectionmgmt salesforce create -n salesforce-1 -c "Salesforce Cloud" --do

```

```
main-name "thaxxxxx--xxxfdc" -d "3MVG9c1ghSpUbLl_2HbT5XAlPgXdC.voJEM.D4jfqQipoLHa.8rtSEO9Zw6qX0MhVdK3V7LrASTjEmBIpQwkv" -t "8177EE1E5967F39381DED2D41B363910B865AD881587AA4DF62C5E813988322A"
```

Example Response

```
{
  "id": "e9b10913-400c-40c3-9f78-e5a4277d5724",
  "uri": "kylo:kylo:connectionmgmt:connections:salesforce-1-e9b10913-400c-40c3-9f78-e5a4277d5724",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-06-13T11:17:25.000529654Z",
  "updatedAt": "2023-06-13T11:17:24.999224701Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "salesforce-1",
  "cloud_name": "Salesforce Cloud",
  "client_id": "3MVG9c1ghSpUbLl_2HbT5XAlPgXdC.voJEM.D4jfqQipoLHa.8rtSEO9Zw6qX0MhVdK3V7LrASTjEmBIpQwkv",
  "domain_name": "thaxxxxx--xxxfdc"
}
```

Example Request 2

```
ksctl connectionmgmt salesforce create --name conn1 --products cckm --salesforce-hostname "thaxxxxx--xxxfdc.sandbox.my.salesforce.com" --clientid "3MVG9c1ghSpUbLl_2HbT5XAlPgXdC.voJEM.D4jfqQipoLHa.8rtSEO9Zw6qX0MhVdK3V7LrASTjEmBIpQwkv" --secret "8177EE1E5967F39381DED2D41B363910B865AD881587AA4DF62C5E813988322A" --cloudname "Salesforce Sandbox Cloud"
```

Example Response

```
{
  "id": "3c0b1b5a-457c-4fc6-b10e-1fec20c29bd1",
  "uri": "kylo:kylo:connectionmgmt:connections:conn1-3c0b1b5a-457c-4fc6-b10e-1fec20c29bd1",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2024-12-27T11:38:15.018068828Z",
  "updatedAt": "2024-12-27T11:38:15.014482643Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "conn1",
  "products": [
    "cckm"
  ],
  "cloud_name": "Salesforce Sandbox Cloud",
  "client_id": "3MVG9c1ghSpUbLl_2HbT5XAlPgXdC.voJEM.D4jfqQipoLHa.8rtSEO9Zw6qX0MhVdK3V7LrASTjEmBIpQwkv",
  "hostname": "thaxxxxx--xxxfdc.sandbox.my.salesforce.com"
}
```

Creating a Salesforce Connection with mTLS enabled

CipherTrust Manager provides an option to secure specific API calls from CipherTrust Manager to Salesforce with an mTLS certificate. You can provide a TLS client certificate and a private key while creating a Salesforce connection to enable mTLS certificate authentication. If the private key is protected with a passphrase, then there is the option to also provide this passphrase. For the prerequisite steps to take within Salesforce to set up the use of this Salesforce mTLS option, refer to [Setting Up a Mutual Authentication Certificate](#) and associated Salesforce Help topics.

We support pkcs8 format for the certificate. If the certificate is protected by a passphrase, we support pkcs1 format for the private key. If the certificate is not protected by a passphrase, we support pkcs8 or pkcs1 formats for the private key.

To create a Salesforce connection with the verification of a TLS client-side certificate, run:

Syntax

```
ksctl connectionmgmt salesforce create --name sfdc1 --clientid "clientid" --secret "salesforce-client-secret" --username "email-of-user" --conn-password "connection-passwd" --cloudname "name-of-salesforce-cloud" --tls-client-cert-with-private-key-file "pointer-to-file-containing-mtls-client-cert" --tls-client-private-key-password "password-of-private-key" --enable-mutual-tls "boolean"
```

Example Request

```
ksctl connectionmgmt salesforce create --name sfdc1 --clientid 3MVG9p1Q1BCe9GmC3YCMWFj2f.NBP_SD5K24usl_MOX3uRIzzi9IH2b77VwN0Ov4XpIi.Glt2CoHSURy4b1AK --secret DF98FB4C8B75C8934E699114E3832C8C57AF45374CF1450CFF88494ADED65A --username xyz@gmail.com --conn-password Thales2023# --cloudname 'Salesforce Cloud' --tls-client-cert-with-private-key-file /home/user1/Downloads/letsencrypt_fullcert_01182023_with_passphrase.pem --tls-client-private-key-password Ssl12345# --enable-mutual-tls true
```

Example Response

```
{
  "id": "e561bc25-63de-461d-aa94-d76baa595dc4",
  "uri": "kylo:kylo:connectionmgmt:connections:sfdc1-e561bc25-63de-461d-aa94-d76baa595dc4",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-03-14T20:45:37.431827324Z",
  "updatedAt": "2023-03-14T20:45:37.427742423Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "sfdc1",
  "cloud_name": "Salesforce Cloud",
  "client_id": "3MVG9p1Q1BCe9GmC3YCMWFj2f.NBP_SD5K24usl_MOX3uRIzzi9IH2b77VwN0Ov4XpIi.Glt2CoHSURy4b1AK",
  "username": "xyz@gmail.com",
  "enable_mutual_tls": true,
  "tls_client_certificate": "-----BEGIN CERTIFICATE-----\nMIIEZTCCA0gAwIBAgISA0mUYC0i/8xCeuuJ5vkaVzeaMA0GCSqGSIb3DQEBCwUA\nnMDIxGZAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQswCQYDVQmWIGhhLp/EGbVY1AiEA\npzfjoX+LN7KUIQnwC61GnZAGXSPMdorj4\nnqIHm+b2XGcZ/1m9kTLKl3GeVqLnsE9+Uj\nkz10VEsL2urkKpDTPnBlziO8E7DxIX8\n\nnoAl/0Yh2SxOYOn67
```

```
Yv0Joofv+Tl5ilVnKVky7ndRehjmlevs5r6cHFib0if4e/WQ\nP03rukbsgCGFDw43Bbs0fBhC1k80oPG
NmA==\n-----END CERTIFICATE-----\n"
}
```

Getting Details of a Salesforce Connection

To get details of a Salesforce connection, run:

Syntax

```
ksctl connectionmgmt salesforce get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt salesforce get --id 5818ec3d-c5e9-4039-b911-efd45cb08f88
```

Example Response

```
{
  "id": "5818ec3d-c5e9-4039-b911-efd45cb08f88",
  "uri": "kylo:kylo:connectionmgmt:connections:salesforce-1-5818ec3d-c5e9-4039-b911-efd45cb08f88",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-08-10T10:21:29.967822Z",
  "updatedAt": "2021-08-10T10:21:29.966606Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "salesforce-1",
  "products": [
    "cckm"
  ],
  "cloud_name": "Salesforce Sandbox Cloud",
  "client_id": "3bf0dbe6-a2c7-431d-9a6f-4843b74c7e12",
  "username": "xyz@gmail.com"
}
```

Updating a Salesforce Connection

To update a Salesforce connection, run:

Syntax

```
ksctl connectionmgmt salesforce modify --id <Connection-Name/ID> --products <Products-Names> --clientid <Salesforce-Key-ID> --secret <Salesforce-Client-Secret> --meta <Key:Values> --salesforce-hostname <salesforce-hostname>
```

Example Request 1


```
ksctl connectionmgmt salesforce modify --id 5818ec3d-c5e9-4039-b911-efd45cb08f88
--username abcd@gmail.com
```

Example Response

```
{
  "id": "5818ec3d-c5e9-4039-b911-efd45cb08f88",
  "uri": "kylo:kylo:connectionmgmt:connections:salesforce-1-5818ec3d-c5e9-4039-b911-efd45cb08f88",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-08-10T10:21:29.967822Z",
  "updatedAt": "2021-08-10T10:25:06.228123698Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "salesforce-1",
  "products": [
    "cckm"
  ],
  "cloud_name": "Salesforce Sandbox Cloud",
  "client_id": "3bf0dbe6-a2c7-431d-9a6f-4843b74c7e12",
  "username": "abcd@gmail.com"
}
```

Example Request 2

```
ksctl connectionmgmt salesforce modify --id conn1 --salesforce-hostname ""
```

Example Response

```
{
  "id": "3c0b1b5a-457c-4fc6-b10e-1fec20c29bd1",
  "uri": "kylo:kylo:connectionmgmt:connections:conn1-3c0b1b5a-457c-4fc6-b10e-1fec20c29bd1",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2024-12-27T11:38:15.018069Z",
  "updatedAt": "2024-12-27T11:39:50.445207352Z",
  "service": "salesforce",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "conn1",
  "products": [
    "cckm"
  ],
  "cloud_name": "Salesforce Sandbox Cloud",
  "client_id": "3MVG9c1ghSpUbLl_2HbT5XAlPgXdC.voJEM.D4jfqQipoLHa.8rtSEO9Zw6qX0MhVdK3V7LrASTjEmBIpQwkv",
  "hostname": ""
}
```

Deleting a Salesforce Connection

To delete a Salesforce connection, run:

Syntax

```
ksctl connectionmgmt salesforce delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt salesforce delete --id c4a11fc1-d226-4f19-8d03-5911198e89e5
```

Example Response

There will be no response if Salesforce connection is deleted successfully.

Getting List of Salesforce Connections

To list all the Salesforce connections, run:

Syntax

```
ksctl connectionmgmt salesforce list
```

Example Request

```
ksctl connectionmgmt salesforce list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "5818ec3d-c5e9-4039-b911-efd45cb08f88",
      "uri": "kylo:kylo:connectionmgmt:connections:salesforce-1-5818ec3d-c5e9-4039-b911-efd45cb08f88",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2021-08-10T10:21:29.967822Z",
      "updatedAt": "2021-08-10T10:21:29.966606Z",
      "service": "salesforce",
      "category": "cloud",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "salesforce-1",
      "products": [
        "cckm"
      ],
      "cloud_name": "Salesforce Sandbox Cloud"
    }
  ]
}
```

```
]
}
```

Testing an Existing Salesforce Connection

To test an existing Salesforce connection, run:

Syntax

```
ksctl connectionmgmt salesforce test --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt salesforce test --id 5087c384-43aa-4750-951e-3a0f2b60d43b
```

Example Response

```
{
  "connection_ok": true,
}
```

Testing Parameters for a Salesforce Connection

To test parameters for a Salesforce connection, run:

Syntax

```
ksctl connectionmgmt salesforce test --id <Connection-Name/ID> --clientid <Salesforce-Key-ID> --secret <Salesforce-Client-Secret>
```

Example Request

```
ksctl connectionmgmt salesforce test --clientid 3bf0dbe6-a2c7-431d-9a6f-4843b74c7e12 --secret BC0556E7A0B4C96E218EF91370C5B --username salesforce-1 --conn-password 123456 --cloudname "Salesforce Sandbox Cloud"
```

Example Response

```
{
  "connection_ok": true,
}
```

Luna Network HSM

Luna Network HSM connections are required for two use cases:

- To use a Luna Network HSM partition as a key source for [CipherTrust Cloud Key Manager \(CCKM\)](#). Luna keys

managed by CCKM are usually part of broader integrations with cloud providers.

- To use a Luna Network HSM partition for an [HSM-anchored domain](#). All keys and secrets within an HSM-anchored domain are wrapped and unwrapped by the HSM itself.

Warning

Thales strongly discourages using a Luna HSM partition containing the CipherTrust Manager [root of trust keys](#) for CCKM or HSM-anchored domains. Do not add a connection to a Luna HSM root-of-trust partition with Connection Manager.

Supported Luna Network HSM Configurations

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.

Managing Luna Network HSM Client

You can register the CipherTrust Manager as client in the following modes :

- Network Trust Link Service (NTLS)
- Secure Trusted Channel (STC)

Using NTLS Mode

To register the CipherTrust Manager as client in NTLS mode:

1. Download the client certificate on the CipherTrust Manager.
2. Register the client certificate on HSM server.
3. Assign the partition to the client.

Using STC Mode

To register the CipherTrust Manager as client in STC mode for CCKM:

1. Download the client identity on the CipherTrust Manager.
2. Register the CipherTrust Manager as an additional client on the first client where partition is registered.

For more information, refer to https://thalesdocs.com/gphsm/luna/7/docs/network/Content/admin_partition/connections/stc/multi_client.htm

Note

The client identity is returned in the base64 format in the APIs, make sure to convert it to normal text format before saving to the file.

Managing LUNA HSM Connections using GUI

For both current purposes, you must configure **Luna Network connection** and **HSM server**.

Warning

It is mandatory to create one or more HSM Servers before creating an HSM Connection.

Adding an Internal Connection (Server)

It allows you to add the HSM Server and download the Luna Client certificate.

Note

Currently, you can add only HSM Servers.

To configure an HSM server:

Click **Add HSM Server** in the **INTERNAL CONNECTIONS** section to add an HSM Server.

- **HSM Hostname/IP** - Provide the hostname/IP of the server.
- **HSM Certificate** - Upload the HSM certificate.
- **HSM Description** - Provide the HSM description.
- **HSM Products** - Select the check boxes in the Products list to select a product associated with the HSM server. Select **Cloud Key Manager** to use this connection as a key source for CCKM. Do not select any products to use this connection for HSM-anchored domains.

Click **Download Luna Client Cert** in the **INTERNAL CONNECTIONS** section to download the Luna client certificate.

Note

- Currently, the only product available in the GUI for HSM Server is Cloud Key Manager. To use this connection for HSM-anchored domains, don't select any check boxes.
- Luna Network HSMs can only be added at the CipherTrust Manager **root domain**.

Click **Create** to add the HSM Server. The new server is now listed in the **INTERNAL CONNECTIONS** list.

Configuring the Luna Network HSM Connection

To configure the Luna Network HSM connection:

- **Partition Server Hostname/IP** - Select the hostname/IP of the server from the drop-down list.
- **Partition Label** - Label of the HSM partition.
- **Partition Serial No** - Serial number of the HSM Partition.
- **Add Partition** - Click this button to add the multiple partitions.

- **Partition Password** - Crypto Officer (CO) password of the HSM partition(s).

Click the **Test Credentials** button to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.

Click **Next** to move to the **Add Products** screen of the Add Connection wizard.

Note

The two products available are **HSM-anchored domain** and **Cloud Key Manager**.

Managing LUNA HSM Connections using ksctl

Luna network HSM management is divided into:

- [Luna Network HSM Servers](#)
- [Luna Network HSM Connections](#)

Luna Network HSM Servers

The following operations can be performed:

- Add/delete/get a Luna network HSM server
- List all Luna network HSM servers
- Enable/Disable STC on Luna Servers
- Get Luna client details such as certificate and hostname

The Luna servers are used to create a connection of type Luna network HSM.

Adding a Luna Server

Note

To use HSM partition in STC mode, make sure to enable STC for the HSM Server.

To add a Luna Server, run:

Syntax

```
ksctl connectionmgmt luna-hsm servers add --hostname <Hostname/IP> --hsm-cert-file <HSM-Certificate>
```

This command requires a hostname or IP of the server and a valid certificate.

Example Request

```
ksctl connectionmgmt luna-hsm servers add --hostname host --hsm-cert-file ~/server.pem
```

Example Response

```
{
  "hostname": "host",
  "hsm_certificate": "-----BEGIN CERTIFICATE-----\nMIIDNzCCAh+gAwIBAgIBADANBgkqhkiG9w0BAQsFAADBMQswCQYDVQQGEwJDQTEQ\nMA4GA1UECAwHT250YXJpbzEPMA0GA1UEBwwGT3R0YXdhMRYwFAYDVQQKDA1DaHJ5\n\nc2FsaXMTSVRTRUwEwYDVQQDDAwxMC4xNjQuNTYuODYwHhcNMjAwODIwMDg1OTQ0\n\nWhcNMzAwODIyMDg1OTQ0WjBfMQswCQYDVQQGEwJDQTEQMA4GA1U..uODYwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB\n\nAQhADnOAJb7jIB3GSBdpBH7VJhQ5VU5sNHqg4ZiNi1vLZPPed\n\nn9HdJPTtbN4019SgY2kSwg1nky8jZY8uA9Qh05izWz3S1p9ZY9kpgRaBCTGCAF/C2\n\nnobI+LA8a7DlU9PQ=\n-----END CERTIFICATE-----\n",
  "id": "83a24275-65ff-42cf-9e22-edd1b7f0c4f3",
  "uri": "kylo:kylo:connectionmgmt:hsm-servers:host-83a24275-65ff-42cf-9e22-edd1b7f0c4f3",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-04T09:25:27.163022185Z",
  "service": "luna network"
}
```

Getting Details of Luna Server

To get details of a Luna Server already registered with the Connection Manager, run:

Syntax

```
ksctl connectionmgmt luna-hsm servers get --id <Hostname/Id>
```

This command requires an identifier that can either be ID or hostname of the server.

Example Request

```
ksctl connectionmgmt luna-hsm servers get --id host
```

Example Response

```
{
  "hostname": "host",
  "hsm_certificate": "-----BEGIN CERTIFICATE-----\nMIIDNzCCAh+gAwIBAgIBADANBgkqhkiG9w0BAQsFAADBMQswCQYDVQQGEwJDQTEQ\n\nMA4GA1UECAwHT250YXJpb/..hADnOAJb7jIB3GSBdpBH7VJhQ5VU5sNHqg4ZiNi1vLZPPed\n\nn9HdJPTtbN4019SgY2kSwg1nky8jZY8uA9Qh05izWz3S1p9ZY9kpgRaBCTGCAF/C2\n\nnobI+LA8a7DlU9PQ=\n-----END CERTIFICATE-----\n",
  "id": "83a24275-65ff-42cf-9e22-edd1b7f0c4f3",
  "uri": "kylo:kylo:connectionmgmt:hsm-servers:host-83a24275-65ff-42cf-9e22-edd1b7f0c4f3",
  "account": "kylo:kylo:admin:accounts:kylo",
}
```



```
{
  "createdAt": "2020-12-04T09:25:27.163022Z",
  "service": "luna network"
}
```

Deleting a Luna Server

To delete a Luna Server, run:

Syntax

```
ksctl connectionmgmt luna-hsm servers delete --id <Hostname/Id>
```

This command requires an identifier that can either be ID or hostname of the server.

There will be no response if server is deleted successfully.

Getting List of Luna Servers

To list all the Luna Servers already registered with the Connection Manager, run:

Syntax

```
ksctl connectionmgmt luna-hsm servers list
```

Example Request

```
ksctl connectionmgmt luna-hsm servers list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "hostname": "host",
      "hsm_certificate": "-----BEGIN CERTIFICATE-----\nMIIDNzCCAh+gAwIBAgIB  
ADANBgkqhkiG9w0BAQsFADBfMQswCQYDVQQGEwJDQTEQ\nnMA4GA1UECAwHT250YXJpbzEPMA0GA1UEBww  
GT3R0YXdhMRYwFAYDVQQKDA1DaHJ5\nc2FsaXMtSVRTMRUwE...hADnOAJb7jIB3GSBdpBH7VJhQ5VU5  
sNHgg4ZiNi1vLZPPed\n9HdJPTtbN4019SgY2kSwg1nky8jZY8uA9Qh05izWz3S1p9ZY9kpgRaBCTGCA  
F/C2\nnobI+LA8a7DlU9PQ=\n-----END CERTIFICATE-----\n",
      "id": "83a24275-65ff-42cf-9e22-edd1b7f0c4f3",
      "uri": "kylo:kylo:connectionmgmt:hsm-servers:host-83a24275-65ff-42c  
f-9e22-edd1b7f0c4f3",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2020-12-04T09:25:27.163022Z",
      "service": "luna network"
    }
  ]
}
```

Enabling STC on Luna Servers

To enable STC channel of HSM Connection run:

Syntax

```
ksctl connectionmgmt luna-hsm servers enable-stc --id <Connection-name/ID>
```

Example Request

```
ksctl connectionmgmt luna-hsm servers enable-stc --id 7ba5172a-39e6-47bb-a115-2f97b6347b76
```

Example Response

```
{
  "hostname": "10.164.10.37",
  "hsm_certificate": "-----BEGIN CERTIFICATE-----\nMIIDKzCCAhOgAwIBAgIBADANBgkqhkiG9w0BAQsFADBMQswCQYDVQQGEwJDQTEQ\nMA4GA1UECAwHT250YXJpbzEPMA0GA1UEBwwGT3R0YXdhMRYwFAYDVQQKDA1DaHJ5\nJ5\nc2Fsa....+x3JCumneEu7WR6A3mYZiCs0OSty99M\n\nBISITYaYqrB0yWLR9EUDQ4CfpmWX2lHqirTMlMXkZMv9WYRC5CFHltgZqyODnob+\n\nncUE72FwxiVjrm3foFtFSraxGttfNBqaPiBKr7W5b1CFVaBhIcG/q/30KxQ7vA8Vm\n\nnZAJEQhkdE0e1kwtXYk2goa2cH4UY7azCWwlcoSH8e6IKXh4H/AZZWlrHn0+HD4I=\n\n-----END CERTIFICATE-----",
  "products": [
    "cckm"
  ],
  "meta": {
    "color": "blue"
  },
  "id": "7ba5172a-39e6-47bb-a115-2f97b6347b76",
  "uri": "kylo:kylo:connectionmgmt:hsm-servers:10-164-10-37-7ba5172a-39e6-47bb-a115-2f97b6347b76",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-09-19T07:21:51.191164Z",
  "service": "luna network",
  "channel": "STC"
}
```

Disabling STC on Luna Servers

To disable STC channel of HSM Connection run:

Syntax

```
ksctl connectionmgmt luna-hsm servers disable-stc --id <Connection-name/ID>
```

Example Request

```
ksctl connectionmgmt luna-hsm servers disable-stc --id 7ba5172a-39e6-47bb-a115-2f97b6347b76
```

Example Response

```
{
  "hostname": "10.164.10.37",
  "hsm_certificate": "-----BEGIN CERTIFICATE-----\nMIIDKzCCAhOgAwIBAgIBADANBgkq
hkiG9w0BAQsFADBMQswCQYDVQQGEwJDQTEQ\nMA4GA1UECAwHT250YXJpbzEPMA0GA1UEBwwGT3R0YXd
hMRYwFAYDVQQKDA1DaHJ5\nc2Fs...y99M\nBISITYaYqrB0yWLR9EUDQ4CfpmWX21HqirTMlMXkZMv9W
YRC5CFHltgZqyODnob+\ncUE72FwxiVjrm3foFtFSraxGttfNBqaPiBKr7W5b1CFVaBhIcG/q/30KxQ7v
A8Vm\nnZAJEghkdE0e1kwtXYk2goa2cH4UY7azCWwlcoSH8e6IKXh4H/AZZWlrHn0+HD4I=\n-----END
CERTIFICATE-----",
  "products": [
    "cckm"
  ],
  "meta": {
    "color": "blue"
  },
  "id": "7ba5172a-39e6-47bb-a115-2f97b6347b76",
  "uri": "kylo:kylo:connectionmgmt:hsm-servers:10-164-10-37-7ba5172a-39e6-47bb-
a115-2f97b6347b76",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-09-19T07:21:51.191164Z",
  "service": "luna network",
  "channel": "NTLS"
}
```

Getting Details of a Luna Client

To get details of a Luna Client registered with a Luna Server, run:

Syntax

```
ksctl connectionmgmt luna-hsm servers client-get
```

Example Request

```
ksctl connectionmgmt luna-hsm servers client-get
```

Example Response

```
{
  "id": "5fc757bd-8e95-4352-8d1c-4bc861d252d9",
  "uri": "kylo:kylo:doorway:Certificate:5fc757bd-8e95-4352-8d1c-4bc861d252d9",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2020-12-03T12:22:46.061088Z",
  "updatedAt": "2020-12-03T12:22:46.056696Z",
  "hostname": "cckm-client-51437b79-4f10-490e-9769-3d5b0526af46",
  "certificate": "-----BEGIN CERTIFICATE-----\nMIIDezCCAmOgAwIBAgIBADANBgkqhkiG
9w0BAQsFADCBGgDELMakGA1UEBhMCQ0Ex\nEDAObgNVBAgMB09udGFyaW8xDzANBgNVBACMBk90dGF3YTE
TMBEGA1UECgwKTXkg\nY29tc...1TrKoChr5nVBugU3BrKZgwm9yrMntuzCqmIVl2dstlbL9R+LSoCns
5\na/PreKkP4DbxqxxgeE7RTqtv+qhjrKyMQVMDsHfCDclJe+NBHkwVrfIdXJrJVuuh\nnxZC/isR370ye
t+J4HM57xsNswI3/YG4l4nXl5jt9dQ==\n-----END CERTIFICATE-----\n"
}
```

Luna Network HSM Connections

The following operations can be performed:

- Create/Get/Update/Delete a Luna Network HSM connection
- List all Luna Network HSM connections
- Test an existing Luna Network HSM connection
- Test the newly created connection

A Luna Network HSM connection can be an HA or non-HA.

HA stands for High Availability, that means there will be more than one partition to ensure availability and load balancing.

In an HA connection, there are multiple partitions of one or more HSM Servers. Whereas, in a non-HA connection there is a single partition of an HSM Server.

Creating a Luna Connection

To create a connection of Luna Network HSM type, run:

Syntax

```
ksctl connectionmgmt luna-hsm connections create --name <Connection-Name> --conn-
password <Partition-Password> --partitions-json-file <xxx.json> --ha-enable <yes/
no> --key-mgmt-mode <clone/key-export> --products <"cckm" or "hsm_anchored_domai
n">
```

Here,

- `name` - Name of the connection.
- `partitions-json-file` - Partition file of JSON type.
- `conn-password` - Password of the Luna partitions.
- `ha-enable` - Signifies if it is High Availability(HA) group or not. The default value is `no`.
- `key-mgmt-mode` - The mode defined in `key_mgmt_mode` is only valid for **CCKM** product. This mode is used for handling private keys. The `key-mgmt-mode` parameter is applicable only when `ha-enable` is set to `yes`. The individual partitions can be configured by using one of the supported modes.
 - `key-export` - CCKM supports the replication of private asymmetric keys within the partition's HA group.
 - `clone` - All keys/objects in CCKM are replicated automatically within the partition's HA group.

Note

If a wrong mode is selected, CCKM RSA key creation on Luna will be affected. Refer to [note](#) for error details.

It is mandatory that all the partitions should be in the same mode. The default value is FALSE. If no value or empty string ("") is provided, then a connection will be created without any mode. For creating partitions and handling keys in a specific mode in HSM, click [here](#).

- `products` - Product type, either `cckm` or `hsm_anchored_domain`.

To create a connection with multiple partitions (with an HA group), the HA flag should be specified as TRUE. The format of the JSON file to create a connection:

```
[
  { "hostname": "xx.xxx.xx.xx", "partition_label": "sample-label1", "serial_number": "xxxxxxx" },
  { "hostname": "xx.xxx.xx.xx", "partition_label": "sample-label2", "serial_number": "xxxxxxx" }
]
```

Note

- If a Luna HSM partition and the associated Luna HSM connection are deleted from the CipherTrust Manager, the Luna source key link from Azure Keys page will not work. To work around this issue:
 - Add the Luna HSM connection again.
 - Add the Luna HSM partitions in the *same order* in which they were added before deletion.
- After deleting and re-adding a partition to CCKM, a refresh should be performed on the CCKM partition. Refer to [Refreshing Specific Partitions](#) for details.

Example Request 1

```
ksctl connectionmgmt luna-hsm connections create --name demo1 --conn-password pascode --partitions-json-file partitions.json --ha-enable yes
```

Example Response 1

```
{
  "id": "c8c1cd6b-1f37-405c-9e12-de2f6bec2c36",
  "uri": "kylo:kylo:connectionmgmt:connections:demo1-c8c1cd6b-1f37-405c-9e12-de2f6bec2c36",
}
```

```

"account": "kylo:kylo:admin:accounts:kylo",
"createdAt": "2020-12-04T09:30:20.592526537Z",
"updatedAt": "2020-12-04T09:30:20.591321554Z",
"service": "luna network",
"category": "hsm",
"last_connection_ok": null,
"last_connection_at": "0001-01-01T00:00:00Z",
"name": "demo1",
"partitions": [
  {
    "hostname": "xx.xxx.xx.xx",
    "serial_number": "14",
    "partition_label": "sample-label"
  },
  {
    "hostname": "xx.xxx.xx.xx",
    "serial_number": "12",
    "partition_label": "sample-label"
  }
],
"is_ha_enabled": true
}

```

Example Request 2

```

ksctl connectionmgmt luna-hsm connections create --name demo1 --conn-password pas
scode --partitions-json-file partitions.json --ha-enable yes --key-mgmt-mode clon
e

```

Example Response 2

```

{
  "id": "c8c1cd6b-1f37-405c-9e12-de2f6bec2c36",
  "uri": "kylo:kylo:connectionmgmt:connections:demo1-c8c1cd6b-1f37-405c-9e12-de
2f6bec2c36",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-04T09:30:20.592526537Z",
  "updatedAt": "2020-12-04T09:30:20.591321554Z",
  "service": "luna network",
  "category": "hsm",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "demo1",
  "partitions": [
    {
      "hostname": "xx.xxx.xx.xx",
      "serial_number": "14",
      "partition_label": "sample-label"
    },
    {
      "hostname": "xx.xxx.xx.xx",
      "serial_number": "12",
      "partition_label": "sample-label"
    }
  ],
  "is_ha_enabled": true,
}

```

```
{
  "key_mgmt_mode": "clone"
}
```

Getting Details of a Luna Connection

To get details of a Luna Network connection, run:

Syntax

```
ksctl connectionmgmt luna-hsm connections get --id <Id/Connection-Name>
```

This command requires a connection identifier that can be either ID or name of the connection.

Example Request

```
ksctl connectionmgmt luna-hsm connections get --id demo1
```

Example Response

```
{
  "id": "c8c1cd6b-1f37-405c-9e12-de2f6bec2c36",
  "uri": "kylo:kylo:connectionmgmt:connections:demo1-c8c1cd6b-1f37-405c-9e12-de2f6bec2c36",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-04T09:30:20.592527Z",
  "updatedAt": "2020-12-04T09:30:20.591322Z",
  "service": "luna network",
  "category": "hsm",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "demo1",
  "partitions": [
    {
      "id": "39c7775c-a72c-4b31-9745-d1e9adbf8946",
      "uri": "kylo:kylo:connectionmgmt:luna-network-partition:demo1-39c7775c-a72c-4b31-9745-d1e9adbf8946",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2020-12-04T09:30:20.597013Z",
      "hostname": "xx.xxx.xx.xx",
      "serial_number": "14",
      "partition_label": "sample-label"
    },
    {
      "id": "e3b7914d-3a88-40de-9385-649c5f019e3f",
      "uri": "kylo:kylo:connectionmgmt:luna-network-partition:demo1-e3b7914d-3a88-40de-9385-649c5f019e3f",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2020-12-04T09:30:20.598614Z",
      "hostname": "xx.xxx.xx.xx",
      "serial_number": "12",
      "partition_label": "sample-label"
    }
  ],
  "is_ha_enabled": true,
}
```

```

    "key_mgmt_mode": "clone",
    "max_session_count": 0,
    "session_count": 0,
    "max_rw_session_count": 0,
    "rw_session_count": 0,
    "max_pin_len": 0,
    "min_pin_len": 0,
    "total_public_memory": 0,
    "free_public_memory": 0,
    "total_private_memory": 0,
    "free_private_memory": 0,
    "operation_status": "",
    "operation_error": ""
  }

```

Updating a Luna Connection

To update a Luna Network connection, run:

Syntax

```

ksctl connectionmgmt luna-hsm connections update --id <Id/Name> --conn-password
<New-Password> --key-mgmt-mode <New-Mode>

```

This command requires:

- A connection identifier that can either be ID or name of the connection
- One or more parameters to update

The Luna Connection Update supports updating the password and other meta information.

Note

This command does not support updating a partition information.

Example Request 1

```

ksctl connectionmgmt luna-hsm connections update --id demo1 --conn-password newPa
sscode

```

Example Response 1

```

{
  "id": "c8c1cd6b-1f37-405c-9e12-de2f6bec2c36",
  "uri": "kylo:kylo:connectionmgmt:connections:demo1-c8c1cd6b-1f37-405c-9e12-de
2f6bec2c36",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-04T09:30:20.592526537Z",

```



```

    "updatedAt": "2020-12-04T09:30:20.591321554Z",
    "service": "luna network",
    "category": "hsm",
    "last_connection_ok": null,
    "last_connection_at": "0001-01-01T00:00:00Z",
    "name": "demo1",
    "partitions": [
      {
        "hostname": "xx.xxx.xx.xx",
        "serial_number": "14",
        "partition_label": "sample-label"
      },
      {
        "hostname": "xx.xxx.xx.xx",
        "serial_number": "12",
        "partition_label": "sample-label"
      }
    ],
  },
}

```

Example Request 2

```

ksctl connectionmgmt luna-hsm connections update --id demo1 --key-mgmt-mode key-export

```

Example Response 2

```

{
  "id": "c8c1cd6b-1f37-405c-9e12-de2f6bec2c36",
  "uri": "kylo:kylo:connectionmgmt:connections:demo1-c8c1cd6b-1f37-405c-9e12-de2f6bec2c36",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-04T09:30:20.592526537Z",
  "updatedAt": "2020-12-04T09:30:20.591321554Z",
  "service": "luna network",
  "category": "hsm",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "demo1",
  "partitions": [
    {
      "hostname": "xx.xxx.xx.xx",
      "serial_number": "14",
      "partition_label": "sample-label"
    },
    {
      "hostname": "xx.xxx.xx.xx",
      "serial_number": "12",
      "partition_label": "sample-label"
    }
  ],
  "is_ha_enabled": true,
  "key_mgmt_mode": "key-export"
}

```

Deleting a Luna Connection

To delete a Luna Network connection, run:

Syntax

```
ksctl connectionmgmt luna-hsm connections delete --id <Id/Name>
```

There will be no response if LUNA Network connection is deleted successfully.

Note

Deleting a Luna Network connection does not delete the HA group. The HA group is kept for future references. If the existing HA group is updated by adding a new partition or by removing a partition that is already part of the existing HA group, the HA group will be updated after performing the test connection.

Getting List of Luna Connections

To list all the connections of Luna Network HSM type, run:

Syntax

```
ksctl connectionmgmt luna-hsm connections list
```

Example Request

```
ksctl connectionmgmt luna-hsm connections list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "c8c1cd6b-1f37-405c-9e12-de2f6bec2c36",
      "uri": "kylo:kylo:connectionmgmt:connections:demo1-c8c1cd6b-1f37-405c-9e12-de2f6bec2c36",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2020-12-04T09:30:20.592527Z",
      "updatedAt": "2020-12-04T09:30:20.591322Z",
      "service": "luna network",
      "category": "hsm",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "demo1",
      "partitions": [
        {

```

```

        "id": "39c7775c-a72c-4b31-9745-d1e9adbf8946",
        "uri": "kylo:kylo:connectionmgmt:luna-network-partition:demo
1-39c7775c-a72c-4b31-9745-d1e9adbf8946",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2020-12-04T09:30:20.597013Z",
        "hostname": "xx.xxx.xx.xx",
        "serial_number": "14",
        "partition_label": "sample-label"
    },
    {
        "id": "e3b7914d-3a88-40de-9385-649c5f019e3f",
        "uri": "kylo:kylo:connectionmgmt:luna-network-partition:demo
1-e3b7914d-3a88-40de-9385-649c5f019e3f",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2020-12-04T09:30:20.598614Z",
        "hostname": "xx.xxx.xx.xx",
        "serial_number": "12",
        "partition_label": "sample-label"
    }
],
"is_ha_enabled": true,
"key_mgmt_mode": "key-export"
}
]
}

```

Adding a partition to the Luna Connection

To add a partition to the Luna Connection, run:

Syntax

```
ksctl connectionmgmt luna-hsm connections add-partition --id <Id/Name> --partitio
ns-json-file <xxx.json>
```

A partition can only be added to a connection if HA flag is TRUE.

The format of the JSON file to add a partition:

```
{
  "hostname": "xx.xxx.xx.xx",
  "partition_label": "sample-label2",
  "serial_number": "xxxxxxx"
}
```

Example Request

```
ksctl connectionmgmt luna-hsm connections add-partition --id demo1 --partitions-j
son-file partition.json
```

Example Response

```
{
  "id": "288b05a9-0e08-4b76-be6c-3713b0e10751",
  "uri": "kylo:kylo:connectionmgmt:luna-network-partition:demo1-288b05a9-0e08-4
b76-be6c-3713b0e10751",

```

```
{
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2020-12-05T06:01:27.482393059Z",
  "hostname": "xx.xxx.xx.xx",
  "serial_number": "1429964054509",
  "partition_label": "sample-label"
}
```

Deleting a Partition from the Luna Connection

To delete a partition from the Luna Connection, run:

Syntax

```
ksctl connectionmgmt luna-hsm connections delete-partition --id <Id/Name> --partition-id <Partition-Id>
```

There will be no response if partition is deleted successfully.

Testing an Existing Luna Connection

To test an existing Luna Network connection, run:

Syntax

```
ksctl connectionmgmt luna-hsm connections test --id <Id/Name>
```

This command requires a connection identifier that can either be ID or name of the connection. This command is asynchronous; therefore, it initiates a connection test and gives the status as `in_progress`. You can fetch the actual status by using the `get` command for the same connection.

Example Request

```
ksctl connectionmgmt luna-hsm connections test --id demo1
```

Example Response

```
{
  "id": "b1c8597a-670e-456f-b2e4-a452311e2916",
  "uri": "kylo:kylo:hsm:connections:b1c8597a-670e-456f-b2e4-a452311e2916",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2020-12-04T09:37:17.578573227Z",
  "updatedAt": "2020-12-04T09:37:17.575470994Z",
  "connection_status": "in_progress"
}
```

Testing a New Luna Connection

To test a New Luna Network connection parameters, run:

Syntax

```
ksctl connectionmgmt luna-hsm connections test --conn-password <Partitions-Password> --partitions-json-file <xxx.json> --ha-enable <Yes/No>
```

This command requires a partition file of JSON type and a password of the luna partitions. HA flag is optional, and the default value is FALSE. To test connection parameters with multiple partitions (with an HA group), the HA flag should be specified as TRUE. The format of the JSON file to create a connection:

```
[
  { "hostname": "xx.xxx.xx.xx", "partition_label": "sample-label1", "serial_number": "xxxxxxx" },
  { "hostname": "xx.xxx.xx.xx", "partition_label": "sample-label2", "serial_number": "xxxxxxx" }
]
```

This command is asynchronous; therefore, it initiates a connection test and gives the status as `in_progress`. The `test-status` command can be used to fetch the actual status by using the ID returned with this command.

Example Request

```
ksctl connectionmgmt luna-hsm connections test --conn-password passcode --partitions-json-file partitions.json --ha-enable yes
```

Example Response

```
{
  "id": "00eb8941-a787-4440-a46d-8f658b7f97d3",
  "uri": "kylo:kylo:hsm:connections:00eb8941-a787-4440-a46d-8f658b7f97d3",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2020-11-23T13:27:20.281086901Z",
  "updatedAt": "2020-11-23T13:27:20.277119471Z",
  "connection_status": "in_progress"
}
```

Getting a Test Status

To get the status of the Luna connection parameters test performed earlier, run:

Syntax

```
ksctl connectionmgmt luna-hsm connections test-status --id <Test-Identifier>
```

This command requires a test ID that is returned as a part of the test command.

Example Request

```
ksctl connectionmgmt luna-hsm connections test-status --id 00eb8941-a787-4440-a46d-8f658b7f97d3
```

Example Response

```
{
  "id": "00eb8941-a787-4440-a46d-8f658b7f97d3",
  "uri": "kylo:kylo:hsm:connections:de7b1255-9ded-4222-8e1b-408110413a19",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2020-11-23T13:32:57.450956Z",
  "updatedAt": "2020-11-23T13:32:57.505909Z",
  "connection_status": "connection ok"
}
```

Managing Luna Network HSM Partitions in STC Mode using ksctl

Note

- HSM-anchored domains are not supported with STC partitions.

To use HSM partitions in the STC mode:

1. Download/scp the partition identity public key (pid) file from the first client or HSM.
2. Register the partition identity public key on the CipherTrust Manager.
3. Add label to the HSM partition in the `partition_label` field.
4. Add serial number to the HSM partition in the `serial_number` field.

The following operations can be performed:

- Create/Get/Delete a Luna Network HSM STC partition
- List all Luna Network HSM STC partitions
- Get test status of a new STC partition

Creating a Luna Network HSM STC Partition

To create a Luna Network HSM STC-partition, run:

Syntax

```
ksctl connectionmgmt luna-hsm stc-partition create --name <connection-name> --products <product-names> --meta <key:value> --partition_identity <partitions-identity-file> --partition_label <partition-label-name> --serial_number <serial-number>
```

Example Request

```
ksctl connectionmgmt luna-hsm stc-partition create --name T1332 --products cckm
--serial-number 14655971025300 --partition-identity 1465065595818.pid --label T12
3
```

Example Response

```
{
  "id": "fda66b48-1191-4c06-b1f7-076b5f59dcbe",
  "uri": "kylo:kylo:connectionmgmt:hsm-stc-partition:t1332-fda66b48-1191-4c06-b
1f7-076b5f59dcbe",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-09-19T12:20:30.481081209Z",
  "updatedAt": "2022-09-19T12:20:30.476753184Z",
  "service": "luna network",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "T1332",
  "products": [
    "cckm"
  ],
  "label": "T123",
  "serial_number": "14655971025300"
}
```

Getting Details of Luna Network HSM STC Partition

To get details of a Luna Network HSM partition in the STC mode, run:

Syntax

```
ksctl connectionmgmt luna-hsm stc-partition get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt luna-hsm stc-partition get --id fda66b48-1191-4c06-b1f7-076b
5f59dcbe
```

Example Response

```
{
  "id": "fda66b48-1191-4c06-b1f7-076b5f59dcbe",
  "uri": "kylo:kylo:connectionmgmt:hsm-stc-partition:t1332-fda66b48-1191-4c06-b
1f7-076b5f59dcbe",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-09-19T12:20:30.481081Z",
  "updatedAt": "2022-09-19T12:20:30.476753Z",
  "service": "luna network",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "T1332",
  "products": [
    "cckm"
  ]
}
```

```

    ],
    "label": "T123",
    "serial_number": "14655971025300"
  }

```

Deleting Luna Network HSM STC Partitions

To delete Luna Network HSM partition in the STC mode, run:

Syntax

```
ksctl connectionmgmt luna-hsm stc-partition delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt luna-hsm stc-partition delete --id fda66b48-1191-4c06-b1f7-076b5f59dcbe
```

Example Response

```

{
  "status": 204
}

```

The above response appears if the Luna Network HSM partition in the STC mode deletes successfully. In case of failure, it will throw an error.

Getting Test Status of a New STC Partition

To get a status of a new Luna network STC partition, run:

Syntax

```
ksctl connectionmgmt luna-hsm stc-partition delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt luna-hsm stc-partition status --id fda66b48-1191-4c06-b1f7-076b5f59dcbe
```

Example Response - if partition is created successfully

```

{
  "connection_ok": true,
}

```

Example Response - if partition is not created


```
{
  "connection_ok": false,
  "connection_error": "Cannot list STC partition slot"
}
```

Elasticsearch

Elasticsearch connections to the CipherTrust Manager can be configured using the following:

- [Managing Elasticsearch Connections using GUI](#)
- [Managing Elasticsearch Connections using ksctl](#)

Note

CipherTrust Manager can have a total of 64 log forwarder connections. Each Elasticsearch, Loki, and Syslog connection is counted towards the 64 connection total.

Managing Elasticsearch Connections using GUI

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.
4. On the Add Connection screen, select category as **Log Forwarder**.
5. Select Log Forwarder Type as **Elasticsearch** and click **Next**.
6. Specify connection **Name** and **Description** and click **Next**.
7. Configure the below parameters.
 - **Host:** IP address or hostname of the Elasticsearch server.
 - **Port:** port number for connecting to the Elasticsearch server.
 - **HTTP User:** name of the HTTP user for basic authentication.
 - **HTTP Password:** password of the HTTP user.
 - **Transport:** protocol for transport. Either TCP or TLS. TLS requires additional settings:

Settings for TLS:

You must provide **CA Certificate** or select **Skip Certificate Verification**

- **CA Certificate:** either upload the CA certificate or paste the certificate content. This is a CA which directly or indirectly signs the Elasticsearch server certificate.
 - **File Upload:** select and click **Upload Certificate** to upload the trusted CA certificate from your machine. In TLS mode, the CA certificate must be in PEM format.
 - **Text:** select and paste the certificate content in the text field.
 - **Skip Certificate Verification:** select if the TLS verification of the CA certificate is not required.
8. Click **Test Credentials** to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.
 9. Click **Next** to move to the **Add Products** screen of the Add Connection wizard.

Managing Elasticsearch Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete an Elasticsearch connection
- List all Elasticsearch connections
- Test an existing Elasticsearch connection
- Test the new Elasticsearch connection parameters before establishing the connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Unique name of the connection.
description	Optional	Connection description.
products	Optional	List of products.
host	Mandatory	Host of the log-forwarder server.
port	Mandatory	Port of the log-forwarder server.
transport	Optional	Transport mode for sending data, supports "tls" and "tcp". The "tls" mode requires either a trusted CA certificate or insecure TLS skip verify to be set to true. The default transport mode is "tcp".
json-file	Optional	meta information in json format. This information is provided in pair. For example, --meta <code>"{"color":"blue","foo":"bar"}"</code> .

Parameter	Mandatory/Optional	Description
http-user	Optional	HTTP basic auth username.
http-password	Optional	HTTP basic auth password.
ca_cert	Optional	CA certificate in PEM format.
insecure-skip-verify	Optional	In TLS mode, skips server certificate validation. This setting should only be used for testing.

Elasticsearch connections can be authenticated with username and password or unauthenticated. Check the Elasticsearch deployment for the authentication required for incoming data.

The `tls` mode requires either a trusted CA certificate in the PEM format or `--insecure-skip-verify` set to `true`. By default, the CipherTrust Manager attempts to validate Elasticsearch's server certificate using the provided CA. The external CAs already present on the CipherTrust Manager are not used. Use `--ca-cert` to provide a CA which directly or indirectly signs the server certificate. Use the `--insecure-skip-verify` flag to indicate that CipherTrust Manager does not verify Elasticsearch's presented server certificate.

Creating an Elasticsearch Connection

To create an Elasticsearch connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder elasticsearch create --name <connection-name>
--products <products-name> --description <description> --host <host> --port <port>
--transport <transport-protocol> --http-user <username> --http-password <password>
--ca-cert <ca-cert> --meta <Key:Value> --insecure-skip-verify
```

The supported transport mode for sending data are `tls` and `tcp`. The `tls` mode requires either a trusted CA certificate in the PEM format or `--insecure-skip-verify` set to `true`.

Example Request

```
ksctl connectionmgmt log-forwarder elasticsearch create --name elasticsearch-conn-1
--description conn-description --host 127.0.0.1 --port 1234 --transport tcp
--http-user admin --http-password password
```

Example Response

```
{
  "id": "9d3af367-d4a3-4838-8663-ce07d3e88353",
  "uri": "kylo:kylo:connectionmgmt:connections:elasticsearch-conn-1-9d3af367-d4a3-4838-8663-ce07d3e88353",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-04T07:53:49.838640861Z",
  "updatedAt": "2022-05-04T07:53:49.837431591Z",
  "service": "elasticsearch",
  "category": "log-forwarders",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "elasticsearch-conn-1",
```

```

    "description": "conn-description",
    "host": "127.0.0.1",
    "port": 1234,
    "elasticsearch_params": {
      "insecure_tls_skip_verify": false,
      "transport": "tcp",
      "http_user": "admin"
    }
  }
}

```

Getting Details of an Elasticsearch Connection

To get details of an Elasticsearch connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder elasticsearch get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt log-forwarder elasticsearch get --id 9d3af367-d4a3-4838-8663-ce07d3e88353
```

Example Response

```

{
  "id": "9d3af367-d4a3-4838-8663-ce07d3e88353",
  "uri": "kylo:kylo:connectionmgmt:connections:elasticsearch-conn-1-9d3af367-d4a3-4838-8663-ce07d3e88353",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-04T07:53:49.838641Z",
  "updatedAt": "2022-05-04T07:53:49.837432Z",
  "service": "elasticsearch",
  "category": "log-forwarders",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "elasticsearch-conn-1",
  "description": "conn-description",
  "host": "127.0.0.1",
  "port": 1234,
  "elasticsearch_params": {
    "insecure_tls_skip_verify": false,
    "transport": "tcp",
    "http_user": "admin"
  }
}

```

Updating an Elasticsearch Connection

To update an Elasticsearch connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder elasticsearch modify --id <Connection-Name/ID> --products <products-name> --description <description> --host <host> --port <port> --transport <transport-protocol> --http-user <username> --http-password <password> --ca-cert <ca-cert> --meta <Key:Value> --insecure-skip-verify
```

The supported transport mode for sending data are `tls` and `tcp`. The `tls` mode requires either a trusted CA certificate in the PEM format or `--insecure-skip-verify` set to `true`.

Example Request

```
ksctl connectionmgmt log-forwarder elasticsearch modify --id 9d3af367-d4a3-4838-8663-ce07d3e88353 --host 127.0.0.1
```

Example Response

```
{
  "id": "9d3af367-d4a3-4838-8663-ce07d3e88353",
  "uri": "kylo:kylo:connectionmgmt:connections:elasticsearch-conn-1-9d3af367-d4a3-4838-8663-ce07d3e88353",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-04T07:53:49.838641Z",
  "updatedAt": "2022-05-04T08:17:20.615258139Z",
  "service": "elasticsearch",
  "category": "log-forwarders",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "elasticsearch-conn-1",
  "description": "conn-description",
  "host": "127.0.0.1",
  "port": 1234,
  "elasticsearch_params": {
    "insecure_tls_skip_verify": false,
    "transport": "tcp",
    "http_user": "admin"
  }
}
```

Deleting an Elasticsearch Connection

To delete an Elasticsearch connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder elasticsearch delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt log-forwarder elasticsearch delete --id 9d3af367-d4a3-4838-8663-ce07d3e88353
```

Example Response

There will be no response if the Elasticsearch connection is deleted successfully.

Getting List of Elasticsearch Connections

To list all the Elasticsearch connections, run:

Syntax

```
ksctl connectionmgmt log-forwarder elasticsearch list
```

Example Request

```
ksctl connectionmgmt log-forwarder elasticsearch list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "9d3af367-d4a3-4838-8663-ce07d3e88353",
      "uri": "kylo:kylo:connectionmgmt:connections:elasticsearch-conn-1-9d3af367-d4a3-4838-8663-ce07d3e88353",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2022-05-04T07:53:49.839242Z",
      "updatedAt": "2022-05-04T07:53:49.839242Z",
      "service": "elasticsearch",
      "category": "log-forwarders",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "elasticsearch-conn-1",
      "description": "conn-description",
      "host": "127.0.0.1",
      "port": 1234
    }
  ]
}
```

Testing an Existing Elasticsearch Connection

Note

Test Connection verifies the status of the server certificate if it is revoked or not by using the [Certificate Revocation Lists \(CRLs\)](#). If the certificate is found to be revoked, the test connection fails. If an error occurs during the evaluation of the certificate's revocation status, the connection remains in a soft fail state and test connection is allowed to continue.

To test an existing Elasticsearch connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder elasticsearch test --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt log-forwarder elasticsearch test --id 9d3af367-d4a3-4838-8663-ce07d3e88353
```

Example Response

```
{
  "connection_ok": true
}
```

Testing a New Elasticsearch Connection

Note

Test Connection verifies the status of the server certificate if it is revoked or not by using the [Certificate Revocation Lists \(CRLs\)](#). If the certificate is found to be revoked, the test connection fails. If an error occurs during the evaluation of the certificate's revocation status, the connection remains in a soft fail state and test connection is allowed to continue.

To test the parameters of a New Elasticsearch connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder elasticsearch test --host <host> --port <port> --transport <transport-protocol> --http-user <username> --http-password <password> --ca-cert <ca-cert> --insecure-skip-verify
```

Example Request

```
ksctl connectionmgmt log-forwarder elasticsearch test --host 127.0.0.1 --port 1234 --transport tcp --http-user admin --http-password password
```

Example Response

```
{
  "connection_ok": true
}
```

SAP Data Custodian

SAP Data Custodian connections to the CipherTrust Manager can be configured using the following:

- [Managing SAP Data Custodian Connections using GUI](#)
- [Managing SAP Data Custodian Connections using ksctl](#)

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.

Managing SAP Data Custodian Connections using GUI

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.
4. On the Add Connection screen, select category as **Cloud**.
5. Select Select Cloud Type as **SAP Data Custodian** and click **Next**.
6. Specify connection **Name** and **Description** and click **Next**.
7. Configure the below parameters.

Technical Users

[Technical Users](#) are essentially API clients. The technical users can have full admin access to an entire service such as Key Management Service.

Note

Technical User credentials are activated internally at the back end while creating/updating connections. Therefore, the create/update connection won't accept the already activated technical user credentials.

- **API Endpoint** - this is the KMS API endpoint of the SAP Data Custodian. Provide HTTP URL with the API

version in it. Only v2 version of the KMS API is supported. To get the SAP API endpoint:

- a. Create a technical user (TU).
- b. Generate its credentials and download them.

The downloaded file "API Endpoints.txt" contains ISM and KMS API endpoints. Use the KMS API endpoint to make the connection.

- **Secret** - provide secret (password).
- **API Key** - provide API key of the technical user.

The **Test Credentials** will fail until the credentials are activated. The credentials will activate only after creating a connection. Therefore, it is recommended to test the connection after creating it. To do so, click the **Test Connection** button corresponding to the newly added connection, on the Connections page.

Standard Users

Standard users are owned by the IAM service, and represent human users. Standard users need to be assigned to the Key Management Service, with a certain role, and then need to be added to groups to see those groups. The standard users can access both the UI and the API.

- **API Endpoint** - this is the KMS API endpoint of the SAP Data Custodian. Provide HTTP URL with the API version in it. Only v2 version of the KMS API is supported. Example - `https://kms-api-demo.datacustodian.cloud.sap/kms/v2`.
- **Username** - provide username to access the SAP data custodian server.
- **Secret** - provide secret (password).
- **Tenant** - provide tenant.

8. Click the **Test Credentials** button to check whether the connection is configured correctly. If the test is successful, the status is `OK` else the status is `Fail`.

You can proceed with adding the SAP connection even if clicking the **Test Credentials** returns an error message. You can test the connection after adding it. Click the **Test Connection** button corresponding to the newly added connection, on the Connections page.

9. Click **Next** to move to the [Add Products](#) screen of the Add Connection wizard.

Note

Currently, the only product supported for SAP Data Custodian connection is *Cloud Key Manager*.

Managing SAP Data Custodian Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete an SAP Data Custodian connection
- List all SAP Data Custodian connections
- Test an existing SAP Data Custodian connection
- Test parameters for a SAP Data Custodian connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Name of the connection.
description	Optional	Connection description.
products	Optional	List of products.
api-endpoint	Mandatory	KMS API endpoint of the SAP Data Custodian. Provide HTTP URL with the API version in it. Only v2 version of KMS API is supported. For example, <code>https://kms-api-demo.datacustodian.cloud.sap/kms/v2</code> .
tech-user-creds	Optional	Technical User Credentials for SAP Data Custodian connection.
api_key	Mandatory	API key of the technical user.
secret	Mandatory	Secret/Password of the standard/technical user.
user-creds	Optional	Standard User Credentials for SAP Data Custodian connection.
user	Mandatory	Username.
tenant	Mandatory	Tenant of the user.
meta	Optional	Meta information in json format. This information is provided in pair. For example, <code>--meta "{\"color\":\"blue\",\"foo\":\"bar\"}"</code> .
json-file	Optional	Connection information provided in a JSON file format. Command line parameters will take precedence over values specified in the JSON file.

Note

The examples are provided for the Windows platform. To run these examples on Linux, see [CLI usage with JSON parameters](#).

Technical Users

Technical Users are essentially API clients. The technical users can have full admin access to an entire service such as Key Management Service.

Note

Technical User credentials are activated internally at the back end while creating/updating connections. Therefore, the create/update connection won't accept the already activated technical user credentials.

Creating a SAP Data Custodian Connection

To create a SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc create --name <Connection-Name> --products <Products-Names> --api-endpoint <SAPDataCustodian-API-Endpoint> --tech-user-creds <API key,secret>
```

Here, `--api-endpoint` is the KMS API endpoint of the SAP Data Custodian. Provide HTTP URL with the API version in it. Only v2 version of the KMS API is supported. To get the SAP API endpoint:

1. Create a technical user (TU).
2. Generate its credentials and download them.

The downloaded file "API Endpoints.txt" contains ISM and KMS API endpoints. Use the KMS API endpoint to make the connection.

Example Request

```
ksctl connectionmgmt sap-dc create --name "test-tu-conn" --api-endpoint "https://kms-api-aws-demo.datacustodian.cloud.sap/kms/v2" --tech-user-creds "{\"api_key\":\"ey----NhCJ9\", \"secret\":\"0U6myfwji--ye\"}"
```

Example Response

```
{
  "id": "9a6a728b-5beb-465a-8e2e-5e4332039b2d",
  "uri": "kylo:kylo:connectionmgmt:connections:test-tu-conn-9a6a728b-5beb-465"
```

```

a-8e2e-5e4332039b2d",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2024-01-09T06:22:48.058817364Z",
  "updatedAt": "2024-01-09T06:22:48.057206571Z",
  "service": "sap-data-custodian",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "test-tu-conn",
  "api_endpoint": "https://kms-api-aws-demo.datacustodian.cloud.sap/kms/v2",
  "technical_user_credentials": {
    "api_key": "ey----NhccJ9"
  }
}

```

Getting Details of a SAP Data Custodian Connection

To get details of a SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt sap-dc get --id 9a6a728b-5beb-465a-8e2e-5e4332039b2d
```

Example Response

```

{
  "id": "9a6a728b-5beb-465a-8e2e-5e4332039b2d",
  "uri": "kylo:kylo:connectionmgmt:connections:test-tu-conn-9a6a728b-5beb-465a-8e2e-5e4332039b2d",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2024-01-09T06:22:48.058817364Z",
  "updatedAt": "2024-01-09T06:22:48.057206571Z",
  "service": "sap-data-custodian",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "test-tu-conn",
  "api_endpoint": "https://kms-api-aws-demo.datacustodian.cloud.sap/kms/v2",
  "technical_user_credentials": {
    "api_key": "ey----NhccJ9"
  }
}

```

Updating a SAP Data Custodian Connection

To update a SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc modify --id <Connection-Name/ID> --products <Product
s-Names> --api-endpoint <SAPDataCustodian-API-Endpoint> --tech-user-creds <api-ke
y,secret> --meta <Key:Values>
```

Example Request

```
ksctl connectionmgmt sap-dc modify --id "9a6a728b-5beb-465a-8e2e-5e4332039b2d" --
api-endpoint "https://kms-api-aws-demo.datacustodian.cloud.sap/kms/v2" --tech-use
r-creds "{\"api_key\":\"eyJjcmVkdW-----cCJ9\", \"secret\":\"oYtZk-----wnWGi\"}"
```

Example Response

```
{
  "id": "9a6a728b-5beb-465a-8e2e-5e4332039b2d",
  "uri": "kylo:kylo:connectionmgmt:connections:test-tu-conn-9a6a728b-5beb-465
a-8e2e-5e4332039b2d",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2024-01-09T06:22:48.058817Z",
  "updatedAt": "2024-01-09T06:34:30.297756622Z",
  "service": "sap-data-custodian",
  "category": "cloud",
  "last_connection_ok": true,
  "last_connection_at": "2024-01-09T06:29:56.429799Z",
  "name": "test-tu-conn",
  "api_endpoint": "https://kms-api-aws-demo.datacustodian.cloud.sap/kms/v2",
  "technical_user_credentials": {
    "api_key": "eyJjcmVkdW-----cCJ9"
  }
}
```

Deleting a SAP Data Custodian Connection

To delete a SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt sap-dc delete --id 9a6a728b-5beb-465a-8e2e-5e4332039b2d
```

Example Response

There will be no response if SAP Data Custodian connection is deleted successfully.

Getting List of SAP Data Custodian Connections

To list all the SAP Data Custodian connections, run:

Syntax

```
ksctl connectionmgmt sap-dc list
```

Example Request

```
ksctl connectionmgmt sap-dc list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "9a6a728b-5beb-465a-8e2e-5e4332039b2d",
      "uri": "kylo:kylo:connectionmgmt:connections:test-tu-conn-9a6a728b-5beb-465a-8e2e-5e4332039b2d",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2024-01-09T06:22:48.058817Z",
      "updatedAt": "2024-01-09T06:34:30.297756622Z",
      "service": "sap-data-custodian",
      "category": "cloud",
      "last_connection_ok": true,
      "last_connection_at": "2024-01-09T06:29:56.429799Z",
      "name": "test-tu-conn",
      "api_endpoint": "https://kms-api-aws-demo.datacustodian.cloud.sap/kms/v2",
      "technical_user_credentials": {
        "api_key": "eyJJcmVkbWZ-----cCJ9"
      }
    }
  ]
}
```

Testing an Existing SAP Data Custodian Connection

To test an existing SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc test --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt sap-dc test --id "9a6a728b-5beb-465a-8e2e-5e4332039b2d"
```

Example Response

```
{
  "connection_ok": true
}
```

Testing Parameters for a SAP Data Custodian Connection

Testing parameters API doesn't activate the technical user credentials. It will only make an API call to `api_endpoint/auth/request`.

To test parameters for a SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc test --api-endpoint <SAPDataCustodian-API-Endpoint>
--tech-user-creds <api-key,secret>
```

Example Request

```
ksctl connectionmgmt sap-dc test --api-endpoint "https://test-endpoint.com" --tec
h-user-creds "{\"api_key\":\"eyJjcmVkdW-----cCJ9\", \"secret\":\"oYtZk-----wnWG
i\"}"
```

Example Response

```
{
  "connection_ok": true
}
```

Standard Users

Standard users are owned by the IAM service, and represent human users. Standard users need to be assigned to the Key Management Service, with a certain role, and then need to be added to Groups to see those groups. The standard users can access both the UI and the API.

Creating a SAP Data Custodian Connection

To create a SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc create --name <Connection-Name> --products <Products-
Names> --api-endpoint <SAPDataCustodian-API-Endpoint> --user-creds <user,secret,t
enant-JSON-Format-String>
```

Example Request

```
ksctl connectionmgmt sap-dc create --name test-conn --products "cckm" --api-endpo
int "https://test-endpoint.com" --user-creds "{\"user\":\"testuser\", \"secre
t\":\"testsecret\", \"tenant\":\"testtenant\"}"
```

Example Response

```
{
  "id": "d2e25ea2-de0f-488a-94f4-d3c925cd5d18",
```

```

    "uri": "kylo:kylo:connectionmgmt:connections:test-conn-d2e25ea2-de0f-488a-94f4-d3c925cd5d18",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2021-10-27T07:21:22.77127493Z",
    "updatedAt": "2021-10-27T07:21:22.770209257Z",
    "service": "sap-data-custodian",
    "category": "cloud",
    "last_connection_ok": null,
    "last_connection_at": "0001-01-01T00:00:00Z",
    "name": "test-conn",
    "products": [
      "cckm"
    ],
    "api_endpoint": "https://test-endpoint.com",
    "user_credentials": {
      "tenant": "testtenant",
      "user": "testuser"
    }
  }
}

```

Getting Details of a SAP Data Custodian Connection

To get details of a SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt sap-dc get --id d2e25ea2-de0f-488a-94f4-d3c925cd5d18
```

Example Response

```

{
  "id": "d2e25ea2-de0f-488a-94f4-d3c925cd5d18",
  "uri": "kylo:kylo:connectionmgmt:connections:test-conn-d2e25ea2-de0f-488a-94f4-d3c925cd5d18",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-10-27T07:21:22.771275Z",
  "updatedAt": "2021-10-27T07:21:22.770209Z",
  "service": "sap-data-custodian",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "test-conn",
  "products": [
    "cckm"
  ],
  "api_endpoint": "https://test-endpoint.com",
  "user_credentials": {
    "tenant": "testtenant",
    "user": "testuser"
  }
}

```



```
}
```

Updating a SAP Data Custodian Connection

To update a SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc modify --id <Connection-Name/ID> --products <Product
s-Names> --api-endpoint <SAPDataCustodian-API-Endpoint> --user-creds <user,secre
t,tenant-JSON-Format-String> --meta <Key:Values>
```

Example Request

```
ksctl connectionmgmt sap-dc modify --id d2e25ea2-de0f-488a-94f4-d3c925cd5d18 --pr
oducts "cckm" --api-endpoint "https://test2-endpoint.com" --user-creds "{\"use
r\":\"testuser2\", \"secret\":\"testsecret2\", \"tenant\":\"testtenant2\"}"
```

Example Response

```
{
  "id": "d2e25ea2-de0f-488a-94f4-d3c925cd5d18",
  "uri": "kylo:kylo:connectionmgmt:connections:test-conn-d2e25ea2-de0f-488a-94f
4-d3c925cd5d18",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-10-27T07:21:22.771275Z",
  "updatedAt": "2021-10-27T07:26:11.431339116Z",
  "service": "sap-data-custodian",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "test-conn",
  "products": [
    "cckm"
  ],
  "api_endpoint": "https://test2-endpoint.com",
  "user_credentials": {
    "tenant": "testtenant2",
    "user": "testuser2"
  }
}
```

Deleting a SAP Data Custodian Connection

To delete a SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt sap-dc delete --id d2e25ea2-de0f-488a-94f4-d3c925cd5d18
```

Example Response

There will be no response if SAP Data Custodian connection is deleted successfully.

Getting List of SAP Data Custodian Connections

To list all the SAP Data Custodian connections, run:

Syntax

```
ksctl connectionmgmt sap-dc list
```

Example Request

```
ksctl connectionmgmt sap-dc list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "d2e25ea2-de0f-488a-94f4-d3c925cd5d18",
      "uri": "kylo:kylo:connectionmgmt:connections:test-conn-d2e25ea2-de0f-488a-94f4-d3c925cd5d18",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2021-10-27T07:21:22.771275Z",
      "updatedAt": "2021-10-27T07:21:22.770209Z",
      "service": "sap-data-custodian",
      "category": "cloud",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "test-conn",
      "products": [
        "cckm"
      ],
      "api_endpoint": "https://test-endpoint.com",
      "user_credentials": {
        "tenant": "testtenant",
        "user": "testuser"
      }
    }
  ]
}
```

Testing an Existing SAP Data Custodian Connection

To test an existing SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc test --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt sap-dc test --id d2e25ea2-de0f-488a-94f4-d3c925cd5d18
```

Example Response

```
{
  "connection_ok": true
}
```

Testing Parameters for a SAP Data Custodian Connection

To test parameters for a SAP Data Custodian connection, run:

Syntax

```
ksctl connectionmgmt sap-dc test --api-endpoint <SAPDataCustodian-API-Endpoint>
--user-creds <user,secret,tenant-JSON-Format-String>
```

Example Request

```
ksctl connectionmgmt sap-dc test --api-endpoint "https://test-endpoint.com" --use
r-creds "{\"user\":\"testuser\",\"secret\":\"testsecret\",\"tenant\":\"testtenan
t\"}"
```

Example Response

```
{
  "connection_ok": true
}
```

CIFS/SMB

CIFS/SMB connections to the CipherTrust Manager can be configured using the following:

- [Managing SMB Connections using GUI](#)
- [Managing SMB Connections using ksctl](#)

Managing SMB Connections using GUI

Note

The Common Internet File System (CIFS) is a dialect of SMB.

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.
4. On the Add Connection screen, select category as **More**.
5. Select connection type as **CIFS/SMB** and click **Next**.
6. Specify connection **Name** and **Description** and click **Next**.
7. Configure the below parameters.
 - **Host** - IP or FQDN of the SMB share server.
 - **Port** - the port where the SMB service is running on the host.

Note

The *Host* and *Port* fields must be specified together, or do not specify any of them. If *Host* and *Port* are not specified while creating a connection, these fields cannot be added later.

- **Username** - username to access the SMB share.
 - **Password** - password to access the SMB share.
 - **Domain** - workgroup or domain under which the username is configured. It is an optional field.
 - **Test Path** - path to the file-share for which the credentials need to be tested. It is only required to test the connections.
8. Click the **Test Credentials** button to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.

Note

The *Host*, *Port*, and *Test Path* fields are mandatory for testing the connection credentials.

- Click **Next** to move to the [Add Products](#) screen of the Add Connection wizard.

Managing SMB Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete an SMB connection
- List all SMB connections
- Test an existing SMB connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Name of the connection.
description	Optional	Connection description.
products	Optional	List of products.
conn-domain	Optional	Domain for SMB share.
port	Optional	Port where SMB service runs on host (usually 445).
username	Mandatory	Username for accessing SMB share.
conn-password	Mandatory	Password for SMB share.
meta	Optional	meta information in json format. This information is provided in pair. For example, --meta <code>{"color\":"blue\","foo\":"bar\}"</code> .
json-file	Optional	Connection information provided in a JSON file format. Command line parameters will take precedence over values specified in the JSON file.

Creating an SMB Connection

To create an SMB connection, run:

Syntax

```
ksctl connectionmgmt smb create --name <Connection-Name> --host <Hostname> --port <Port> --Username <SMB-Connection-Username> --conn-password <SMB-Connection-Password>
```

Example Request

```
ksctl connectionmgmt smb create --name smbConnection1 --host xx.xxx.xxx.xxx --port 445 --username test1 --conn-password 1234
```

Example Response

```
{
  "id": "5b32eb85-5c8e-4416-a749-b0b0b8916213",
  "uri": "kylo:kylo:connectionmgmt:connections:smbconnection1-5b32eb85-5c8e-4416-a749-b0b0b8916213",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-02-03T08:49:10.5042906Z",
  "updatedAt": "2021-02-03T08:49:10.502989201Z",
  "service": "SMB",
  "category": "File-Share",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "smbConnection1",
  "username": "test1",
  "host": "xx.xxx.xxx.xxx",
  "Port": "445"
}
```

Getting Details of an SMB Connection

To get details of an SMB connection, run:

Syntax

```
ksctl connectionmgmt smb get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt smb get --id 7ee52a57-19d4-4202-b900-edd299bb6d7a
```

Example Response

```
{
  "id": "7ee52a57-19d4-4202-b900-edd299bb6d7a",
  "uri": "kylo:kylo:connectionmgmt:connections:smbconnection-7ee52a57-19d4-4202-b900-edd299bb6d7a",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-02-03T03:55:03.541011Z",
  "updatedAt": "2021-02-03T03:55:03.538119Z",
}
```

```

    "service": "SMB",
    "category": "File-Share",
    "last_connection_ok": null,
    "last_connection_at": "0001-01-01T00:00:00Z",
    "name": "smbConnection",
    "products": [
      "CTE"
    ],
    "description": "a description of the connection",
    "username": "user",
    "domain": "domainName",
    "host": "1.2.3.4",
    "Port": "445"
  }

```

Updating an SMB Connection

To update an SMB connection, run:

Syntax

```

ksctl connectionmgmt smb modify --id <Connection-Name/ID> --products <Product-Names> --conn-password <SMB-Connection-Password> --meta <Key-Values>

```

Example Request

```

ksctl connectionmgmt smb modify --id 7ee52a57-19d4-4202-b900-edd299bb6d7a --conn-password 1234 --username test2

```

Example Response

```

{
  "id": "7ee52a57-19d4-4202-b900-edd299bb6d7a",
  "uri": "kylo:kylo:connectionmgmt:connections:smbconnection-7ee52a57-19d4-4202-b900-edd299bb6d7a",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-02-03T03:55:03.541011Z",
  "updatedAt": "2021-02-03T08:58:21.490072287Z",
  "service": "SMB",
  "category": "File-Share",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "smbConnection",
  "products": [
    "CTE"
  ],
  "meta": "",
  "description": "a description of the connection",
  "host": "1.2.3.4",
  "domain": "domainName",
  "username": "test2",
  "port": "445"
}

```

Deleting an SMB Connection

To delete an SMB connection, run:

Syntax

```
ksctl connectionmgmt smb delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt smb delete --id 7ee52a57-19d4-4202-b900-edd299bb6d7a
```

There will be no response if SMB connection is deleted successfully.

Getting List of SMB Connections

To list all the SMB connections, run:

Syntax

```
ksctl connectionmgmt smb list
```

Example Request

```
ksctl connectionmgmt smb list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "id": "5b32eb85-5c8e-4416-a749-b0b0b8916213",
      "uri": "kylo:kylo:connectionmgmt:connections:smbconnection1-5b32eb85-5c8e-4416-a749-b0b0b8916213",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2021-02-03T08:49:10.504291Z",
      "updatedAt": "2021-02-03T08:49:10.502989Z",
      "service": "SMB",
      "category": "File-Share",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "smbConnection1",
      "username": "test1",
      "host": "xx.xxx.xxx.xxx",
      "Port": "445"
    },
    {
      "id": "7ee52a57-19d4-4202-b900-edd299bb6d7a",
      "uri": "kylo:kylo:connectionmgmt:connections:smbconnection-7ee52a57-1"
```



```

9d4-4202-b900-edd299bb6d7a",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-02-03T03:55:03.541011Z",
  "updatedAt": "2021-02-03T03:55:03.538119Z",
  "service": "SMB",
  "category": "File-Share",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "smbConnection",
  "products": [
    "CTE"
  ],
  "description": "a description of the connection",
  "username": "user",
  "domain": "domainName",
  "host": "1.2.3.4",
  "Port": "445"
}
]
}

```

Testing an Existing SMB Connection

To test an existing SMB connection, run:

Syntax: Option 1

```
ksctl connectionmgmt smb test --id <Connection-Name/ID> --share-path <Share-Path>
```

Syntax: Option 2

```
ksctl connectionmgmt smb test --host <Hostname> --port <Port> --username <SMB-Connection-Username> --conn-password <SMB-Connection-Password> --share-path <Share-Path>
```

Example Request 1

```
ksctl connectionmgmt smb test --id 7ee52a57-19d4-4202-b900-edd299bb6d7a --share-path shared
```

Example Response 1

```
{
  "connection_ok": true
}
```

Example Request 2

```
ksctl connectionmgmt smb test --host xx.xxx.xxx.xxx --port 445 --username test1 --conn-password 1234 --share-path shared
```

Example Response 2

```
{
  "connection_ok": true
}
```

Syslog

Syslog connections to the CipherTrust Manager can be configured using the following:

- [Managing Syslog Connections using GUI](#)
- [Managing Syslog Connections using ksctl](#)

Note

CipherTrust Manager can have a total of 64 log forwarder connections. Each [Elasticsearch](#), [Loki](#), and Syslog connection is counted towards the 64 connection total.

Managing Syslog Connections using GUI

Log on to CipherTrust Manager UI as an administrator.

1. Navigate to **Access Management > Connections**.
2. Click **Add Connection**.
3. On the Add Connection screen, select category as **Log Forwarder**.
4. Select Log Forwarder Type as **Syslog** and click **Next**.

Add Connection ✕

1 Select Connection Type
2 General Info
3 Configure Connection
4 Add Products

ⓘ To add an Azure or SFDC connection using external certificates, you first need to create and sign a Connection CSR using the [CSR generator in CA](#).

Select Category

Cloud

HSM

Key Manager

OIDC/LDAP

TDP

SFTP/SCP

Log Forwarder

More

Select Log Forwarder Type*

Loki
Loki
Elasticsearch
Syslog

Next

5. Specify connection **Name** and **Description** and click **Next**.

6. Configure the below parameters.

- **Host:** IP address or hostname of the Syslog server.
- **Port:** port number for connecting to the Syslog server.
- **Transport Format:** select the transport mode for sending data. The **TLS** mode requires a trusted CA certificate in the PEM format.

Note

If you set the transport format to **UDP**, log messages are limited to a size of 1024 bytes. After this size, the log message is truncated.

- **CA Cert:** either upload the CA certificate or paste the certificate content. Make sure the server certificate contains the valid IP SANs.
 - **Upload CSR:** select and click **Upload CSR** to upload the trusted CA certificate from your machine.
 - **Text:** select and paste the certificate content in the text field.

- **Message Format:** select the log message format.

7. Click **Test Credentials** to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.
8. Click **Next** to move to the **Add Products** screen of the Add Connection wizard.

Managing Syslog Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete a Syslog connection
- List all Syslog connections
- Test an existing Syslog connection
- Test the new Syslog connection parameters before establishing the connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Unique name of the connection.
description	Optional	Connection description.
products	Optional	List of products.
host	Mandatory	Host of the log-forwarder server.
port	Mandatory	Port of the log-forwarder server.
transport	Mandatory	Transport mode for sending data, supports "udp", "tls" and "tcp". The tls mode requires a trusted CA certificate in the PEM format.
json-file	Optional	meta information in json format. This information is provided in pair. For example, --meta <code>{"color\":"blue\","foo\":"bar\}"</code> .
message-format	Optional	The log message format for new log messages: • rfc5424 (default) • rfc3164 • cef • leef
ca_cert	Optional	The trusted CA certificate in the PEM format. Only used in the TLS transport mode.

Note

In a multi-node clustered environment, the Syslog connections configuration is synchronized. Each node is aware of all the Syslog servers and Syslog messages are sent from the currently active node. This implies, that if an event that results in an audit record is performed on node 1, the Syslog message will originate from node 1. Similarly, if an audit event is conducted on node 2, the Syslog message will originate from node 2.

Log Message Formats

The supported message formats for Syslog are:

- rfc5424 (default)
- rfc3164
- cef
- leef

Examples

rfc5424 (plaintext)

```
2022-06-29T04:54:42.868478Z - CipherTrust_Manager_k170v citrus Server_Audit -
...{"principal":{"acct":"kylo:kylo:admin:accounts:kylo","sub":"local|d7c6473a-286
d-4e10-9455-3f078743d4a5","acc":"kylo","iss":"kylo"},"requestId":"c6ce87a7-1775-4
d4e-92ad-67e031b620f9","success":true,"username":"admin","details":{"category":"c
loud","identifier":"Syslog Connection","name":"Syslog Connection","port":555,"ser
vice":"syslog"},"account":"kylo:kylo:admin:accounts:kylo","id":"d0a8f585-35b3-441
4-99c1-c21db5d244d1","domain_id":"00000000-0000-0000-0000-000000000000","messag
e":"Update connection","severity":"info","service":"citrus"}
```

rfc3164

```
Jun 29 04:58:56 CipherTrust_Manager_k170v[citrus]: {"requestId":"bceb9c29-fda7-4c
d5-ae9c-6521b4913a09","success":true,"username":"admin","details":{"identifie
r":"Syslog Connection","name":"Syslog Connection","port":555,"service":"syslo
g","category":"cloud"},"severity":"info","id":"27dc8135-24be-4ee7-8b26-157f53bd62
64","account":"kylo:kylo:admin:accounts:kylo","domain_id":"00000000-0000-0000-000
0-000000000000","service":"citrus","message":"Update connection","principal":{"ac
c":"kylo","iss":"kylo","acct":"kylo:kylo:admin:accounts:kylo","sub":"local|d7c647
3a-286d-4e10-9455-3f078743d4a5"}}}
```

cef

```
Jun 29 05:01:43 CEF:0|Thales Group|CipherTrust_Manager_k170v|2.9.0-beta7+7422|Ser
ver_Audit|Update connection|1|log={"requestId":"b280abc6-4e88-46e4-9550-09409978a
2e1","success":true,"username":"admin","details":{"identifier":"Syslog Connectio
n","name":"Syslog Connection","port":4444,"service":"syslog","category":"clou
```

```
d"},"account":"kylo:kylo:admin:accounts:kylo","id":"b575dba4-0fc4-4ce9-8d04-215b29f9efea","domain_id":"00000000-0000-0000-0000-000000000000","severity":"info","service":"citrus","message":"Update connection","principal":{"acc":"kylo","sub":"local|d7c6473a-286d-4e10-9455-3f078743d4a5"},"acct":"kylo:kylo:admin:accounts:kylo","iss":"kylo"}}
```

leef

```
Jun 29 05:03:47 LEEF:2.0|Thales Group|CipherTrust_Manager_k170v|2.9.0-beta7+7422|Server_Audit|^|log={"account":"kylo:kylo:admin:accounts:kylo","id":"2ebf0f96-396c-4626-9f48-f425d21f358d","service":"citrus","principal":{"sub":"local|d7c6473a-286d-4e10-9455-3f078743d4a5"},"acc":"kylo","iss":"kylo","acct":"kylo:kylo:admin:accounts:kylo"},"domain_id":"00000000-0000-0000-0000-000000000000","requestId":"2a7c2428-ee7c-4864-a93f-820b303c958c","severity":"info","success":true,"message":"Update connection","username":"admin","details":{"category":"cloud","identifier":"Syslog Connection","name":"Syslog"}}
```

Creating a Syslog Connection

To create a Syslog connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder syslog create --name <connection-name> --products <products-name> --description <description> --host <host> --port <port> --transport <transport-protocol> --ca-cert <ca-cert> --message-format <message-format> --meta <Key:Value>
```

The supported transport mode for sending data are `tls`, `tcp`, and `udp`. The `tls` mode requires a trusted CA certificate in the PEM format. In `udp` mode, log messages are limited to a size of 1024 bytes. After this size, the log message is truncated.

Example Request

```
ksctl connectionmgmt log-forwarder syslog create --name syslog-conn-1 --description conn-description --host 127.0.0.1 --port 1234 --transport tcp --message-format efc5425
```

Example Response

```
{
  "id": "2ecc1922-57e2-416d-9023-95720419fa25",
  "uri": "kylo:kylo:connectionmgmt:connections:syslog-conn-1-2ecc1922-57e2-416d-9023-95720419fa25",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-05T04:53:58.893569708Z",
  "updatedAt": "2022-05-05T04:53:58.890021892Z",
  "service": "syslog",
  "category": "log-forwarders",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "syslog-conn-1",
}
```

```

    "description": "conn-description",
    "host": "127.0.0.1",
    "port": 1234,
    "syslog_params": {
      "transport": "tcp",
      "message_format": "rfc5424"
    }
  }
}

```

Getting Details of a Syslog Connection

To get details of a Syslog connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder syslog get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt log-forwarder syslog get -id 2ecc1922-57e2-416d-9023-95720419fa25
```

Example Response

```

{
  "id": "2ecc1922-57e2-416d-9023-95720419fa25",
  "uri": "kylo:kylo:connectionmgmt:connections:syslog-conn-1-2ecc1922-57e2-416d-9023-95720419fa25",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-05T04:53:58.893569708Z",
  "updatedAt": "2022-05-05T04:53:58.890021892Z",
  "service": "syslog",
  "category": "log-forwarders",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "syslog-conn-1",
  "description": "conn-description",
  "host": "127.0.0.1",
  "port": 1234,
  "syslog_params": {
    "transport": "tcp",
    "message_format": "rfc5424"
  }
}

```

Updating a Syslog Connection

To update a Syslog connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder syslog modify --id <Connection-Name/ID> --products <products-name> --description <description> --host <host> --port <port> --transport <transport-protocol> --ca-cert <ca-cert> --message-format <message-format> --meta <Key:Value>
```

The supported transport mode for sending data are `tls`, `tcp`, and `udp`. The `tls` mode requires a trusted CA certificate in the PEM format.

Example Request

```
ksctl connectionmgmt log-forwarder syslog modify --id 9d3af367-d4a3-4838-8663-ce07d3e88353 --host 127.0.0.1
```

Example Response

```
{
  "id": "2ecc1922-57e2-416d-9023-95720419fa25",
  "uri": "kylo:kylo:connectionmgmt:connections:syslog-conn-1-2ecc1922-57e2-416d-9023-95720419fa25",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-05T04:53:58.893569708Z",
  "updatedAt": "2022-05-05T04:53:58.890021892Z",
  "service": "syslog",
  "category": "log-forwarders",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "syslog-conn-1",
  "description": "conn-description",
  "host": "127.0.0.1",
  "port": 1234,
  "syslog_params": {
    "transport": "tcp",
    "message_format": "rfc5424"
  }
}
```

Deleting a Syslog Connection

To delete a Syslog connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder syslog delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt log-forwarder syslog delete --id 9d3af367-d4a3-4838-8663-ce07d3e88353
```

Example Response

There will be no response if the Syslog connection is deleted successfully.

Getting List of Syslog Connections

To list all the Syslog connections, run:

Syntax

```
ksctl connectionmgmt log-forwarder syslog list
```

Example Request

```
ksctl connectionmgmt log-forwarder syslog list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "2ecc1922-57e2-416d-9023-95720419fa25",
      "uri": "kylo:kylo:connectionmgmt:connections:syslog-conn-1-2ecc1922-57e2-416d-9023-95720419fa25",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2022-05-05T04:53:58.893569708Z",
      "updatedAt": "2022-05-05T04:53:58.890021892Z",
      "service": "syslog",
      "category": "log-forwarders",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "syslog-conn-1",
      "description": "conn-description",
      "host": "127.0.0.1",
      "port": 1234,
      "syslog_params": {
        "transport": "tcp",
        "message_format": "rfc5424"
      }
    }
  ]
}
```

Testing an Existing Syslog Connection

Note

Test Connection verifies the status of the server certificate if it is revoked or not by using the [Certificate Revocation Lists \(CRLs\)](#). If the certificate is found to be revoked, the test connection fails. If an error occurs during the evaluation of the certificate's revocation status, the connection remains in a soft fail state and test connection is allowed to continue.

To test an existing Syslog connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder syslog test --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt log-forwarder syslog test --id 9d3af367-d4a3-4838-8663-ce07d3e88353
```

Example Response

```
{
  "connection_ok": true
}
```

Testing a New Syslog Connection

Note

Test Connection verifies the status of the server certificate if it is revoked or not by using the [Certificate Revocation Lists \(CRLs\)](#). If the certificate is found to be revoked, the test connection fails. If an error occurs during the evaluation of the certificate's revocation status, the connection remains in a soft fail state and test connection is allowed to continue.

To test the parameters of a New Syslog connection, run:

Syntax

```
ksctl connectionmgmt log-forwarder syslog test --host <host> --port <port> --transport <transport-protocol> --ca-cert <ca-cert>
```

Example Request

```
ksctl connectionmgmt log-forwarder syslog test --host 127.0.0.1 --port 1234 --transport tcp
```

Example Response

```
{
  "connection_ok": true
}
```

Oracle Cloud Infrastructure

Oracle Cloud Infrastructure connections to the CipherTrust Manager can be configured using the following:

- [Managing Oracle Cloud Infrastructure Connections using GUI](#)
- [Managing Oracle Cloud Infrastructure Connections using ksctl](#)

Managing Oracle Cloud Infrastructure Connections using GUI

Note

- The CipherTrust Manager should be reachable over a static IP address from OCI when creating a connection to your OCI account for Oracle External resources. This static IP address of the CipherTrust Manager must be added to the SAN field of the web server certificate.
- Use Oracle FastConnect to connect the CipherTrust Manager to the OCI to minimize network latency.

To configure an Oracle Cloud Infrastructure connection:

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.
4. On the Add Connection screen, select category as **Cloud**.
5. Select Select Cloud Type as **Oracle Cloud Infrastructure** and click **Next**.
6. Specify connection **Name** and **Description** and click **Next**.
7. Configure the below parameters.
 - **Tenancy OCID**: OCID of the tenancy.
 - **User OCID**: OCID of the user.
 - **Region**: An Oracle Cloud Infrastructure region.
 - **Fingerprint**: Fingerprint of the public key added to this user.
 - **Key File**: Private key file for the Oracle Cloud Infrastructure connection in the PEM format. Either upload the key file or paste the file content.
 - **File Upload**: Select and click **Upload Certificate** to upload the key file from your machine.

- **Text:** Select and paste the certificate content in the text field.
 - **Passphrase:** Passphrase of the encrypted key file.
8. Click **Test Credentials** to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.
 9. Click **Next** to move to the **Add Products** screen of the Add Connection wizard.

Note

Currently, the only product supported for Oracle Cloud Infrastructure connection is Cloud Key Manager.

Managing Oracle Cloud Infrastructure Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete an Oracle Cloud Infrastructure connection
- List all Oracle Cloud Infrastructure connections
- Test an existing Oracle Cloud Infrastructure connection
- Test parameters for an Oracle Cloud Infrastructure Connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Name of the connection.
description	Optional	Connection description.
products	Optional	List of products.
tenancy-ocid	Mandatory	OCID of the tenancy.
user-ocid	Mandatory	OCID of the user.
oci-region	Mandatory	An Oracle Cloud Infrastructure region.
conn-creds	Mandatory	Credentials of the OCI connection.
fingerprint	Mandatory	Fingerprint of the public key added to this user.
meta	Optional	meta information in json format. This information is provided in pair. For example, --meta <code>{"color\":\"blue\", \"foo\":\"bar\"}</code> .

Parameter	Mandatory/Optional	Description
json-file	Optional	Connection information provided in a JSON file format. Command line parameters will take precedence over values specified in the JSON file.

Creating an Oracle Cloud Infrastructure Connection

Note

- The CipherTrust Manager should be reachable over a static IP address from OCI when creating a connection to your OCI account for Oracle External resources. This static IP address of the CipherTrust Manager must be added to the SAN field of the web server certificate.
- Use Oracle FastConnect to connect the CipherTrust Manager to the OCI to minimize network latency.

To create an Oracle Cloud Infrastructure connection, run:

Syntax

```
ksctl connectionmgmt oci create --name <connection-name> --products <product-name>
s> --user-ocid <user-ocid> --tenancy-ocid <tenancy-ocid> --oci-region <region> --
fingerprint <fingerprint> --conn-creds <key_file, pass_phrase-in-json-format>
```

Example Request

```
ksctl connectionmgmt oci create --name oci-connection --products cckm --user-oci
d ocid1.user.oc1..asdaaaat2x4wy2jz4iat56kk7kqbzcevwyasdy2bquujjhwcstmcfvbfq --
tenancy-ocid ocid1.tenancy.oc1..7777aaaadixb52q2mvlsn634ql577776hb2vg7audpd4d4mcf
5zluymff644 --oci-region ap-sydney-1 --fingerprint c4:a9:89:47:21:11:11:ac:c4:a
9:89:47:21:31:9e --conn-creds conn-cred.json
```

Example Response

```
{
  "id": "666b4d8f-8dec-49c3-860d-33dd4a9cc355",
  "uri": "kylo:kylo:connectionmgmt:connections:oci-connection-666b4d8f-8dec-49c
3-860d-33dd4a9cc355",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-01-19T04:32:15.490282327Z",
  "updatedAt": "2022-01-19T04:32:15.488831158Z",
  "service": "oci",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "oci-connection",
  "products": [
```

```

    "cckm"
  ],
  "user_ocid": "ocidl.user.oc1..asdaaaat2x4wy2jz4iat56kk7kqbzcevwyrasdty2bquuj
jhwcmcfvbfq",
  "tenancy_ocid": "ocidl.tenancy.oc1..7777aaaadixb52q2mvlsn634ql577776hb2vg7aud
pd4d4mcf5zluymff644",
  "fingerprint": "c4:a9:89:47:21:11:11:ac:c4:a9:89:47:21:31:9e",
  "region": "ap-sydney-1"
}

```

conn-cred.json

```

{
  "key_file": "-----BEGIN RSA PRIVATE KEY-----\nMIICXAIBAAKBgQC+abfqs+wQOmoLn4
wldRSty/6fLubJ/JfuBZVV+GMI//Oa/UT\n+s4ZNqn1fta42oN4uIKwsBdnJ4CaoHv5dX6phGirYh3PYT
sC9azdW2wgJ/WCiin8\nkdGNfhPDirOe4TwpzczkP870EEfDS/O3f78x1ubRuIpagzJQv2XTT8QYP+wIDA
QAB\nAoGAERTuaqe/jbWx0VlgefQK5ELVkmhyavlXYcMEZQJGksfKKCQGqAyGFYr6Ghofe\nnwrzfEvmAx
8NuzbRVxmUEFV+C5Uc3uh+sX9qwikfFszTjwJNACHADO3EhPKmMDK3\nkEtBH6edcKa4cJ91NHPJuDpti
yUZdVH7WVzuKrjo4mzFkAECQQDp4fMwnjwyJPGk\nX0dU/0bA69hPCQK5MhVSvVD8fzp6usbeSA/EZSu5
FPnfJT/9f/BVUZ3h0/2WqO1l\nnuhUgnKU5AkEA0GtqDAmTRB5YzRMnmA/QGrCEBkBWdnkXKXZS3Svp19X
HxF9AAQjq\nnyU0YRNHXaxdowWc64tFy2cP4Z78fQ4ry0wJABNe93lrYaj1j14C1jGgAwgvGhbrV\nnCJql
4GG1JJVJ07K8XWvmj618m0d4xpaR3aDhjBK1jzCBhrYWvE1/FH7J2QJAP5Jj\nn+GP7TW3MPFE5ZIJ+QYX
R325EcUKiM/1pbRj17OXCvz2OckJcCya+3k77XCj5xPRN\nn291zIMVLwalkSd/aDQJBAJNbm0RQ4gjj71
0aEbjYnGZlKHtbPP6zD6J/Jiyo+mgZ\nnrvr26CvjtflGi/a56QC6Kd8hSRjeM03yTOvqu9+1TW
Y=\n-----END RSA PRIVATE KEY-----",
  "pass_phrase": "password"
}

```

Getting Details of an Oracle Cloud Infrastructure Connection

To get details of an Oracle Cloud Infrastructure connection, run:

Syntax

```
ksctl connectionmgmt oci get --id <connection-name/id>
```

Example Request

```
ksctl connectionmgmt oci get --id oci-connection
```

Example Response

```

{
  "id": "666b4d8f-8dec-49c3-860d-33dd4a9cc355",
  "uri": "kylo:kylo:connectionmgmt:connections:oci-connection-666b4d8f-8dec-49c
3-860d-33dd4a9cc355",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-01-19T04:32:15.490282Z",
  "updatedAt": "2022-01-19T04:32:15.488831Z",
  "service": "oci",
  "category": "cloud",
  "last_connection_ok": null,

```

```

    "last_connection_at": "0001-01-01T00:00:00Z",
    "name": "oci-connection",
    "products": [
        "cckm"
    ],
    "user_ocid": "ocid1.user.oc1..asdaaaat2x4wy2jz4iat56kk7kqbzcevwyrasdty2bquuj
jhwcmstmcfvbfq",
    "tenancy_ocid": "ocid1.tenancy.oc1..7777aaaadixb52q2mvlsn634ql577776hb2vg7aud
pd4d4mcf5zluymff644",
    "fingerprint": "c4:a9:89:47:21:11:11:ac:c4:a9:89:47:21:31:9e",
    "region": "ap-sydney-1"
}

```

Updating an Oracle Cloud Infrastructure Connection

To update an Oracle Cloud Infrastructure connection, run:

Syntax

```

ksctl connectionmgmt oci modify --id <connection-name/id> --products <product-nam
es> --user-ocid <user-ocid> --tenancy-ocid <tenancy-ocid> --oci-region <region>
--fingerprint <fingerprint> --conn-creds <key_file,pass_phrase-in-json-format> --
meta <key:values>

```

Example Request

```

ksctl connectionmgmt oci modify --id oci-connection --user-ocid ocid2.user.oc2..a
sdaaaaktch502jz4iat56kk7kqbzcevk45kugv0ienuuujjhwcmstmcfvbfq

```

Example Response

```

{
    "id": "666b4d8f-8dec-49c3-860d-33dd4a9cc355",
    "uri": "kylo:kylo:connectionmgmt:connections:oci-connection-666b4d8f-8dec-49c
3-860d-33dd4a9cc355",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2022-01-19T04:32:15.490282Z",
    "updatedAt": "2022-01-19T04:40:36.311287549Z",
    "service": "oci",
    "category": "cloud",
    "last_connection_ok": null,
    "last_connection_at": "0001-01-01T00:00:00Z",
    "name": "oci-connection",
    "products": [
        "cckm"
    ],
    "user_ocid": "ocid2.user.oc2..asdaaaaktch502jz4iat56kk7kqbzcevk45kugv0ienuuuj
jhwcmstmcfvbfq",
    "tenancy_ocid": "ocid1.tenancy.oc1..7777aaaadixb52q2mvlsn634ql577776hb2vg7aud
pd4d4mcf5zluymff644",
    "fingerprint": "c4:a9:89:47:21:11:11:ac:c4:a9:89:47:21:31:9e",
    "region": "ap-sydney-1"
}

```

Deleting an Oracle Cloud Infrastructure Connection

To delete an Oracle Cloud Infrastructure connection, run:

Syntax

```
ksctl connectionmgmt oci delete --id <connection-name/id>
```

Example Request

```
ksctl connectionmgmt oci delete --id oci-connection
```

Example Response

There will be no response if Oracle Cloud Infrastructure Connection is deleted successfully.

Getting List of Oracle Cloud Infrastructure Connections

To list all the Oracle Cloud Infrastructure connections, run:

Syntax

```
ksctl connectionmgmt oci list
```

Example Request

```
ksctl connectionmgmt oci list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "666b4d8f-8dec-49c3-860d-33dd4a9cc355",
      "uri": "kylo:kylo:connectionmgmt:connections:oci-connection-666b4d8f-8dec-49c3-860d-33dd4a9cc355",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2022-01-19T04:32:15.490836Z",
      "updatedAt": "2022-01-19T04:40:36.312949Z",
      "service": "oci",
      "category": "cloud",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "oci-connection",
      "products": [
        "cckm"
      ],
      "user_ocid": "ocid2.user.oc2..asdaaaaktchn502jz4iat56kk7kqbzcevk45kugv0ienuujjhwcstmcfvbfq",
    }
  ]
}
```



```

        "tenancy_ocid": "ocid1.tenancy.oc1..7777aaaadixb52q2mvlsn634ql577776hb2vg7audpd4d4mcf5zluymff644",
        "fingerprint": "c4:a9:89:47:21:11:11:ac:c4:a9:89:47:21:31:9e",
        "region": "ap-sydney-1"
    }
]
}

```

Testing an Existing Oracle Cloud Infrastructure Connection

To test an existing Oracle Cloud Infrastructure connection, run:

Syntax

```
ksctl connectionmgmt oci test --id <connection-name/id>
```

Example Request

```
ksctl connectionmgmt oci test --id oci-connection
```

Example Response

```

{
  "connection_ok": true
}

```

Testing Parameters for an Oracle Cloud Infrastructure Connection

To test parameters for an Oracle Cloud Infrastructure connection, run:

Syntax

```
ksctl connectionmgmt oci test --user-ocid <user-ocid> --tenancy-ocid <tenancy-ocid> --oci-region <region> --fingerprint <fingerprint> --conn-creds <key_file,password-in-json-format>
```

Example Request

```
ksctl connectionmgmt oci test --user-ocid ocid1.user.oc1..asdaaaaat2x4wy2jz4iat56kk7kqbzcevwyrsdty2bquujjhwcmcfvbfq --tenancy-ocid ocid1.tenancy.oc1..7777aaaadixb52q2mvlsn634ql577776hb2vg7audpd4d4mcf5zluymff644 --oci-region ap-sydney-1 --fingerprint c4:a9:89:47:21:11:11:ac:c4:a9:89:47:21:31:9e --conn-creds conn-cred.json
```

Example Response

```

{
  "connection_ok": true
}

```

conn-cred.json

```
{
  "key_file": "-----BEGIN RSA PRIVATE KEY-----\nMIICXAIBAAKBgQC+abfqs+wQOmoLnF4\nwldRSty/6fLubJ/JfuBZVV+GMI//Oa/UT\n+s4ZNqn1fta42oN4uIKwsBdnJ4CaoHv5dX6phGirYh3PYT\nsC9azdW2wgJ/WCiin8\n\\nkdGNfhPDirOe4TwpczkP870EEfDS/O3f78x1ubRuIpagzJQv2XTT8QYP+wIDA\nQAB\n\\nAoGAERtuage/jbWx0VlgefQK5ELVkmhyavlXYcMEZQJGksfKKCQGqAyGFYr6Ghofe\n\\nwrzfEvmAxF\n8NuzbRVxMUEFV+C5Uc3uh+sX9qwikfFszTjwJNACHADO3EhPKmMDK3\n\\nkEtBH6edcKa4cJ91NHPJuDpti\nyUZdVH7WVzuKrj\no4mzFkAECQQDp4fMwnjwyJPGk\n\\nX0dU/0bA69hPCQK5MhVSvVD8fz\np6usbeSA/EZSu5\nFPnfJT/9f/BVUZ3h0/2WqO1l\n\\nuhUgnKU5AkEA0GtqDAmTRB5YzRMnmA/QGrCEBkBWdnkXKXZS3Svp19X\nHxF9AAQjq\n\\nyU0YRNHXaxdowWc64tFy2cP4Z78fQ4ry0wJABNe93lrYa\nj1j14C1jGgAwgvG\nHbrV\\nCJql\n4GG1JJVJ07K8XWvmj\n618m0d4xpaR3aDhjBK1jzCBhrYWvE1/FH7J2QJAP5Jj\n\\n+GP7TW3MPFE5ZIJ+QYX\nR325EcUKiM/1pbRj\n170XCVz2OckJcCya+3k77XCj5xPRN\n\\n291zIMVLwalkSd/aDQJBAJNbm0RQ4gjj71\n0aEbjYnGZlKHtbPP6zD6J/Jiyo+mgZ\n\\nrvr26CvjtflGi/a56QC6Kd8hSRjeM03yTOvqu9+1TW\nY=\\n-----END RSA PRIVATE KEY-----",
  "pass_phrase": "password"
}
```

DSM Connection

DSM connections to the CipherTrust Manager can be configured using the following:

- [Managing DSM Connections using GUI](#)
- [Managing DSM Connections using ksctl](#)

Managing DSM Connections using GUI

To manage DSM connections using GUI, perform the following steps:

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.
4. On the Add Connection screen, select category as **Key Manager**.
5. Select Key Manager Type as **DSM Connection** and click **Next**.
6. Specify connection **Name** and **Description** and click **Next**.
7. Configure the below sections.

Add DSM Node

- **Node Hostname/IP** - provide hostname or IP of a DSM node.

Note

If the DSM hostname cannot be resolved, then a DNS entry must be added under **Admin Settings > DNS Hosts**.

- **Certificate** - upload the DSM node certificate. This is the [DSM server certificate](#) (for example, `DSM.cer`) you downloaded to your local machine.

To add multiple nodes in a DSM connection, click **+Add DSM Node**.

Note

Nodes must be from the same DSM cluster.

For Authentication

- **Username** - username of the DSM server.
- **Password** - password of the DSM server.
- **Domain ID** - provide domain Id if DSM user is restricted to a domain. It is an optional field.

Note

While creating a connection to DSM on the CipherTrust Manager, if the user is a local domain admin in the DSM, then a domain ID is required.

Let's assume, user "U1" manages/owns the domain "D1".

To get the domain id, follow this process:

1. "U1" requests the System Administrator of DSM to provide the domain id for "D1".
2. System Administrator calls the `/dsm/v1/domains` API to fetch the list of domains. This API returns details of all domains including their IDs.
3. System Administrator finds the domain ID of domain "D1" and provides it to "U1".

8. Click the **Test Credentials** button to check whether the connection is configured correctly. If the test is successful, the status is `OK` else the status is `Fail`.
9. Click **Next** to move to the [Add Products](#) screen of the Add Connection wizard.

Managing DSM Connections using ksctl

The following operations can be performed:

- Create/get/update/delete a DSM connection
- List all DSM connections
- Test an existing DSM connection
- Test parameters for a DSM connection
- Add/get/update/delete a node in an existing DSM connection
- List all nodes in a DSM connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Unique connection name.
description	Optional	Connection description.
username	Mandatory	Username for accessing DSM server.
conn-password	Mandatory	Password of DSM server.
dsm-domain-id	Optional	If DSM user is restricted to a domain, provide domain id.
json-file	Optional	Connection information passed in JSON format through a file. Parameters passed via the command line will override values in the JSON file. For example, --json-file
nodes-json-file	Optional	List of server nodes in a JSON file (For example, 'ServerNodes.json')
hostname	Mandatory	Hostname of the DSM Server.
server_certificate	Mandatory	SSL certificate of DSM Server for TLS communication.
products	Optional	Attribute inside the user object that contains the username to login with.

Creating a DSM Connection

To create a DSM connection, run:

Syntax

```
ksctl connectionmgmt dsm create --name <Connection-Name> --products <Products-Nam
es> --username <Server-Username> --conn-password <Server-Password> --meta <Key-Va
lues> --dsm-domain-id <Domain-ID> --nodes-json-file <Json-File>
```

Example Request

```
ksctl connectionmgmt dsm create --name dsm-connection --username admin --conn-pas
sword Ssl12345# --nodes-json-file ./dsmnodes.json
```

Example DSM Nodes JSON File

```
[
  {
    "hostname": "test.node1",
    "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIIEETCCAvmgAwIBAgIGCBP
pVT3oMA0GCSqGSIb3DQEBAUAMHMxIzAhBgNVBAMT\nGkNHIENBIFMgb24gc3lzNzgxMDAucWEuY29tMQ
swCQYDVQQLewJRQTESMBAGA1UE\nnChMJVJm9ybWV0cmljMREwDwYDVQQHEwhTYW4gSm9zZTELMakGA1UEC
BMCQ0ExCzAJ\nnBgNVBAYTA1VTMB4XDTIxMDEzMzYwOV0oXDTMxMDEzMzYwOV0wADEYMBYG\nnA1
UEAxMPc3lzNzgxMDAucWEuY29tMQswCQYDVQQLewJRQTESMBAGA1UEChMJVJm9y\nnbWV0cmljMREwDwYDV
QQHEwhTYW4gSm9zZTELMakGA1UECBMCQ0ExCzAJBgNVBAYT\nnAlVTMIIIBIjANBgkqhkiG9w0BAQEFAAOC
AQ8AMIIBCqKCAQEAwVYaXdacb0/g0u41\nnuPtRowUK11roUUNBZ8ileAwPfItkZPo3pSVzxLKngB+7vKS
Ey6t6JfBH/zTzVQxd\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJzm8QuXKw2ANQNKbVWc
EZ\nhpy9C9WQUM2Td9MD0vhzDGCDg1eZnJtqDQdgTM95Xs8RqK/lbGn3x8YLUKiUcTVA\nn4NhTyNVGpnnv
u0i/SPeLZNScRUXI0awqYuavMS3ty5Ooxd78AF3qHiutddvzeR5Ry\nn20w/Xralbk6AuVHTBkaB6CKvSG
ldUi/pFHUxw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\nnHBZCQwIDAQABo4G1MIGyMBYGA1UdJQEB/wQMMAoGC
CsGAQUFBwMBMAwGA1UdEwEB\nn/wQCAAwdGyDVR0PAQH/BAQDAgOoMB0GA1UdDgQWBBT6eagam2IV9s2X
kOesFOxL\nnNRRmZTAnBgNVHSMEIDAegBQLj1Enc9uzLZKw8msCBSyUnRGdKIIGALOnqfHDMBoG\nnA1UdE
QQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzg8AAAABBAatTRVJWRVJfTk9E\nnRTANBgkqhkiG9w0BAQwF
AAOCAQEAoQVMDr8SeS6+F1O3C7ALWp5V0b9FUeab49n7\nnM4fv6yPy+ZC07cVkkG1qde9lIsn5vOHuhdM
JOQeX0EC1FG4zs0gvbQ8/HVcoD7dN\nnfB98nJXm+jOrKOKcNAvGaddtAUX7mDI48mIVrqstjibXxMlgHB
MZ382Ujp5xcOnF\nn1KU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNm3y1Di6hnsijvH6aJf5UVWgYq7Z3\
nFOSPBlo0lyF8Tfavbq8IUpHxYr3eWwV86uJsrV9Pty1xfAy0TqSm9py2+WBIL+ZU\nnCpMRws+XTtAG5j
s7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\n-----END CERTIFICATE-----\n"
  },
  {
    "hostname": "test.node2",
    "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIIEETCCAvmgAwIBAgIGCBP
pVT3oMA0GCSqGSIb3DQEBAUAMHMxIzAhBgNVBAMT\nGkNHIENBIFMgb24gc3lzNzgxMDAucWEuY29tMQ
swCQYDVQQLewJRQTESMBAGA1UE\nnChMJVJm9ybWV0cmljMREwDwYDVQQHEwhTYW4gSm9zZTELMakGA1UEC
BMCQ0ExCzAJ\nnBgNVBAYTA1VTMB4XDTIxMDEzMzYwOV0oXDTMxMDEzMzYwOV0wADEYMBYG\nnA1
UEAxMPc3lzNzgxMDAucWEuY29tMQswCQYDVQQLewJRQTESMBAGA1UEChMJVJm9y\nnbWV0cmljMREwDwYDV
QQHEwhTYW4gSm9zZTELMakGA1UECBMCQ0ExCzAJBgNVBAYT\nnAlVTMIIIBIjANBgkqhkiG9w0BAQEFAAOC
AQ8AMIIBCqKCAQEAwVYaXdacb0/g0u41\nnuPtRowUK11roUUNBZ8ileAwPfItkZPo3pSVzxLKngB+7vKS
Ey6t6JfBH/zTzVQxd\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJzm8QuXKw2ANQNKbVWc
EZ\nhpy9C9WQUM2Td9MD0vhzDGCDg1eZnJtqDQdgTM95Xs8RqK/lbGn3x8YLUKiUcTVA\nn4NhTyNVGpnnv
u0i/SPeLZNScRUXI0awqYuavMS3ty5Ooxd78AF3qHiutddvzeR5Ry\nn20w/Xralbk6AuVHTBkaB6CKvSG
ldUi/pFHUxw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\nnHBZCQwIDAQABo4G1MIGyMBYGA1UdJQEB/wQMMAoGC
CsGAQUFBwMBMAwGA1UdEwEB\nn/wQCAAwdGyDVR0PAQH/BAQDAgOoMB0GA1UdDgQWBBT6eagam2IV9s2X
kOesFOxL\nnNRRmZTAnBgNVHSMEIDAegBQLj1Enc9uzLZKw8msCBSyUnRGdKIIGALOnqfHDMBoG\nnA1UdE
QQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzg8AAAABBAatTRVJWRVJfTk9E\nnRTANBgkqhkiG9w0BAQwF
AAOCAQEAoQVMDr8SeS6+F1O3C7ALWp5V0b9FUeab49n7\nnM4fv6yPy+ZC07cVkkG1qde9lIsn5vOHuhdM
JOQeX0EC1FG4zs0gvbQ8/HVcoD7dN\nnfB98nJXm+jOrKOKcNAvGaddtAUX7mDI48mIVrqstjibXxMlgHB
MZ382Ujp5xcOnF\nn1KU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNm3y1Di6hnsijvH6aJf5UVWgYq7Z3\
```

```
nFOSPBlo0lyF8Tfavbq8IUphXyR3eWwV86uJsrv9Pty1xfAy0TqSm9py2+WBIL+ZU\nCpMRws+XTtAG5j
s7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\n-----END CERTIFICATE-----\n"
}
]
```

Example Response

```
{
  "id": "d873bac7-adeb-43a9-b3cf-2e76cbd40bd2",
  "uri": "kylo:kylo:connectionmgmt:connections:dsm-connection-d873bac7-ade
f-43a9-b3cf-2e76cbd40bd2",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-02-03T10:36:52.944110478Z",
  "updatedAt": "2021-02-03T10:36:52.941941401Z",
  "service": "DSM",
  "category": "KeyManager",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "dsm-connection",
  "username": "admin",
  "nodes": [
    {
      "id": "7a523daa-aa94-4b11-b31b-c945853663e4",
      "uri": "kylo:kylo:connectionmgmt:dsm-node:dsm-connectio
n-7a523daa-aa94-4b11-b31b-c945853663e4",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2021-02-03T10:36:52.946654385Z",
      "hostname": "test.node1",
      "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIIEE
TCCAvmgAwIBAgIGCBPpVT3oMA0GCSqGSIb3DQEBAUAMHMxIzAhBgNVBAMT\nGkNHIENBIFMgb24gc3lz
NzgxMDAucWEuY29tMQswCQYDVQQLLEwJRQTESMBAGA1UE\nChMjVWm9ybWV0cm1jMREwDwYDVQQHEWhTYW4
gSm9zZTELMakGA1UECBMCQ0ExCzAJ\nBgnVBAYTA1VTMB4XDTIxMDEzMzYwOV0XDTMxMDEzMzYwOV0w
aDEYMBYG\nA1UEAxMPC3lzNzgxMDAucWEuY29tMQswCQYDVQQLLEwJRQTESMBAGA1UEChMjVWm9y
nbWV0cm1jMREwDwYDVQQHEWhTYW4gSm9zZTELMakGA1UECBMCQ0ExCzAJBgNVBAYT\nA1VMTMIIIBIjANBg
kqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAWVYaXdacb0/g0u41\nnuPtRowUK11roUUNBZ8ileAwPfitkZ
Po3pSVzxLKngB+7vKSEy6t6JfBH/zTzVQxd\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJ
zm8QuXKw2ANQNKbVWcEZ\nnhpy9C9WQUM2Td9MD0vhzDGCDg1eZnJtqDQdgTM95Xs8RqK/lbGn3x8YLUKi
UcTVA\n4NhTyNVGpnu0i/SPeLZNScRUXI0awqYuavMS3ty5Ooxd78AF3qHiutddvzeR5Ry\nn20w/Xral
bk6AuVHTbKaB6CKvSG1dUi/pFHUxw+49Fn2+gVPP0RYXGEVwM5d6wJ3L\nnHBZCQwIDAQABo4G1MIGYMBY
GA1UdJQEB/wQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB\n/n/wQCAAawDgYDVR0PAQH/BAQDAgOoMB0GA1UdDg
QWBBT6eagam2IV9s2XkOesFOxL\nnNRRmZTANBgNVHSMEIDAegBQLj1Enc9uzLzKw8msCBSyUnRGdKIIGA
LONqfHDMBoG\nA1UdEQQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzg8AAAABBAAtTRVJWRVJfTk9E\nnRT
ANBgkqhkiG9w0BAQwFAAOCAQEAOQVMDr8SeS6+F1O3C7ALWp5V0b9FUeab49n7\nnM4fV6yPy+ZC07cVkk
G1qde9lIsn5vOHuhdMJOQeX0EC1FG4zs0gvbQ8/HVcoD7dN\nnfB98nJXm+jOrKOKcNAvgaddtAUX7mDI4
8mIVrqstjibXxMlgHBMZ382Ujp5xcOnF\nnlKU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNm3y1Di6hnsi
jvH6aJf5UVWgYq7Z3\nnFOSPBlo0lyF8Tfavbq8IUphXyR3eWwV86uJsrv9Pty1xfAy0TqSm9py2+WBIL
+ZU\nCpMRws+XTtAG5js7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\n-----END CERTIFICAT
E-----\n",
      "last_connection_ok": null,
      "connection_id": "d873bac7-adeb-43a9-b3cf-2e76cbd40bd2"
    },
    {
      "id": "bf333cb2-ad27-45ef-b1ba-5a85408e4141",
      "uri": "kylo:kylo:connectionmgmt:dsm-node:dsm-connection-
bf333cb2-ad27-45ef-b1ba-5a85408e4141",
      "account": "kylo:kylo:admin:accounts:kylo",

```

```

      "createdAt": "2021-02-03T10:36:52.948553989Z",
      "hostname": "test.node2",
      "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIEE
TCCAvmgAwIBAgIGCBPpVT3oMA0GCSqGSIb3DQEBAUAMHMxIzAhBgNVBAMT\nGknNHIENBIFMgb24gc3lz
NzgxmDAucWEuY29tMQswCQYDVQQLLEwJRQTESMBAGA1UE\nnChmJVm9ybWV0cm1jMREwDwYDVQQHEwhTYW4
gSm9zZTELMakGA1UECBMCQ0ExCzAJ\nnBgNVBAYTA1VTMB4XDITixMDEzMzYwOVVoXDTMxMDEzMzYwOVowaDEYMBYg\nnA1UEAxMPc3lzNzgxmDAucWEuY29tMQswCQYDVQQLLEwJRQTESMBAGA1UEChmJVm9y\nnbWV0cm1jMREwDwYDVQQHEwhTYW4gSm9zZTELMakGA1UECBMCQ0ExCzAJBgNVBAYT\nnA1VTMIIBIjANBg
kqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAWVYaXdacB0/g0u41\nnuPtRowUK1lroUUNBZ8ileAwPfItkZ
Po3pSVzxLKnqB+7vKSEy6t6JfBH/zTzVQxd\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJ
zm8QuxKw2ANQNKBVWcEZ\nnhpy9C9WQUM2Td9MD0vhzDGCdglEzNjtgDQdgTM95Xs8RqK/1bGn3x8YLUKi
UcTVA\nn4NhTyNVGpnuv0i/SPeLZNScRUXI0awqYuavMS3ty50oxd78AF3qHiutddvzeR5Ry\nn20w/Xral
bk6AuVHtBkaB6CKvSG1dUi/pFHuXw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\nnHBZCQwIDAQABo4G1MIGYMBY
GA1UdJQEB/wQMMAOGCCsGAQUFBwMBMAwGA1UdEwEB\nn/wQMAAwDgYDVR0PAQH/BAQDAgOoOMB0GA1UdDg
QWBBT6eagam2IV9s2XkOesFOxL\nnNRRmZTAnBgNVHSMEIDAegBQLjlEnc9uzLzKw8msCBSyUnRGdKIIGA
LONqfHDMBoG\nnA1UdEQQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzg8AAAABBBatTRVJWRVJfTk9E\nnRT
ANBgkqhkiG9w0BAQwFAAOCAQEAOQVMDr8SeS6+F1O3C7ALWp5V0b9FUeab49n7\nnM4fV6yPy+ZC07cVkg
G1qde9lIsn5vOHuhdMJOQeX0EC1FG4zs0gvbQ8/HVcoD7dN\nnfB98nJXm+jOrKOKcNAvGaddtAUX7mDI4
8mIVrqstjibXxMlgHBMZ382Ujp5xcOnF\nnlKU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNm3y1Di6hnsi
jvH6aJf5UVWgYq7Z3\nnFOSPBlo0lyF8Tfavbq8IUphXyR3eWwV86uJsrV9Pty1xfAy0TqSm9py2+WBI
L+ZU\nnCpMRws+XTtAG5js7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\nn-----END CERTIFICAT
E-----\nn",
      "last_connection_ok": null,
      "connection_id": "d873bac7-adeb-43a9-b3cf-2e76cbd40bd2"
    }
  ]
}

```

Getting Details of a DSM Connection

To get details of a DSM connection, run:

Syntax

```
ksctl connectionmgmt dsm get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt dsm get --id dsm-connection
```

Example Response

```

{
  "id": "d873bac7-adeb-43a9-b3cf-2e76cbd40bd2",
  "uri": "kylo:kylo:connectionmgmt:connections:dsm-connection-d873bac7-adeb-43a9-b3cf-2e76cbd40bd2",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-02-03T10:36:52.94411Z",
  "updatedAt": "2021-02-03T10:36:52.941941Z",
  "service": "DSM",
  "category": "KeyManager",

```

```

    "last_connection_ok": null,
    "last_connection_at": "0001-01-01T00:00:00Z",
    "name": "dsm-connection",
    "username": "admin",
    "nodes": [
        {
            "id": "7a523daa-aa94-4b11-b31b-c945853663e4",
            "uri": "kylo:kylo:connectionmgmt:dsm-node:dsm-connection-7a523daa-aa94-4b11-b31b-c945853663e4",
            "account": "kylo:kylo:admin:accounts:kylo",
            "createdAt": "2021-02-03T10:36:52.946654Z",
            "hostname": "test.node1",
            "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIEETCCAvmgAwIBAgIGCBPpVT3oMA0GCSqGSIb3DQEBAQUAMHxIzAhBgNVBAMT\nNzgxMDAucWEuY29tMQswCQYDVQQL\nEwJkZm9zZTEuMAkGA1UECBMCMQ0ExCzAJ\nnBgNVBAYTA1VTMB4XDTEwMDEzMzYwOV\nOwXDTMxMDEzMzYwOVowDEYMBYg\nnA1UEAxMPC3lzNzgxMDAucWEuY29tMQswCQYDVQQL\nEwJkZm9zZTEuMAkGA1UECBMCMQ0ExCzAJ\nBgNVBAYT\nnA1VTMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMI\nIBCgKAQEAAwVYaXdacb0/g0u41\nnuPtRowUK11roUUNBZ8ileAwPfItkZ\nPo3pSVzxLKnqB+7vKSEy6t6JfBH/zTzVQxd\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2X\nRuv9yGxJzm8QuxKw2ANQNKbVWcEZ\nnhpy9C9WQUM2Td9MD0vhzDGCDg1eZ\nNjtgDQdgTM95Xs8RqK/lbGn3x8YLUKiUcTVA\nn4NhTyNVGpnu0i/SPeLZNScRUXI0awqYuavMS3ty5\nOoxd78AF3qHiutddvzeR5Ry\nn20w/Xralbk6AuVHtBkaB6CKvSG1dUi/pFH\nuXw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\nnHBZCQwIDAQABo4G1MIGYMBYGA1UdJQEB\n/wQMMAAoGCCsGAQUFBwMBMAwGA1UdEwEB\nn/wQCMAAwDgYDVR0PAQH/BAQDAgOoMB0GA1UdDg\nQWBBT6eagam2IV9s2XkOesFOxL\nnNRRmZTAnBgNVHSMEIDAeGBQLj1Enc9uzLzKw8ms\nCBSyUnRGdKIIGA\nLONqfHDMBoG\nnA1UdEQQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgc\nrzg8AAAABBA\nTRVJWRVJfTk9E\nnRTANBgkqhkiG9w0BAQwFAAOCAQEAAoQVMDr8SeS6+F\n1O3C7ALWp5V0b9FUeab49n7\nnM4fv6yPy+ZC07cVkkG1qde9l\nIsn5vOHuhdMJOQeX0EC1FG4zs0gvbQ8/HVcoD7dN\nnfB98nJXm+jOrKOKcNAvGaddtAUX7mDI48mIVr\nqstjibXxMlgHBMZ382Ujp5xcOnF\nnlKU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNm\nx3y1Di6hnsijvH6aJf5UVWgYq7Z3\nnFOSPBlo0lyF8Tfavbq8IUpHxYr3eWwV86uJsrV9\nPty1xfAy0TqSm9py2+WBI\nL+ZU\nnCpMRws+XTtAG5js7OkJbgKvoVf+GtWpSw4/OSF4Pg\nGnfzoZ0YA==\n\n-----END CERTIFICATE\n\n",
            "last_connection_ok": null,
            "connection_id": "d873bac7-adeb-43a9-b3cf-2e76cbd40bd2"
        },
        {
            "id": "bf333cb2-ad27-45ef-b1ba-5a85408e4141",
            "uri": "kylo:kylo:connectionmgmt:dsm-node:dsm-connection-bf333cb2-ad27-45ef-b1ba-5a85408e4141",
            "account": "kylo:kylo:admin:accounts:kylo",
            "createdAt": "2021-02-03T10:36:52.948554Z",
            "hostname": "test.node2",
            "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIEETCCAvmgAwIBAgIGCBPpVT3oMA0GCSqGSIb3DQEBAQUAMHxIzAhBgNVBAMT\nNzgxMDAucWEuY29tMQswCQYDVQQL\nEwJkZm9zZTEuMAkGA1UECBMCMQ0ExCzAJ\nnBgNVBAYTA1VTMB4XDTEwMDEzMzYwOV\nOwXDTMxMDEzMzYwOVowDEYMBYg\nnA1UEAxMPC3lzNzgxMDAucWEuY29tMQswCQYDVQQL\nEwJkZm9zZTEuMAkGA1UECBMCMQ0ExCzAJ\nBgNVBAYT\nnA1VTMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMI\nIBCgKAQEAAwVYaXdacb0/g0u41\nnuPtRowUK11roUUNBZ8ileAwPfItkZ\nPo3pSVzxLKnqB+7vKSEy6t6JfBH/zTzVQxd\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2X\nRuv9yGxJzm8QuxKw2ANQNKbVWcEZ\nnhpy9C9WQUM2Td9MD0vhzDGCDg1eZ\nNjtgDQdgTM95Xs8RqK/lbGn3x8YLUKiUcTVA\nn4NhTyNVGpnu0i/SPeLZNScRUXI0awqYuavMS3ty5\nOoxd78AF3qHiutddvzeR5Ry\nn20w/Xralbk6AuVHtBkaB6CKvSG1dUi/pFH\nuXw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\nnHBZCQwIDAQABo4G1MIGYMBYGA1UdJQEB\n/wQMMAAoGCCsGAQUFBwMBMAwGA1UdEwEB\nn/wQCMAAwDgYDVR0PAQH/BAQDAgOoMB0GA1UdDg\nQWBBT6eagam2IV9s2XkOesFOxL\nnNRRmZTAnBgNVHSMEIDAeGBQLj1Enc9uzLzKw8ms\nCBSyUnRGdKIIGA\nLONqfHDMBoG\nnA1UdEQQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgc\nrzg8AAAABBA\nTRVJWRVJfTk9E\nnRTANBgkqhkiG9w0BAQwFAAOCAQEAAoQVMDr8SeS6+F\n1O3C7ALWp5V0b9FUeab49n7\nnM4fv6yPy+ZC07cVkkG1qde9l\nIsn5vOHuhdMJOQeX0EC1FG4zs0gvbQ8/HVcoD7dN\nnfB98nJXm+jOrKOKcNAvGaddtAUX7mDI48mIVr\nqstjibXxMlgHBMZ382Ujp5xcOnF\nnlKU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNm\nx3y1Di6hnsi

```



```
jvH6aJf5UVWgYq7Z3\nFOSPBlo0lyF8Tfavbq8IUphXyR3eWwV86uJsrV9Pty1xfAy0TqSm9py2+WBI
L+ZU\nCpMRws+XTtAG5js7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\n-----END CERTIFICAT
E-----\n",
    "last_connection_ok": null,
    "connection_id": "d873bac7-adeb-43a9-b3cf-2e76cbd40bd2"
  }
]
}
```

Updating a DSM Connection

To update a DSM connection, run:

Syntax

```
ksctl connectionmgmt dsm modify --id <Connection-Name/ID> --products <Products-Names> --username <Server-Username> --conn-password <Server-Password> --meta <Key:Values>
```

Example Request

```
ksctl connectionmgmt dsm modify --id dsm-cli-1 --username myadmin
```

Example Response

```
{
  "id": "110b0166-6320-4de2-bbd4-c037665616cf",
  "uri": "kylo:kylo:connectionmgmt:connections:dsm-cli-1-110b0166-6320-4de2-bbd4-c037665616cf",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-02-03T08:22:58.459353Z",
  "updatedAt": "2021-02-03T10:40:41.242490229Z",
  "service": "DSM",
  "category": "KeyManager",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "dsm-cli-1",
  "meta": "",
  "username": "myadmin"
}
```

Deleting a DSM Connection

To delete a DSM connection, run:

Syntax

```
ksctl connectionmgmt dsm delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt dsm delete --id dsm-cli-1
```

There will be no response if DSM connection is deleted successfully.

Getting List of DSM Connections

To list all the DSM connections, run:

Syntax

```
ksctl connectionmgmt dsm list
```

Example Request

```
ksctl connectionmgmt dsm list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "id": "d873bac7-adeb-43a9-b3cf-2e76cbd40bd2",
      "uri": "kylo:kylo:connectionmgmt:connections:dsm-connection-d873bac7-adeb-43a9-b3cf-2e76cbd40bd2",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2021-02-03T10:36:52.94411Z",
      "updatedAt": "2021-02-03T10:36:52.941941Z",
      "service": "DSM",
      "category": "KeyManager",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "dsm-connection",
      "username": "admin"
    },
    {
      "id": "110b0166-6320-4de2-bbd4-c037665616cf",
      "uri": "kylo:kylo:connectionmgmt:connections:dsm-cli-1-110b0166-6320-4de2-bbd4-c037665616cf",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2021-02-03T08:22:58.459353Z",
      "updatedAt": "2021-02-03T08:23:03.937019Z",
```

```

        "service": "DSM",
        "category": "KeyManager",
        "last_connection_ok": null,
        "last_connection_at": "0001-01-01T00:00:00Z",
        "name": "dsm-cli-1",
        "meta": "",
        "username": "kylo_d1_rst"
    }
]
}

```

Testing an Existing DSM Connection

To test an existing DSM connection, run:

Syntax

```
ksctl connectionmgmt dsm test --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt dsm test --id dsm-connection
```

Example Response

```

{
  "status": true,
  "nodes": [
    {
      "hostname": "test.node1",
      "connection_ok": true
    },
    {
      "hostname": "node2",
      "connection_ok": true
    }
  ],
  "admin_type": "SYSTEM_ADMIN",
  "managed_domains": [
    {
      "id": 1,
      "url": "/v1/domains/1",
      "name": "System Domain",
      "description": "Internal system domain",
      "kmipEnabled": false,
      "fingerprintRegistrationDisabled": false,
      "registrationTokenEnabled": false,
      "MoreInfo": true
    }
  ]
}

```

Testing Parameters for a DSM Connection

To test parameters for a DSM connection, run:

Syntax

```
ksctl connectionmgmt dsm test --username <Server-Username> --conn-password <Server-Password> --nodes-json-file <JSON-File>
```

Example Request

```
ksctl connectionmgmt dsm test --username admin --conn-password Ssl12345# --nodes-json-file ./dsmnodes.json
```

Example Response

```
{
  "status": true,
  "nodes": [
    {
      "hostname": "test.node1",
      "connection_ok": true
    },
    {
      "hostname": "test.node2",
      "connection_ok": true
    }
  ],
  "admin_type": "SYSTEM_ADMIN",
  "managed_domains": [
    {
      "id": 1,
      "url": "/v1/domains/1",
      "name": "System Domain",
      "description": "Internal system domain",
      "kmipEnabled": false,
      "fingerprintRegistrationDisabled": false,
      "registrationTokenEnabled": false,
      "MoreInfo": true
    }
  ]
}
```

Adding Node in an Existing DSM Connection

To add node in a existing DSM connection, run:

Syntax

```
ksctl connectionmgmt dsm node add --id <Connection-Name/ID> --hostname <Server-Host-Name> --server-cert-file <Server-Certificate-File>
```

Note

Nodes must be from the same DSM cluster.

Example Request

```
ksctl connectionmgmt dsm node add --id dsm-connection --hostname node2 --server-cert-file ./cert.txt
```

Example Certificate File

```
-----BEGIN CERTIFICATE-----
MIIETCCAvmgAwIBAgIGCBPpVT3oMA0GCSqGSIb3DQEBAUAMHMxIzAhBgNVBAMT
GkNHIENBIFMgb24gc3lzNzgxMDAucWEuY29tMQswCQYDVQQLewJRQTESMBAGA1UE
ChMJVWm9ybWV0cm1jMREwDwYDVQQHEWhTYW4gSm9zZTELMakGA1UECBMCQ0ExCzAJ
BgNVBAYTA1VTMB4XDTIxMDEzMzYwOVowXDTMxMDEzMzYwOVowADEYMBYG
A1UEAxMPc3lzNzgxMDAucWEuY29tMQswCQYDVQQLewJRQTESMBAGA1UEChMJVWm9y
bWV0cm1jMREwDwYDVQQHEWhTYW4gSm9zZTELMakGA1UECBMCQ0ExCzAJBgNVBAYT
A1VTMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAWVYaXdacb0/g0u4l
uPtRowUK1lroUUNBZ8ileAwPfItkZPo3pSVzxLKnqB+7vKSEy6t6JfBH/zTzVQxd
6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJzm8QuxKw2ANQNkbVWcEZ
hpy9C9WQUM2Td9MD0vzhzDGCDg1eZNjtqDQdgTM95Xs8RqK/lbGn3x8YLUKiUcTVA
4NhTyNVGpnvu0i/SPeLZNScRUXI0awqYuavMS3ty5Ooxd78AF3qHiutddvzeR5Ry
20w/Xralbk6AuVHTbkaB6CKvSGldUi/pFHUxw+49Fn2+gVPP0RYXGEwVm5d6wJ3L
HBZCQwIDAQABo4G1MIGYMBYGA1UdJQEB/wQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB
/wQCAAwDgYDVR0PAQH/BAQDAgOoMB0GA1UdDgQWBBT6eagam2IV9s2XkOesFOxL
NRRmZTANBgNVHSMEIDAeBQLj1Enc9uzLZKw8msCBSyUNRGdKIIIGALOnqfHDMBoG
A1UdEQQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzqg8AAAABBAAtTRVJWRVJfTk9E
RTANBgkqhkiG9w0BAQwFAAOCAQEAAoQVMDr8SeS6+F1O3C7ALWp5V0b9FUeab49n7
M4fv6yPy+ZC07cVkgQ1qde9lIsn5vOHuhdMJOQeX0EC1FG4zs0gvbQ8/HVcoD7dN
fB98nJXm+jOrKOKcNAvgaddtAUX7mDI48mIVrqstjibXxMlgHBMZ382Ujp5xcOnF
lKU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNm3y1Di6hnsijvH6aJf5UVWgYq7Z3
FOSPblo0lyF8Tfavbq8IUpHxYR3eWwV86uJsrV9Pty1xfAy0TqSm9py2+WBIL+ZU
CpMRws+XTtAG5js7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==
-----END CERTIFICATE-----
```

Example Response

```
{
  "id": "82292365-39a0-4000-a756-963e7952b483",
  "uri": "kylo:kylo:connectionmgmt:DSM-node:dsm-connection-82292365-39a0-4000-a756-963e7952b483",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-02-03T10:50:44.563323356Z",
```

```

    "hostname": "node2",
    "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIEETCCAvmgAwIBAgIGC
BPpVT3oMA0GCSqGSIb3DQEBAUAMHMxIzAhBgNVBAMT\nGkNHIENBIFMgb24gc3lzNzgxMDAucWEuY29t
MQswCQYDVQQLLEwJRQTESMBAGA1UE\nChMJVWm9ybWV0cm1jMREwDwYDVQQHEwhTYW4gSm9zZTELMakGA1U
ECBMCCQ0ExCzAJ\nBgNVBAYTAlVTMB4XDTEwMDEzMDA3MzYwOV0XDTMxMDEzMTEwMDEzMDA3MzYwOV0w
aDEYMBYg\nA1UEAxMPC3lzNzgxMDAucWEuY29tMQswCQYDVQQLLEwJRQTESMBAGA1UEChMJVWm9y\n\nnbWV0cm1jMREwDwY
DVQQHEwhTYW4gSm9zZTELMakGA1UECBMCCQ0ExCzAJBgNVBAYT\nAlVTMIIIBIjANBgkqhkiG9w0BAQEFAA
OCAQ8AMIIBCgKCAQEAWVYaXdacb0/g0u41\n\nnuPtRowUK1lroUUNBZ8ileAwPfItkZPo3pSVzxLKnqB+7v
KSEy6t6JfBH/zTzVQxd\n\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJzm8QuxKw2ANQNKbV
WcEZ\n\nnhpy9C9WQUM2Td9MD0vhzDGCdgleZnJtqDQdgTM95Xs8RqK/lbGn3x8YLUKiUcTVA\n\nn4NhTyNVGp
nvu0i/SPeLZNScRUXI0awqYuavMS3ty5Ooxd78AF3qHiutddvzeR5Ry\n\nn20w/Xralbk6AuVHtBkaB6CKv
SG1dUi/pFHuXw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\n\nnHBZCQwIDAQABo4G1MIGYMBYGA1UdJQEB/wQMMAo
GCCsGAQUFBwMBMAwGA1UdEwEB\n\n/n/wCMAAwDgYDVR0PAQH/BAQDAgOoMB0GA1UdDgQWBBT6eagam2IV9s
2XkOesFOxL\n\nnNRRmZTAnBgNVHSMEIDAegBQLjleEnc9uzLZKw8msCBSyUnRGdKIIGALOnqfHDMBoG\n\nA1U
dEQQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzg8AAAABBAtrVJWRVJfTk9E\n\nnRTANBgkqhkiG9w0BAQ
wFAAOCAQEAAoQVMDr8SeS6+F103C7ALWp5V0b9FUeab49n7\n\nnM4fV6yPy+ZC07cVkgG1qde9lIsn5vOHuh
dMJOQeX0EC1FG4zs0gvbQ8/HVcoD7dN\n\nnfB98nJXm+jOrKOKcNAvGaddtAUX7mDI48mIVrqstjibXxMlg
HBMZ382Ujp5xcOnF\n\nnlKU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNm3y1Di6hnsijvH6aJf5UVWgYq7Z
3\n\nnFOSPBlo0lyF8Tfavbq8IUphXyR3eWwV86uJsrv9Pty1xfAy0TqSm9py2+WBIL+ZU\n\nnCpMRws+XTtAG
5js7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\n\n-----END CERTIFICATE-----\n\n",
    "last_connection_ok": null,
    "connection_id": "d873bac7-adeb-43a9-b3cf-2e76cbd40bd2"
}

```

Getting Node Details in a DSM Connection

To get node details in a DSM connection, run:

Syntax

```
ksctl connectionmgmt dsm node get --id <Connection-Name/ID> --node-id <Server-Nod
e-ID>
```

Example Request

```
ksctl connectionmgmt dsm node get --id dsm-connection --node-id test.node1
```

Example Response

```

{
  "id": "7a523daa-aa94-4b11-b31b-c945853663e4",
  "uri": "kylo:kylo:connectionmgmt:dsm-node:dsm-connection-7a523daa-aa94-4b
11-b31b-c945853663e4",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-02-03T10:36:52.946654Z",
  "hostname": "test.node1",
  "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIEETCCAvmgAwIBAgIGC
BPpVT3oMA0GCSqGSIb3DQEBAUAMHMxIzAhBgNVBAMT\nGkNHIENBIFMgb24gc3lzNzgxMDAucWEuY29t
MQswCQYDVQQLLEwJRQTESMBAGA1UE\nChMJVWm9ybWV0cm1jMREwDwYDVQQHEwhTYW4gSm9zZTELMakGA1U
ECBMCCQ0ExCzAJ\nBgNVBAYTAlVTMB4XDTEwMDEzMDA3MzYwOV0XDTMxMDEzMTEwMDEzMDA3MzYwOV0w
aDEYMBYg\nA1UEAxMPC3lzNzgxMDAucWEuY29tMQswCQYDVQQLLEwJRQTESMBAGA1UEChMJVWm9y\n\nnbWV0cm1jMREwDwY
DVQQHEwhTYW4gSm9zZTELMakGA1UECBMCCQ0ExCzAJBgNVBAYT\nAlVTMIIIBIjANBgkqhkiG9w0BAQEFAA
OCAQ8AMIIBCgKCAQEAWVYaXdacb0/g0u41\n\nnuPtRowUK1lroUUNBZ8ileAwPfItkZPo3pSVzxLKnqB+7v
KSEy6t6JfBH/zTzVQxd\n\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJzm8QuxKw2ANQNKbV
WcEZ\n\nnhpy9C9WQUM2Td9MD0vhzDGCdgleZnJtqDQdgTM95Xs8RqK/lbGn3x8YLUKiUcTVA\n\nn4NhTyNVGp
nvu0i/SPeLZNScRUXI0awqYuavMS3ty5Ooxd78AF3qHiutddvzeR5Ry\n\nn20w/Xralbk6AuVHtBkaB6CKv
SG1dUi/pFHuXw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\n\nnHBZCQwIDAQABo4G1MIGYMBYGA1UdJQEB/wQMMAo
GCCsGAQUFBwMBMAwGA1UdEwEB\n\n/n/wCMAAwDgYDVR0PAQH/BAQDAgOoMB0GA1UdDgQWBBT6eagam2IV9s
2XkOesFOxL\n\nnNRRmZTAnBgNVHSMEIDAegBQLjleEnc9uzLZKw8msCBSyUnRGdKIIGALOnqfHDMBoG\n\nA1U
dEQQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzg8AAAABBAtrVJWRVJfTk9E\n\nnRTANBgkqhkiG9w0BAQ
wFAAOCAQEAAoQVMDr8SeS6+F103C7ALWp5V0b9FUeab49n7\n\nnM4fV6yPy+ZC07cVkgG1qde9lIsn5vOHuh
dMJOQeX0EC1FG4zs0gvbQ8/HVcoD7dN\n\nnfB98nJXm+jOrKOKcNAvGaddtAUX7mDI48mIVrqstjibXxMlg
HBMZ382Ujp5xcOnF\n\nnlKU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNm3y1Di6hnsijvH6aJf5UVWgYq7Z
3\n\nnFOSPBlo0lyF8Tfavbq8IUphXyR3eWwV86uJsrv9Pty1xfAy0TqSm9py2+WBIL+ZU\n\nnCpMRws+XTtAG
5js7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\n\n-----END CERTIFICATE-----\n\n"
}

```

```

A1UEAxMPc3lzNzgxMDAucWEuY29tMQswCQYDVQQLLEwJRQTESMBAGA1UEChMJVm9y\nnbWV0cm1jMREwDwY
DVQQHEWhTYW4gSm9zZTELMakGA1UECBMCQ0ExCzAJBgNVBAYT\nAlVTMIIIBIjANBgkqhkiG9w0BAQEFAA
OCAQ8AMIIBCgKCAQEAwVYaXdacb0/g0u41\nuPtRowUK1lroUUNBZ8ileAwPfItkZPo3pSVzxLKnqB+7v
KSEy6t6JfBH/zTzVQxd\n6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJzm8QuxKw2ANQNkbV
WcEZ\nhpy9C9WQUM2Td9MD0vhzDGCDg1eZNjtqDQdgTM95Xs8RqK/lbGn3x8YLUKiUcTVA\nn4NhTyNVGp
nvu0i/SPeLZNScRUXI0awqYuavMS3ty5Ooxd78AF3qHiutddvzeR5Ry\nn20w/Xralbk6AuVHtBkaB6CKv
SG1dUi/pFHUxw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\nnHBZCQwIDAQABo4G1MIGyMBYGA1UdJQEB/wQMAAo
GCCsGAQUFBwMBMAwGA1UdEwEB\nn/wQCAAAdDgYDVR0PAQH/BAQDAgOoMB0GA1UdDgQWBBT6eagam2IV9s
2XkOesFOxL\nnNRRmZTANBgNVHSMEIDAegBQLj1Enc9uzLzKw8msCBSyUnRGdKIIGALOnqfHDMBoG\nnAlU
dEQQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzg8AAAABBAAtTRVJWRVJfTk9E\nnRTANBgkqhkiG9w0BAQ
wFAAOCAQEAoQVMDr8SeS6+F1O3C7ALWp5V0b9FUeab49n7\nnM4fV6yPy+ZC07cVkgQ1qde9lIsn5vOHuh
dMJOQeX0EC1FG4zs0gvbQ8/HVcoD7dN\nnfB98nJXm+jOrKOKcNAvgAddtAUX7mDI48mIVrqstjibXxMlg
HBMZ382Ujp5xcOnF\nnlKU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNm3y1Di6hnsijvH6aJf5UVWgYq7Z
3\nfOFPB1o0lyF8Tfavbq8IUphXyR3eWwV86uJsrV9Pty1xfAy0TqSm9py2+WBIL+ZU\nnCpMRws+XTtAG
5js7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\n-----END CERTIFICATE-----\n",
    "last_connection_ok": null,
    "connection_id": "d873bac7-adeb-43a9-b3cf-2e76cbd40bd2"
}

```

Updating a Node in a DSM Connection

To update a node in a DSM Connection, run:

Syntax

```

ksctl connectionmgmt dsm node modify --id <Connection-Name/ID> --node-id <Server-
Node-ID> --hostname <Server-Hostname> --server-cert-file <Server-Certificate-File>

```

Example Request

```

ksctl connectionmgmt dsm node modify --id dsm-connection --node-id test.node2 --h
ostname node3

```

Example Response

```

{
  "id": "bf333cb2-ad27-45ef-b1ba-5a85408e4141",
  "uri": "kylo:kylo:connectionmgmt:dsm-node:dsm-connection-bf333cb2-ad27-45ef-b
1ba-5a85408e4141",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2021-02-03T10:36:52.948554Z",
  "hostname": "node3",
  "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIIEETCCAvmgAwIBAgIGCBPpV
T3oMA0GCSqGSIb3DQEBAUAMHMxIzAhBgNVBAMT\nnGkNHIENBIFMgb24gc3lzNzgxMDAucWEuY29tMQsw
CQYDVQQLLEwJRQTESMBAGA1UE\nnChMJVm9ybWV0cm1jMREwDwYDVQQHEWhTYW4gSm9zZTELMakGA1UECBM
CQ0ExCzAJ\nnBgNVBAYTAlVTMB4XDTIxMDEzMzYwOV0XDTMxMDEzMzYwOV0wADEYMBYG\nnAlUE
AxMPc3lzNzgxMDAucWEuY29tMQswCQYDVQQLLEwJRQTESMBAGA1UEChMJVm9y\nnbWV0cm1jMREwDwYDVQQ
HEWhTYW4gSm9zZTELMakGA1UECBMCQ0ExCzAJBgNVBAYT\nnAlVTMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ
8AMIIBCgKCAQEAwVYaXdacb0/g0u41\nnuPtRowUK1lroUUNBZ8ileAwPfItkZPo3pSVzxLKnqB+7vKSEy
6t6JfBH/zTzVQxd\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJzm8QuxKw2ANQNkbVWcE
Z\nhpy9C9WQUM2Td9MD0vhzDGCDg1eZNjtqDQdgTM95Xs8RqK/lbGn3x8YLUKiUcTVA\nn4NhTyNVGpnvu

```

```
0i/SPeLZNScRUXI0awqYuavMS3ty5Ooxd78AF3qHiutddvzeR5Ry\n20w/Xralbk6AuVHtBkaB6CKvSG1
dUi/pFHuXw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\nHBZCQwIDAQABo4G1MIGyMBYGA1UdJQEB/wQMMAoGCC
sGAQUFBwMBMAwGA1UdEwEB\n/wQCMAAwDgYDVR0PAQH/BAQDAgOoOMB0GA1UdDgQWBbT6eagam2IV9s2Xk
OesFOxL\nNRRmZTAnBgNVHSMEIDAegBQLj1Enc9uzLZKw8msCBSyUnRGdKIIGALOnqfHDMBoG\nA1UdEQ
QTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzg8AAAABBAAtTRVJWRVJfTk9E\nnRTANBgkqhkiG9w0BAQwFA
AOCAQEAAoQVMDr8SeS6+F1O3C7ALWp5V0b9FUeab49n7\nnM4fV6yPy+ZC07cVkgG1qde9lIsn5vOHuhdMJ
OQeX0EC1FG4zs0gvbQ8/HVcoD7dN\nnfB98nJXm+jOrKOKcNAvGaddtAUX7mDI48mIVrqstjibXxMlgHBM
Z382Ujp5xcOnF\nnlKU9NSqicW3wYAFaJH9NiVSINQEA6AizVuNmX3y1Di6hnsijvH6aJf5UVWgYq7Z3\nn
FOSPBlo0lyF8Tfavbq8IUpHxYR3eWwV86uJsrV9Pty1xfAy0TqSm9py2+WBIL+ZU\nnCpMRws+XTtAG5js
7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\n-----END CERTIFICATE-----\n",
    "last_connection_ok": null,
    "connection_id": "d873bac7-edef-43a9-b3cf-2e76cbd40bd2"
  }
}
```

Deleting a Node in a DSM Connection

To delete a node in a DSM connection, run:

Syntax

```
ksctl connectionmgmt dsm node delete --id <Connection-Name/ID> --node-id <Node-I
D>
```

Example Request

```
ksctl connectionmgmt dsm node delete --id dsm-connection --node-id node3
```

There will be no response if node is deleted successfully.

Getting List of all Nodes in a DSM Connection

To list all the nodes in a DSM connection, run:

Syntax

```
ksctl connectionmgmt DSM node list --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt dsm node list --id dsm-connection
```

Example Response

```
{
  "skip": 0,
```



```

"limit": 10,
"total": 2,
"resources": [
    {
        "id": "bf333cb2-ad27-45ef-b1ba-5a85408e4141",
        "uri": "kylo:kylo:connectionmgmt:dsm-node:dsm-connection-
bf333cb2-ad27-45ef-b1ba-5a85408e4141",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2021-02-03T10:36:52.948554Z",
        "hostname": "test.node2",
        "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIEE
TCCAvmgAwIBAgIGCBPpVT3oMA0GCSqGSIb3DQEBAUAMHMxIzAhBgNVBAMT\nnGkNHIENBIFMgb24gc3lz
NzgxMDAucWEuY29tMQswCQYDVQLEwJRQTESMBAGA1UE\nnChMJVm9ybWV0cmljMREwDwYDVQQHEwhTYW4
gSm9zZTELMakGA1UECBMCQ0ExCzAJ\nnBgNVBAYTA1VTMB4XDTIxMDEzMzYwOVVoXDTMxMDEzMzYw
YwOVowaDEYMBYG\nnA1UEAxMPC3lzNzgxMDAucWEuY29tMQswCQYDVQLEwJRQTESMBAGA1UEChMJVm9y
\nbWV0cmljMREwDwYDVQQHEwhTYW4gSm9zZTELMakGA1UECBMCQ0ExCzAJBgNVBAYT\nnA1VTMIIIBi
jANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAWVYaXdacB0/g0u41\nnuPtRowUK1lroUUNBZ8ileAwPfItkZ
Po3pSVzXLKnqB+7vKSEy6t6JfBH/zTzVQxd\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJ
zm8QuXKw2ANQNKBvVwCEZ\nnhpy9C9WQUM2Td9MD0vhzDGCDg1eZNjtqDQdgTM95Xs8RqK/lbGn3x8YLUKi
UcTVA\n4NhTyNVGpnuv0i/SPeLZNScRUXI0awqYuavMS3ty5Ooxd78AF3qHiutddvzeR5Ry\nn20w/Xral
bk6AuVHTbKaB6CKvSG1dUi/pFHUxw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\nnHBZCQwIDAQABo4G1MIGyMBY
GA1UdJQEB/wQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB\n/nwQMAAwDgYDVR0PAQH/BAQDAgOoMB0GA1UdDg
QWBBT6eagam2IV9s2XkOesFOxL\nnNRRmZTAnBgNVHSMEIDAegBQLj1Enc9uzLZKw8msCBSyUnRGdKIIGA
LONqfHDMBoG\nnA1UdEQQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzg8AAAABBAAtTRVJWRVJfTk9E\nnRT
ANBgkqhkiG9w0BAQwFAAOCAQEAOQVMDr8SeS6+F1O3C7ALWp5V0b9FUeab49n7\nnM4fv6yPy+ZC07cVvkq
Glqde9lIsn5vOHuhdMJOQeX0EClFG4zs0gvbQ8/HVcoD7dN\nnfB98nJXm+jOrKOKcNavGaddtAUX7mDI4
8mIVrqstjibXxMlgHBMZ382Ujp5xcOnF\nnlKU9NSqicW3wYAFaJH9NiVSINQEa6AizVuNmX3y1Di6hnsi
jvH6aJf5UVWgYq7Z3\nnFOSPBlo0lyF8TfFavbq8IUphXyR3eWwV86uJsrV9Pty1xfAy0TqSm9py2+WBI
L+ZU\nnCpMRws+XTtAG5js7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\n-----END CERTIFICAT
E-----\n",
        "last_connection_ok": null,
        "connection_id": "d873bac7-adeef-43a9-b3cf-2e76cbd40bd2"
    },
    {
        "id": "7a523daa-aa94-4b11-b31b-c945853663e4",
        "uri": "kylo:kylo:connectionmgmt:dsm-node:dsm-connectio
n-7a523daa-aa94-4b11-b31b-c945853663e4",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2021-02-03T10:36:52.946654Z",
        "hostname": "test.node1",
        "server_certificate": "-----BEGIN CERTIFICATE-----\nMIIEE
TCCAvmgAwIBAgIGCBPpVT3oMA0GCSqGSIb3DQEBAUAMHMxIzAhBgNVBAMT\nnGkNHIENBIFMgb24gc3lz
NzgxMDAucWEuY29tMQswCQYDVQLEwJRQTESMBAGA1UE\nnChMJVm9ybWV0cmljMREwDwYDVQQHEwhTYW4
gSm9zZTELMakGA1UECBMCQ0ExCzAJ\nnBgNVBAYTA1VTMB4XDTIxMDEzMzYwOVVoXDTMxMDEzMzYw
YwOVowaDEYMBYG\nnA1UEAxMPC3lzNzgxMDAucWEuY29tMQswCQYDVQLEwJRQTESMBAGA1UEChMJVm9y
\nbWV0cmljMREwDwYDVQQHEwhTYW4gSm9zZTELMakGA1UECBMCQ0ExCzAJBgNVBAYT\nnA1VTMIIIBi
jANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAWVYaXdacB0/g0u41\nnuPtRowUK1lroUUNBZ8ileAwPfItkZ
Po3pSVzXLKnqB+7vKSEy6t6JfBH/zTzVQxd\nn6cPtG5rnUeDSeGiYDmbVSULY9XmaETb2nA2XRuv9yGxJ
zm8QuXKw2ANQNKBvVwCEZ\nnhpy9C9WQUM2Td9MD0vhzDGCDg1eZNjtqDQdgTM95Xs8RqK/lbGn3x8YLUKi
UcTVA\n4NhTyNVGpnuv0i/SPeLZNScRUXI0awqYuavMS3ty5Ooxd78AF3qHiutddvzeR5Ry\nn20w/Xral
bk6AuVHTbKaB6CKvSG1dUi/pFHUxw+49Fn2+gVPP0RYXGEwVm5d6wJ3L\nnHBZCQwIDAQABo4G1MIGyMBY
GA1UdJQEB/wQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB\n/nwQMAAwDgYDVR0PAQH/BAQDAgOoMB0GA1UdDg
QWBBT6eagam2IV9s2XkOesFOxL\nnNRRmZTAnBgNVHSMEIDAegBQLj1Enc9uzLZKw8msCBSyUnRGdKIIGA
LONqfHDMBoG\nnA1UdEQQTMBGCD3N5czc4MTAwLnFhLmNvbTAWBgcrzg8AAAABBAAtTRVJWRVJfTk9E\nnRT
ANBgkqhkiG9w0BAQwFAAOCAQEAOQVMDr8SeS6+F1O3C7ALWp5V0b9FUeab49n7\nnM4fv6yPy+ZC07cVvkq
Glqde9lIsn5vOHuhdMJOQeX0EClFG4zs0gvbQ8/HVcoD7dN\nnfB98nJXm+jOrKOKcNavGaddtAUX7mDI4
8mIVrqstjibXxMlgHBMZ382Ujp5xcOnF\nnlKU9NSqicW3wYAFaJH9NiVSINQEa6AizVuNmX3y1Di6hnsi
jvH6aJf5UVWgYq7Z3\nnFOSPBlo0lyF8TfFavbq8IUphXyR3eWwV86uJsrV9Pty1xfAy0TqSm9py2+WBI
L+ZU\nnCpMRws+XTtAG5js7OkJbgKvoVf+GtWpSw4/OSF4PgGnfzoZ0YA==\n-----END CERTIFICAT

```

```
E-----\n",
      "last_connection_ok": null,
      "connection_id": "d873bac7-edef-43a9-b3cf-2e76cbd40bd2"
    ]
  }
}
```

OIDC

OIDC connections to the CipherTrust Manager for CTE agents can be configured using the following:

- [Managing OIDC Connections using GUI](#)
- [Managing OIDC Connections using ksctl](#)

Caution

You can only create an OIDC connection through Connection Manager, as described on this page, to manage CTE agents.

You cannot create an OIDC connection to manage CipherTrust Manager users through Connection Manager. Manage these OIDC connections through [Access Management menus](#) and `ksctl connections` commands, and not Connection Manager.

If you create an OIDC connection for CipherTrust Manager users through Access Management, a duplicate connection manager OIDC connection is visible in Connection Manager, with no product. This duplicate is created solely for secure storage of credentials; we strongly recommend you do not modify the duplicate.

Managing OIDC Connections using GUI

To manage OIDC connections using GUI, perform the following steps:

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.
4. On the Add Connection screen, select category as **OIDC/LDAP**.
5. Select OIDC/LDAP Type as **OIDC** and click **Next**.
6. Specify connection **Name** and **Description** and click **Next**.
7. Configure the below parameters.
 - **OIDC Provider**: URL of the host server where an OIDC connection exists.

- **Client ID:** this is an Application ID of the OIDC application. It is used with Client Secret to authenticate the application.

Note

The *Client ID* is a unique field and cannot be updated after the connection is established.

- **Client Secret:** this authentication method uses the application password of the Client ID to enable communication between OIDC and CipherTrust Manager.

8. Click **Next** to move to the **Add Products** screen of the Add Connection wizard.

Managing OIDC Connections using ksctl

You can use ksctl to manage OIDC connections to authenticate CTE agents.

Note

OIDC connections to authenticate CipherTrust Manager users are configured [outside of connection manager](#).

The following operations can be performed:

- Create/Get/Update/Delete an OIDC connection
- List all OIDC connections

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Unique connection name.
description	Optional	Connection description.
clientid	Mandatory	Client Id for the connection.
conn-url	Mandatory	URL for the connection.
products	Optional	CipherTrust products associated with the connection.
secret	Mandatory	Client Secret of the OIDC connection.

Creating an OIDC Connection

To create an OIDC connection, run:

Syntax

```
ksctl connectionmgmt oidc create --name <connection-name> --description <description> --products <products-name> --meta <key:value> --clientid <client-id> --secret <client-secret> --conn-url <URL>
```

Note

The `clientid` is a unique field and cannot be updated after the connection is established.

Example Request

```
ksctl connectionmgmt oidc create --name oidc-connection --clientid 2cc2d7db155c472fb2484ca4072d1bb3 --secret client_secret --conn-url www.oidc-test-connection.com
```

Example Response

```
{
  "id": "901c9158-9ea1-4715-a92c-882c4a9828ee",
  "uri": "kylo:kylo:connectionmgmt:connections:oidc-connection-901c9158-9ea1-4715-a92c-882c4a9828ee",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-16T05:21:20.725493705Z",
  "updatedAt": "2022-05-16T05:21:20.723628387Z",
  "service": "OIDC",
  "category": "IdP",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "oidc-connection",
  "client_id": "2cc2d7db155c472fb2484ca4072d1bb3",
  "url": "www.oidc-test-connection.com",
  "checksum": "87de71462a719f4de23cbec7be6e22f5390705a009e0dd0516ddd8be17cad187"
}
```

Note

The checksum is a SHA256 checksum value generated from the `client_secret` and `url` parameters. It will be updated if any of these two parameter values get changed.

Getting Details of an OIDC Connection

To get details of an OIDC connection, run:

Syntax

```
ksctl connectionmgmt oidc get <connection-name/ID>
```

Example Request

```
ksctl connectionmgmt oidc get --id 901c9158-9ea1-4715-a92c-882c4a9828ee
```

Example Response

```
{
  "id": "901c9158-9ea1-4715-a92c-882c4a9828ee",
  "uri": "kylo:kylo:connectionmgmt:connections:oidc-connection-901c9158-9ea1-4715-a92c-882c4a9828ee",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-05-16T05:21:20.725494Z",
  "updatedAt": "2022-05-16T05:21:20.723628Z",
  "service": "OIDC",
  "category": "IdP",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "oidc-connection",
  "client_id": "2cc2d7db155c472fb2484ca4072d1bb3",
  "url": "www.oidc-test-connection.com",
  "checksum": "87de71462a719f4de23cbec7be6e22f5390705a009e0dd0516ddd8be17cad187"
}
```

Updating an OIDC Connection

To update an OIDC connection, run:

Syntax

```
ksctl connectionmgmt oidc modify --id <connection-name/ID> --description <description> --products <products-name> --meta <key:value> --secret <client-secret> --conn-url <URL>
```

Example Request

```
ksctl connectionmgmt oidc modify --id 901c9158-9ea1-4715-a92c-882c4a9828ee --conn-url https://new-oidc-url/ --secret new_secret
```

Example Response

```
{
```

```

    "id": "901c9158-9ea1-4715-a92c-882c4a9828ee",
    "uri": "kylo:kylo:connectionmgmt:connections:oidc-connection-901c9158-9ea1-4715-a92c-882c4a9828ee",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2022-05-16T05:21:20.725494Z",
    "updatedAt": "2022-05-16T05:29:02.230980966Z",
    "service": "OIDC",
    "category": "IdP",
    "last_connection_ok": null,
    "last_connection_at": "0001-01-01T00:00:00Z",
    "name": "oidc-connection",
    "client_id": "2cc2d7db155c472fb2484ca4072d1bb3",
    "url": "https://new-oidc-url/",
    "checksum": "6277c08784599bda9d4ccb0c6b0c9ed87b6f5add00c3babcb2858a290cdc70523"
  }

```

Deleting an OIDC Connection

To delete an OIDC connection, run:

Syntax

```
ksctl connectionmgmt oidc delete --id <connection-name/Id> --force
```

Example Request 1

If connection is not being used currently, run;

```
ksctl connectionmgmt oidc delete --id 901c9158-9ea1-4715-a92c-882c4a9828ee
```

There will be no response if the OIDC connection is deleted successfully.

Example Request 2

If connection is in-use, run:

```
ksctl connectionmgmt oidc delete --id 901c9158-9ea1-4715-a92c-882c4a9828ee --force
```

There will be no response if the OIDC connection is deleted successfully.

Getting List of OIDC Connections

To list all the OIDC connections, run:

Syntax

```
ksctl connectionmgmt oidc list
```

Example Request

```
ksctl connectionmgmt oidc list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "901c9158-9ea1-4715-a92c-882c4a9828ee",
      "uri": "kylo:kylo:connectionmgmt:connections:oidc-connection-901c9158-9ea1-4715-a92c-882c4a9828ee",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2022-05-16T05:21:20.725494Z",
      "updatedAt": "2022-05-16T05:21:20.723628Z",
      "service": "OIDC",
      "category": "IdP",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "oidc-connection",
      "client_id": "2cc2d7db155c472fb2484ca4072d1bb3",
      "url": "www.oidc-test-connection.com"
    }
  ]
}
```

LDAP

LDAP connection to the CipherTrust Manager can be configured using the following:

- Managing LDAP Connections using GUI
- Managing LDAP Connections using ksctl

Note

The LDAP connection available through connection manager is separate and additional to the LDAP connection available through Access Management.

Managing LDAP Connections using GUI

To manage LDAP connections using GUI, perform the following steps:

1. Log on to CipherTrust Manager UI as an administrator.
2. Navigate to **Access Management > Connections**.
3. Click **Add Connection**.

-
4. On the Add Connection screen, select category as **OIDC/LDAP**.
 5. Select OIDC/LDAP Type as **LDAP** and click **Next**.
 6. Specify connection **Name** and **Description** and click **Next**.
 7. Configure the below parameters.

- **Server URL** - enter the URL of the LDAP server.

If the **Server URL** scheme begins with 'ldaps://', following additional configuration fields become available:

- **Server's Root CA Certificate** - upload the server's root CA certificate or provide details in the text format. Certificate details are verified to check if the server is trusted. If server's root CA certificate is not provided, then server verification is performed using the operating system's default trust store.

Click **+ Server's Root CA Certificate** to upload multiple root CA certificates of the server.

- **Disable verifying the server certificate** - select this check box to disable the server's certificate verification. If selected, you don't need to upload any server's root CA certificate and the **Server's Root CA Certificate** field disappears.
- **Bind DN** - enter the bind dn object that has permission to search on the root DN for users.
- **Server Bind Password** - enter the bind password of the server.
- **Base DN** - enter the starting point an LDAP server uses when searching for users.
- **Search Filter for listing users** - enter the LDAP search filter to restrict the set of users who will be allowed to log on to the LDAP server.
- **User's login name attribute** - enter the attribute on the user object that contains the username required to log on with.
- **Group Base DN** - enter the starting point an LDAP server uses when searching for groups.
- **Search filter for listing groups** - enter the LDAP search filter to restrict the set of groups who will be allowed to log on to the LDAP server.
- **Group Name Attribute** - enter the attributes on the group object that contain the friendly name of the group.
- **User's Distinguished Name** - enter the user's distinguished name
- **Group id attribute for group mapping** - enter the attribute on the group object that contains the group identifier (name). This value should be unique.
- **User's group membership attribute** - enter the attribute that contains the group membership information, that is referring to users which are member of the group.

- **Username for test (required for Test)** - enter the username required for test connection.
 - **Password for test (required for Test)** - enter the password required for test connection.
 - **LDAP server max page Size** - sets the page size which determines the number of users/groups to be fetched from the LDAP server using pagination.
 - **LDAP early listing stop** - select/clear the check box to set the value to true/false. When set to false displays the total number of users/groups that are present on the ldap server. When set to `true`, the value of total in the response is displayed as `-1` indicating that more records are present on the ldap server than the limit specified, otherwise the correct `total` records number will be visible.
8. Click the **Test Credentials** button to check whether the connection is configured correctly. If the test is successful, the status is `OK` else the status is `Fail`.
 9. Click **Next** to move to the [Add Products](#) screen of the Add Connection wizard.

Note

Currently, the only product supported for LDAP connection is CTE.

Managing LDAP Connections using ksctl

The following operations can be performed:

- Create/Get/Update/Delete an LDAP connection
- List all LDAP connections
- Test an existing LDAP connection
- Test the new LDAP connection parameters before establishing the connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Unique connection name.
description	Optional	Connection description.
server-url	Mandatory	LDAP URL for your server. (For example, ldap://172.16.2.2:3268)
bind-dn	Mandatory	Object which has permission to search under the root DN for LDAP users.
bind-password	Mandatory	Password for the Bind DN object of

Parameter	Mandatory/Optional	Description
		the LDAP connection.
base-dn	Optional	Starting point to use when searching for LDAP users.
search-filter	Optional	LDAP search filter to further restrict the set of users who will be allowed to log in.
user-login-attribute	Optional	Attribute inside the user object that contains the username to login with.
group-base-dn	Optional	Starting point to use when searching for groups. This value can be left empty to disable group support for this connection.
group-search-filter	Optional	Search filter for listing groups.
group-name-attribute	Optional	Attribute inside the group object which contains the friendly name of the group.
group-member-field	Optional	Attribute inside the group object that contains group membership information, basically which users are members of the group. For example: member, memberUid.
root-ca	Optional	PEM encoded certificate used to determine if the server is trusted. Only applies if the <code>server_url</code> scheme is <code>ldaps</code> . If not provided, then the server's certificate is verified using the CA of the operating system.
server-maximum-page-size	Optional	Sets the page size, which determines the number of users/groups to be fetched from the LDAP server using pagination. By default, when <code>early_listing_stop</code> is <code>false</code> and <code>server-maximum-page-size</code> is 0, entries are retrieved from the LDAP server without pagination.
early-listing-stop	Optional	When the <code>early-listing-stop</code> parameter is set to <code>false</code> , it shows the total number of users or groups on the LDAP server. If it is set to <code>true</code> , the response indicates the total count as <code>-1</code> , which means there are more records on the LDAP server than the limit specified. The default value for <code>early-</code>

Parameter	Mandatory/Optional	Description
		listing-stop is false. For example, if we set early-listing-stop to true and the LDAP server has a total of 5000 users while the limit parameter on the CipherTrust Manager is set to 1000, the response will return 1000 entries, and the total will be displayed as -1.

Creating an LDAP Connection

To create an LDAP connection, run:

Syntax

```
ksctl connectionmgmt ldap create --name <connection-name> --server-url <url> --bind-dn <bind-dn> --bind-password <bind-password> --base-dn <base-dn> --search-filter <user-search-filter> --user-login-attribute <user-login-attribute> --group-base-dn <group-base-dn> --group-search-filter <group-search-filter> --group-name-attribute <group-name-attribute> --group-member-field <group-member-field> --root-ca '-----BEGIN CERTIFICATE-----...-----END CERTIFICATE-----' --server-maximum-page-size <server-maximum-page-size> --early-listing-stop <early-listing-stop>
```

Example Request 1

```
ksctl connectionmgmt ldap create --name "ldapConn" --server-url "ldap://172.27.0.6:389" --bind-dn "cn=admin,dc=planetexpress,dc=com" --bind-password "GoodNews Everyone" --base-dn "dc=planetexpress,dc=com" --search-filter "(objectclass=User)" --user-login-attribute "uid" --group-base-dn "ou=people, dc=planetexpress, dc=com" --group-search-filter "(objectclass=Group)" --group-name-attribute "test1" --group-member-field "member"
```

Example Response 1

```
{
  "id": "200bb517-854a-43c8-af54-7bc12cd35eb0",
  "uri": "kylo:kylo:connectionmgmt:connections:ldapconn-200bb517-854a-43c8-af54-7bc12cd35eb0",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-07-28T07:48:09.151996264Z",
  "updatedAt": "2022-07-28T07:48:09.150581053Z",
  "service": "LDAP",
  "category": "IdP",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "ldapConn",
  "server_url": "ldap://172.27.0.6:389",
  "insecure_skip_verify": false,
  "bind_dn": "cn=admin,dc=planetexpress,dc=com",
  "base_dn": "dc=planetexpress,dc=com",
  "search_filter": "(objectclass=User)",
```

```
{
  "user_login_attribute": "uid",
  "group_base_dn": "ou=people, dc=planetexpress, dc=com",
  "group_filter": "(objectclass=Group)",
  "group_name_attribute": "test1",
  "group_member_field": "member"
}
```

Example Request 2

```
ksctl connectionmgmt ldap create --name "ldapConn1" --server-url "ldap://172.27.0.6:389" --bind-dn "cn=admin,dc=planetexpress,dc=com" --bind-password "GoodNewsEveryone" --base-dn "dc=planetexpress,dc=com" --search-filter "(objectclass=User)" --user-login-attribute "uid" --group-base-dn "ou=people, dc=planetexpress, dc=com" --group-search-filter "(objectclass=Group)" --group-name-attribute "test1" --group-member-field "member" --server-maximum-page-size 10000 --early-listing-stop true
```

Example Response 2

```
{
  "id": "200bb517-854a-43c8-af54-7bc12cd35eb0",
  "uri": "kylo:kylo:connectionmgmt:connections:ldapconn-200bb517-854a-43c8-af54-7bc12cd35eb0",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-07-28T07:48:09.151996264Z",
  "updatedAt": "2022-07-28T07:48:09.150581053Z",
  "service": "LDAP",
  "category": "IdP",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "ldapConn1",
  "server_url": "ldap://172.27.0.6:389",
  "insecure_skip_verify": false,
  "bind_dn": "cn=admin,dc=planetexpress,dc=com",
  "base_dn": "dc=planetexpress,dc=com",
  "server_maximum_page_size": 10000,
  "early_listing_stop": true,
  "search_filter": "(objectclass=User)",
  "user_login_attribute": "uid",
  "group_base_dn": "ou=people, dc=planetexpress, dc=com",
  "group_filter": "(objectclass=Group)",
  "group_name_attribute": "test1",
  "group_member_field": "member",
}
```

Getting Details of an LDAP Connection

To get details of an LDAP connection, run:

Syntax

```
ksctl connectionmgmt ldap get --id <connection-name/Id>
```

Example Request

```
ksctl connectionmgmt ldap get --id "ldapConn1"
```

Example Response

```
{
  "id": "045f76ee-3407-445e-84cf-edebdddc688",
  "uri": "kylo:kylo:connectionmgmt:connections:ldapconn-045f76ee-3407-445e-84cf-edebdddc688",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-07-28T13:25:04.652637Z",
  "updatedAt": "2022-07-28T13:25:17.190261Z",
  "service": "LDAP",
  "category": "IdP",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "ldapConn",
  "server_url": "ldap://54.27.0.7:389",
  "insecure_skip_verify": false,
  "bind_dn": "cn=admin,dc=planetexpress,dc=com",
  "base_dn": "dc=planetexpress,dc=com",
  "server_maximum_page_size": 10000,
  "early_listing_stop": true,
  "search_filter": "(objectclass=User)",
  "user_login_attribute": "uid",
  "group_base_dn": "ou=people, dc=planetexpress, dc=com",
  "group_filter": "(objectclass=Group)",
  "group_name_attribute": "test1",
  "group_member_field": "member"
}
```

Updating an LDAP Connection

To update an LDAP connection, run:

Syntax

```
ksctl connectionmgmt ldap modify --id <connection-name/Id> --server-url <url> --bind-dn <bind-dn> --bind-password <bind-password> --base-dn <base-dn> --search-filter <user-search-filter> --user-login-attribute <user-login-attribute> --group-base-dn <group-base-dn> --group-search-filter <group-search-filter> --group-name-attribute <group-name-attribute> --group-member-field <group-member-field> --root-ca '-----BEGIN CERTIFICATE-----...-----END CERTIFICATE-----' --server-maximum-page-size <server-maximum-page-size> --early-listing-stop <--early-listing-stop>
```

Example Request 1

```
ksctl connectionmgmt ldap modify --id "ldapConn" --server-url "ldap://54.27.0.7:389"
```

Example Response 1

```
{
```

```

    "id": "045f76ee-3407-445e-84cf-edebdddc688",
    "uri": "kylo:kylo:connectionmgmt:connections:ldapconn-045f76ee-3407-445e-84cf-edebdddc688",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2022-07-28T13:25:04.652637Z",
    "updatedAt": "2022-07-28T13:25:17.190260884Z",
    "service": "LDAP",
    "category": "IdP",
    "last_connection_ok": null,
    "last_connection_at": "0001-01-01T00:00:00Z",
    "name": "ldapConn",
    "server_url": "ldap://54.27.0.7:389",
    "insecure_skip_verify": false,
    "bind_dn": "cn=admin,dc=planetexpress,dc=com",
    "base_dn": "dc=planetexpress,dc=com",
    "search_filter": "(objectclass=User)",
    "user_login_attribute": "uid",
    "group_base_dn": "ou=people, dc=planetexpress, dc=com",
    "group_filter": "(objectclass=Group)",
    "group_name_attribute": "test1",
    "group_member_field": "member"
  }

```

Example Request 2

```

ksctl connectionmgmt ldap modify --id "ldapConn1" --server-url "ldap://54.27.0.7:389" --server-maximum-page-size 10000 --early-listing-stop true

```

Example Response 2

```

{
  "id": "045f76ee-3407-445e-84cf-edebdddc688",
  "uri": "kylo:kylo:connectionmgmt:connections:ldapconn-045f76ee-3407-445e-84cf-edebdddc688",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2022-07-28T13:25:04.652637Z",
  "updatedAt": "2022-07-28T13:25:17.190260884Z",
  "service": "LDAP",
  "category": "IdP",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "ldapConn",
  "server_url": "ldap://54.27.0.7:389",
  "insecure_skip_verify": false,
  "bind_dn": "cn=admin,dc=planetexpress,dc=com",
  "base_dn": "dc=planetexpress,dc=com",
  "search_filter": "(objectclass=User)",
  "user_login_attribute": "uid",
  "server_maximum_page_size": 10000,
  "early_listing_stop": true,
  "group_base_dn": "ou=people, dc=planetexpress, dc=com",
  "group_filter": "(objectclass=Group)",
  "group_name_attribute": "test1",
  "group_member_field": "member"
}

```

Deleting an LDAP Connection

To delete an LDAP connection, run:

Syntax

```
ksctl connectionmgmt ldap delete --id <connection-name/Id>
```

Example Request

```
ksctl connectionmgmt ldap delete --id "ldapConn"
```

There will be no response if the LDAP connection is deleted successfully.

Getting List of LDAP Connections

To list all the LDAP connections, run:

Syntax

```
ksctl connectionmgmt ldap list
```

Example Request

```
ksctl connectionmgmt ldap list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "id": "200bb517-854a-43c8-af54-7bc12cd35eb0",
      "uri": "kylo:kylo:connectionmgmt:connections:ldapconn-200bb517-854a-43c8-af54-7bc12cd35eb0",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2022-07-28T13:25:04.652637Z",
      "updatedAt": "2022-07-28T13:25:17.190261Z",
      "service": "LDAP",
      "category": "IdP",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "ldapConn",
      "server_url": "ldap://54.27.0.7:389",
      "insecure_skip_verify": false,
      "bind_dn": "cn=admin,dc=planetexpress,dc=com",
      "base_dn": "dc=planetexpress,dc=com",
      "search_filter": "(objectclass=User)",
      "user_login_attribute": "uid",
      "group_base_dn": "ou=people, dc=planetexpress, dc=com",
    }
  ]
}
```

```

        "group_filter": "(objectclass=Group)",
        "group_name_attribute": "test1",
        "group_member_field": "member"
    },
    {
        "id": "045f76ee-3407-445e-84cf-edebdddc688",
        "uri": "kylo:kylo:connectionmgmt:connections:ldapconn1-045f76ee-3
407-445e-84cf-edebdddc688",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2022-07-28T13:25:04.652637Z",
        "updatedAt": "2022-07-28T13:25:17.190261Z",
        "service": "LDAP",
        "category": "IdP",
        "last_connection_ok": null,
        "last_connection_at": "0001-01-01T00:00:00Z",
        "name": "ldapConn",
        "server_url": "ldap://54.27.0.7:389",
        "insecure_skip_verify": false,
        "bind_dn": "cn=admin,dc=planetexpress,dc=com",
        "base_dn": "dc=planetexpress,dc=com",
        "search_filter": "(objectclass=User)",
        "user_login_attribute": "uid",
        "server_maximum_page_size": 10000,
        "early_listing_stop": true,
        "group_base_dn": "ou=people, dc=planetexpress, dc=com",
        "group_filter": "(objectclass=Group)",
        "group_name_attribute": "test1",
        "group_member_field": "member"
    }
]
}

```

Testing an Existing LDAP Connection

To test an existing LDAP connection, run:

Syntax

```
ksctl connectionmgmt ldap test --id <connection-name/Id>
```

Example Request

```
ksctl connectionmgmt ldap test --id "045f76ee-3407-445e-84cf-edebdddc688"
```

Example Response

```
{
  "connection_ok": true
}
```

Testing a new LDAP Connection

To test the parameters of a new LDAP connection, run:

Syntax

```
ksctl connectionmgmt ldap test --server-url <url> --bind-dn <bind-dn> --bind-pass
word <bind-password> --base-dn <base-dn> --search-filter <user-search-filter> --u
ser-login-attribute <user-login-attribute> --group-base-dn <group-base-dn> --grou
p-search-filter <group-search-filter> --group-name-attribute <group-name-attribut
e> --group-member-field <group-member-field> -- root-ca '-----BEGIN CERTIFICAT
E-----...-----END CERTIFICATE-----'
```

Example Request

```
ksctl connectionmgmt ldap test --server-url "ldap://54.80.101.85:389" --bind-dn
"cn=admin,dc=planetexpress,dc=com" --bind-password "GoodNewsEveryone" --base-dn
"dc=planetexpress,dc=com" --search-filter "(objectclass=User)" --user-login-attri
bute "uid" --group-base-dn "ou=people, dc=planetexpress, dc=com" --group-search-f
ilter "(objectclass=Group)" --group-name-attribute "test1" --group-member-field
"member" --username "fry" --conn-password "fry"
```

Example Response

```
{
  "connection_ok": true
}
```

Attestation Authority Connection

You can manage Attestation Authority connections in the CipherTrust Manager GUI. CipherTrust Manager uses these connections for communicating with the relevant Attestation Authority to complete the attestation process.

You can add, update, or delete these connections as per your requirement. For more information on provisioning Confidential Computing on CTE clients, see [Confidential Computing](#).

Configure Attestation Authority Connection

Before configuring the connection, [select a connection type](#) for this connection.

To configure an **Attestation Authority** connection in CipherTrust Manager:

1. Enter the URL for the trusted attestation authority.
2. Enter the **Base URL**.
3. Enter the **API Key**.
4. (Optional) Use the **Show API Key** checkbox to show/hide the API key.
5. Enter the **Meta** information.
6. Select the **Admin User** checkbox to create a connection with administrator privileges (admin connection). By default, a connection without administrator privileges (non-admin connection) will be created.

Note

To provision Confidential Computing, you need both an admin and a non-admin connection. The admin connection is necessary to retrieve the policies from the Attestation Authority, while the details of the non-admin connection are shared with the CTE agents.

7. Click **Next**.

After configuring the connection, [select the product](#) associated to this connection.

Update Attestation Authority Connection

After the connection is created, you can update the connection details.

1. In CipherTrust Manager, select **Access Management > Connections**.
2. Click the three dots (...) button for the desired Attestation Authority connection.
3. Select **View/Edit**.
4. Update the details of the **General**, **Connection**, and **Products** sections as required.
5. Click **Update**.

Delete Attestation Authority Connection

To delete an Attestation Authority connection:

1. In CipherTrust Manager, select **Access Management > Connections**.
2. Click the three dots (...) button for the desired Attestation Authority connection.
3. Select **Delete**.
4. In the **Delete Connection** confirmation dialog, click **Delete**.

CM Connection/External CM

The external CM server stores and manages client credentials for CCKM. These connections can be configured and managed by the CipherTrust Manager Administrator or the users who are part of the *admin* group. This functionality allows the CipherTrust Manager to act as an external key source in the connection manager.

Important Notes

- This document, may at times, abbreviate **CipherTrust Manager** as **CM**.

-
- The CM server on which the CCKM will run is referred to as **Client CM**.
 - The CM server that acts as an external key source is referred to as **External CM**.

CM connections to the CipherTrust Manager can be configured and managed:

- Using the GUI
- Using the ksctl

Managing CM Connections using GUI

This section explains the steps to configure the CM/external CM connections using GUI.

Prerequisites

On the client CM.

1. [Create a CipherTrust \(External\) Connection](#)
2. [Generate a Connection CSR](#)

On the domain of the external CM where you want to create a connection.

1. [Create a Client Profile](#)
2. [Generate a Registration Token](#)
3. [Add the Client](#)

After completing the steps, [configure the CM connection](#) on the client CM.

Create a CipherTrust (External) Connection

Add a connection to the external CipherTrust Server.

1. Log on to the CipherTrust Manager.
2. In the left pane, click **Access Management > Connections**.
3. On the Connections screen, under CIPHERTRUST (EXTERNAL) CONNECTIONS, click **Add CipherTrust (external) Server**.
4. On the Add CipherTrust (external) Server screen, enter the **Name** of a CipherTrust (external) Server.
5. Enter the **Node Hostname/IP** of a CipherTrust Manager server node (external CM).

Note

If the external CM is configured to use a custom port (that is, other than 443), then append the custom port to the hostname/IP address. For example, specify <hostname or IP address>:<custom-port>.

To add additional nodes, click **Add CM Node**.

6. Add a **Certificate**. You need to add the root CA of the external CM. The options are:

- **File Upload:** Select and upload the certificate (in PEM format).
- **Text:** Select and paste the certificate content.

Sample Root CA:

```
-----BEGIN CERTIFICATE-----
MIIFrjCCA5agAwIBAgIQe4lLpV6szJdk3He6ctFDszANBgkqhkiG9w0BAQsFADBa MQswCQYDVQQGE
wJVUzELMAkGA1UECBMVFgxDzANBgNVBACTBkF1c3RpbjEPMA0G A1UEChMGVGVhbnBzMRwwGgYDVQ
QDExNDNaXBoZXRJUCnVzdCBSb290IENBMB4XDTEz MDUzMDExMTgwMFoXDTMzMDUyODExMTgwMFowWjE
LMAkGA1UEBhMCVVMxMzA5BjNV BAgTAlRYMQ8wDQYDVQQHEwZBdXN0aW4xDzANBgNVBAoTB1RoYWxl
czEcMBoGA1UE AxMTQ21waGVyVHJ1c3QgUm9vdCBDQTCcAiIwDQYJKoZIhvcNAQEBBQADggIPADCC
AgoCggIBAJD0zVgX8QT2nHGybmOJQUPCwqc0ajQn1lrpzz+eWNuIBYHCRs9WRYO3 Z3+Rc9LcuwoNS
hQDb+XBQHcX9FbN9ZP1h+/UCM78LIXhBCIpJRK1XIcLy38ZHunU oVeaxWRwy2TwdfpRlP9dvOkeZn
lTTq0vsiybTh4Gt5E3sRU7oceWuSJEzT4PF5Wx0 iZXeCoOyqXXqi8lEPS4HqIAGOXY/TkTmr/JohBY
WcZttCA20PemQEfiwLeCRynTe GG4k7/TkoHVkGmGbudg2ot2sR9AaUL10aVz14rSsq2purJO60DzR
A5BtcLLx7537 e6PLCdP5/7B5FVT2KiY9/oj1AXgd4Rp0pzDG3qrfjMhh2rho5Ab7s2zZfZNW/mpm
n2xDnczt2NYnDhVGHViTA3lezsbkpaRaKML1Bnb0JRHPmc+bbpTcs1igrWxjXuyp WDNxQ3WorEy0y
MANAfCrRaeMoN1t58UZKJ0WwRaJyeYpCREBSbRIHz9+uQhRFTSP RZ/V6EacEDQbfueiVigLbeVyhj
RJ65sN+5obEpaVlU3soJPRjRVnhwe6qQdpunf1 u7pcC84Tdmfk6lP2tMHDmEUwTd99ssuGAvdcDAF
NTb+0Cb5JWBOUeSaG3kGTKwaI YF+cfy8sFnHKQ2SYuMxUB7ujf+JgywJ6LnSgzTMbtWeneuxXUGvz
AgMBAAGjcDBu MA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBrnlLCu
e5kDRyTLxtqoWTbEkCSQKjAsBgNVHREEJTAjgSF0ZWNobmljYWwuc3VwcG9ydEB0 aGFsZXNncm91c
C5jb20wDQYJKoZIhvcNAQELBQADggIBACj3OQqhK/Gv+aNhjYi/ cs6QlW5zKBOh4bSy9+6gJ0Nssz
BKJwVT5w3wOhlfwIIJyH/P4IqVVgp9wE9Ymuir efrQGbIHVbBrjzBqPDt9e9wdsZXPbhz88fxNB0X
qgV1Sg8ip5Ccqd68u6mGo40Cu zZFSyDz003ctHJ0THV1jy+u2S7o7eYZ0T9qul1DLTieV743NZI
2/acFqi2G4cIv nSV3xbnfKC+4kzf3wKoful/xs+H8vD+d2lJsAXq4d6ck3ptfLKl035IUd3nS3sJT
k2KCIOjdKU6wfwio/s3XXDYWLdiVPlWrf/+HglIVPP3np2hWUzAeIznm01Y4wUh0 jsDISk8/H6L3/
NBb2DlF5zKpGZCXDoUjw/2dNFaYcGJEL3fVnCHaQVn8yyetJIXS PXzv0ZuvE/6cTCAlhxBkAJqMYt
eJJomOeXv0PB1J+ePM52sYzAVkt3YkFM1QdJ1i LbUgNLQg55s4nswiZK4Dva2S3me+ahrSO9CFw8a
XYh+3r7yKu4kNfQbWiX7RGETH +2UPMg+QbkaqOOnbVMSjGJBK2oe43YKVgdhgLDSBRHRWmlqmjsvY
W0ZKdIgFhDIa tijRkOfDjdh2GyAv7nLbqsyQZN5RS1zjZh9LEGZ8+ETw/hHleQwRSr8B5JtRyNR2
grZ+pEXAN+a6jmqMed5BNnBR
```

-----END CERTIFICATE-----

Note

- To add additional nodes, click **Add CM Certificate**.
- The external CM web server certificate must contain the IP address.

7. Click **Add CipherTrust (external) Server**.

The newly created CipherTrust (external) Server is displayed in the list of CipherTrust (external) Servers.

Generate a Connection CSR

1. Navigate to the CSR Generator page (**CA > CSR Generator**).
2. Generate a Connection CSR and download the CSR certificate.

Create a Client Profile

1. Navigate to the Client Profiles page (**Access Management > Client Profiles**).
2. Add a client profile.

Note

- You can create a client profile using the Local CA and External CA.

Generate a Registration Token

Refer to [Creating a Registration Token](#) for details.

Note

To generate a registration token, you first need to [create a client profile](#).

Add the Client

1. Navigate to the Client Hub page (**Access Management > Client Hub**).

2. Click **Add Client**.
3. Specify a name for the client.
4. Select the Registration Token that you generated.
5. Add the connection **CSR** that you generated [above](#). The options are:
 - **File Upload:** Select and upload the certificate (in PEM format).
 - **Text:** Select and paste the certificate content.
6. Add the client.
7. Save the Client ID and the Client Certificate.

If the external CM is in a clustered environment, the external CM administrator needs to add the client to the Cluster Info Viewers group, so that client CM can read the cluster information.

To add the client to the Cluster Info Viewers group.

1. On the Client Hub page, click the **Name** of the client that you created above.
2. Expand the **GROUPS** section.
3. Disable **Show '<client name>' groups**.
4. Search for `Cluster Info Viewers`.
5. Click **Add**.

Configure a CM connection

1. Select the **CipherTrust (external)** server you created from the drop-down list.
2. Under Authentication:
 - a. Enter the **Client ID** (for example, `18455d84-1b39-48d5-ac86-c649f78703a2`) that you saved while adding the client.
 - b. Add the **Client Certificate** that you saved while [adding the client](#). The options are:
 - **File Upload:** Select and upload the certificate (in PEM format).
 - **Text:** Select and paste the certificate content.

Sample Client Certificate:

```
-----BEGIN CERTIFICATE-----
MIIE4jCCAsqgAwIBAgIQZK0f1r4RadGoL6BHoNm8TANBgkqhkiG9w0BAQsFADBa MQswCQYD
```

```
VQQGEwJVUzELMAkGA1UECBMCFVgxZDZANBgNVBAcTBkF1c3RpbjEPMA0G A1UEChMGVGHbGVz
MRwwGgYDVQQDExNDaXB0ZXJUCnVzdCBSb290IENBMB4XDTIz MDUzMTA2MTc0MV0xXDTI1MDUz
MTA2MTc0MV0wRjEOMAwGA1UEAxMFYWRtaW4xNDAY BgoJkiaJk/IsZAEBEyQxODQ1NWQ4NC0x
YjM5LTQ4ZDUtYW4Ni1jNjQ5Zjc4NzAz YTIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQC9bYT/1nW7EGbf/btk 07j1N76jAFXT8XvUwTrqcmrwAsiUZcTd7bke9vp6w7bokN1I
LQJVfb4EeZYD9lWK Eosnh1h+PwePS2SlrIac9NombfrKNecceB4+cIaU3/UKNo0BAI06/7Q
XPn/sPwL LIF/eG19UzttJdKkg0gg57vwxK3MkvT+TJ55EjVleHGO85JcGWMb8x01AuN1N/K8
fKlSKq9WwopPdiuzbwqya+ulP8KDFz/pmgMo54tIjwUvaxTtUVgoqZFFBiOQYTMx ZK2oDtYq
ioHraXw71Q2fbQ0SjzTubQLU1li6VQlwuaJVouXtjQw1SbiyCL19axIw ydalAgMBAAGjgbcw
gbQwDgYDVR0PAQH/BAQDAgOIMBMGA1UdJQZMMMAoGCCsGAQUF BwMCMawGA1UdEwEB/wQCMAAw
HwYDVR0jBBgwFoAUZ5SwrnuZA0cky8baqFk2xJHE kCowXgYDVR0fBFcwVTBToFGgT4ZNaHR0
cDovL2NpcGhlcnRydXN0bWFuYWdlci5s b2Nhbc9jcmxzL2VlZjgyNjViLWI1MDYtNGM2Yi04
YWRkLWU5YTliOTY3ZjI3OS5j cmwwDQYJKoZIhvcNAQELBQADggIBAAjxjAmuVS9Q9LV9lXpJ
7dRPWE35Xa76AW4I fEvrNlm2z9aBd5yzXFgeRv3jYK354fKzMjLAbShu98CkK56DEz0ThgY4
a14kiXCE Heko6ZGbW2MORfa8yD/S0YD3Z6Cq0bQth4iouHFTv17zYfZFEojk9Aq9HrT4hoTh
of8K8ddT0aufy298WN9y2vOL7XrpVZbwRydTtzazBrsjuJgJFY27P7jnp1ntOpey nZI6Oa2z
Y3cb3EPQkW7LjzRoXbUDx6iZfiLr7sT2se4z6EeVPscjaDrM1ZlZAB5T gY2UtruT/Bbyk5lT
oAI21kEBguie2Z5wzCnWqAx+F/BYPnfFI9ZdlLsF4Tf5iek8 90PsBFFO0XAefMsenwBjokhr
UQ/eBzurrMe4ILhgFNHXDpQiwTj0Ljph3qDQ3m43 P8IXHtZg2Wy5MYbgn58IxjG2rABo5Nwr
rc5whWDvA6U9vO9KpBtAK5wt8JxRT8jH A9L3uKdlwJ1r8POJ2MKZl0ooWcP0NC/uy5aTaIcW
MgMmrFA5gaLrXdKrha5MDUKP UJ6mVJZRxCzCoRkvLz3/zb0jvu6Ldlpnbc1Xc3fkyfe7JetZ0
APNwi9PqOrRrwkcg 3ODrtiyHhbmIEjZMZJjYEFF+jegZPKbvFDCGW9T/h9ny3RZ+cSHKBe
x+jB1/dhNF qyjekzHR
-----END CERTIFICATE-----
```

3. (Optional) Click **Test Credentials** to check whether the connection is configured correctly. If the test is successful, the status is **OK** else the status is **Fail**.
4. Click **Next** to move to the **Add Products** screen of the Add Connection wizard.

Configuring CM Connections using ksctl

The subsequent sections explain the steps to configure the CM/external CM connections.

- Create External CM Server
- Establish Connection between Client CM and External CM

Create External CM Server

On the client CM, create an external CM server using the below command.

Syntax

```
ksctl connectionmgmt external-cm create --name <Connection-Name> --nodes-json-file <Json-File> --trusted-ca <Certificate>
```

Here,

- `nodes-json-file` contains the hostname/IP address of the external CM.

Note

If the external CM is configured to use a custom port (that is, other than 443), then append the custom port to the hostname/IP address. For example, specify `<hostname or IP address>:<custom-port>` in the `nodes-json-file` file.

- `trusted-ca` is the root CA of the external CM.

Example Request

```
kscctl connectionmgmt external-cm create --name "ext-cm" --nodes-json-file ./nodes.json --trusted-ca ./trustedCA.json
```

nodes.json

```
[
  {
    "hostname": "3.85.100.8"
  }
]
```

trustedCA.json

```
{
  "certificate":"-----BEGIN CERTIFICATE-----\nMIIFrjCCA5agAwIBAgIQQMRYHIyAt
HEkSVPPpLSLGljANBgkqhkiG9w0BAQsFADBa\nmQswCQYDVQQGEwJVUzELMAkGA1UECBMCVFgxZDZANBgNV
BACTBkF1c3RpbjEPMA0G\na1UEChMGVGVhbnhGVzMRwwGgYDVQQDEwNDaXBoZXJUCnVzdCBSb290IENBMB4
XDTIz\nnmDEyMzExMzk1NFoXDTMzMDEyMTEzMzk1NFowWjELMAkGA1UEBhMCVVMxZCZAJBgNV\nnBAGTA1RY
MQ8wDQYDVQQHEwZBdXN0aW4xZDZANBgNVBAoTB1RoYWxlczEcMBoGA1UE\nnAxMTQ21waGVyVHJ1c3QgUm9
vdCBDQTCcAiIwDQYJKoZIhvcNAQEBBQADggIPADCC\nnAgoCggIBAMkCwTjYYILCsz+Goh4vB7AUUnwKfZA
HRqySAMBUIkzPsk5wMrgXwD2kc\nnMFdTeyda8o9qp6vbTnkeM50SmC7Zn+cEeIGnj1cpAw3gVEuZex/mF
cu7MYJsoS7m\nn9J/RqmaaLugUxoM2GbrIzaHREJVOsR0Hp9RH6Fue0fxgGWUVTPtEgSrzaPeVw/y01\nnLZ
UW56FPeT67CUmAfmgTSvmb619EKZ7UP8KwgMXkK7JASCD+NwGoUmMqAF7uRE1u\nneTJQ6dj2vdASrVKV
WuU2Kobn7SqTTZcvLcYixis1zHcJ2/mOaXE65rKtb99hmFH\nnWtg1U81Em2TkQkXq29YFB33ED90fYY+i
4j1FkT+qBZ+f2kr8Tk4pi8EQHBS9/1O2\nntO/T8L6I8zUIxgQqPJXlaasaJ/NURkPMEFn6zBWbUAJHOCd
Y/S396n29feYC0sGI\nnm2VZdXB/SOWoSNyEEo/322+gKF6wjdd8S6H6gt48V9hWT+3Cp/XNuopXATVa5Yb
NQ\nnGiB30j1J1Bw5lpon88rfL5G5ggE8WT/RbVfx/5U0Pl3bFZNk1DzLs5MVCFeJPSfq\nnNZQZW2Jp7cA
ZhWV1AFn0UARM7o8R84pxEole+7fngDuInev9u8CB3Igbqp++uK+g\nn4OvhjmsN3uZJIXHF9hSGqGq3kZ
58s2yNmnGSY2s2p4I1M18BKhzfAgMBAAGjCDBu\nnMA4GA1UdDwEB/wQEAWIBBjAPBgNVHRMBAf8EBTADA
QH/MBOGA1UdDgQWBBT3StjC\nn79rueuMtXSUGv64f3mctczASBgNVHREETAJtgSF0ZWNobmljYWWuc3Vw
cG9yZEB0\nnaGFsZXNncm91cC5jb20wDQYJKoZIhvcNAQELBQADggIBABTnVfZEHQwVjKLUB0niJ\nnPLt9S
6B8t1KRw4tbDvskyPL9Dr1fCicIUao7QGwVvPe0wyEE/p26i/DNFy0RsdVl\nnoajg6L6b7ss6rmsjMlLWs
4FhFMnmwlyVjySJ5oE+7naSUCTm7oYNJ8JLQgayH2Ia\nnXX/iY6KaR4xsWuNTH4iC3zfOa6zL4cM4AiX
15ulqBBtu7Ai11qmr2uws+Fl+zRBA\nn9ResGp1sXtf6mEPUDdDo6Y6r3BsK8hcndP3+prxFZMdbw++515
tb8MfsFMHHGuxD\nn7MmLf6Vu5zPusc7wacoK1+h0fq0j1iUBppbGn12hDu54Fp35CBHRI9btvALyHVW
```



```
nvBrz+7hdSclvVFNA84RQOtEwxS+StXEgP+7JGv69a7r8fWF6b057MRp5ixXdo70G\ndRxZuyxg LHwbPH
UGLFC3LKJd3OGBFO06nDUWqeaql215LzFO8/DWcPK//XuAeKfE\ncF3dc0WqBbiK3czy7gYGMa2KEL79/
0ZmTF/e4efe7j9vfqByPocMZWJWViIMpspr\nYDFr5KhKyLVpmUX7ZEpVA0JsZoWlja6kjgblF3Q3XcvK
8o3oaWKRKuc3eYzq+hCU\nHXrwXOHLfVfw7nQNgQjHFgumhPo+rRGIDr8hJzWAc/8oMeEreL5ZJF8R583
j31lw\n5GfTsb1JgSD0+NRYLa73bryL\n-----END CERTIFICATE-----"
    }
}
```

Example Response

```
{
  "id": "33a9ba2a-638a-46ef-b13d-521634c22d0d",
  "uri": "kylo:kylo:connectionmgmt:external-cm:test3-33a9ba2a-638a-46ef-b13
d-521634c22d0d",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-02-21T07:02:43.483599381Z",
  "name": "ext-cm"
  "nodes": [
    {
      "id": "47096049-c256-4e5e-b2ff-2ac9eac0731d",
      "uri": "kylo:kylo:connectionmgmt:cm-node:test3-47096049-c
256-4e5e-b2ff-2ac9eac0731d",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-02-21T07:02:43.488838252Z",
      "hostname": "3.85.100.8",
      "external_cm_id": "33a9ba2a-638a-46ef-b13d-521634c22d0d"
    }
  ],
  "trusted_ca": [
    {
      "id": "ac663bce-cc96-4d52-8dd0-16a894513d12",
      "uri": "kylo:kylo:connectionmgmt:cm-ca:test3-ac663bce-cc9
6-4d52-8dd0-16a894513d12",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-02-21T07:02:43.491888443Z",
      "certificate": "-----BEGIN CERTIFICATE-----\nMIIFrjCCA5ag
AwIBAgIQQMRYHIyAtHEkSVPPpLSLG1jANBgkqhkiG9w0BAQsFADBa\nMQswCQYDVQQGEwJVUzELMAkGA1U
ECBMCVFGxDbANBgNVBAcTBkF1c3RpbjEPMA0G\na1AUEChMGVGHhbGVzMRwwGgYDVQQDEwNDaXBoZXJUCn
VzdCBSb290IENBMB4XDTEz\nmDEyMzExMzk1NFoXDTMzMDEyMTEzMTNFowWjELMAkGA1UEBhMCVVMx
CzAJBgNV\nbAgtAlRyMQ8wDQYDVQQHEwZBdXN0aW4xDzANBgNVBAoTB1RoYWxlczEcMBoGA1UE\nnAMTQ2
lwaGVyVHJlc3QgUm9vdCBDQTCcAiiwDQYJKoZIhvcNAQEBBQADggIPADCC\nnAgoCggIBAMkCwTjYYILCs
z+Goh4vB7AUwKfZAHrqySAMBUIkzPsk5wMrgXwD2kc\nnMfdTeYda8o9qp6vbTnkeM50SmC7Zn+cEeIGn
j1cpAw3gVEuZex/mFcu7MYJsoS7m\nn9J/RqmaaLugUxoM2GbrIzaHREJVOsR0Hp9RH6Fue0fxgGWUVTPe
gSrzaPeVw/yO1\nnLZUW56FPeT67CUMafmGTSvmb619EKZ7UP8KwgMXkK7JAsCD+NwGoUmMqAF7uRE1u\nn
eTJQ6dj2vdASrVKVWuU2Kobn7SqTTZcvLcYixis1zHcJ2/mOaXE65rKTb99hmFH\nnWtg1U81Em2TkQkX
q29YFB33ED90fYY+i4j1FkT+qBZ+f2kr8Tk4pi8EQHBS9/1O2\nntO/T8L6I8zUIxgQqPJXlaasaJ/NURk
PMEFn6zBWBuAJHocDY/S396n29feYC0sGI\nnm2VZdXB/SOWoSNyEEo/322+gKF6wjd8S6H6gt48V9hW
T+3Cp/XNuopXATVa5YbNQ\nnGiB30j1J1Bw5lpon88rfL5G5ggE8WT/RbVfx/5U0P13bFZNk1DzLs5MVCF
eJPSfq\nnNZQZW2Jp7cAZhWV1AFn0UARm7o8R84pxEole+7fnGDuInev9u8CB3Igbqp++uK+g\nn4Ovhjms
N3uZJIXHF9hSGGqGq3kZ58s2yNmnGSY2s2p4I1M18BKhzfAgMBAAGjCDBu\nnMA4GA1UdDwEB/wQEAwIBBj
APBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBbT3STjC\nn79rueuMtXSuGv64f3mctczAsBgNVHREEJTAjg
SF0ZWNoBmljYWwuc3VwcG9ydeB0\nnaGFsZXNncm91cC5jb20wDQYJKoZIhvcNAQELBQADggIBABTnfZEH
QwVjKLUB0niJ\nnPLt9S6B8t1KRw4tbDvskyPL9Dr1fCicIUao7QGwWVpe0wyEE/p26i/DNFy0RsdV1\nno
ajg6Ln7ss6rmsjM1LWS4FHFmnmwlyVjySj5oE+7naSUCtm7oYNJ8JlQgayH2Ian\nnXX/iY6KaR4xsWuNT
H4iC3zfOa6zL4cP4AiX15ulqBBtu7Ai1lqmr2uws+Fl+zRba\nn9REsGplsXtf6mEPudDdo6Y6r3BsK8hc
ndP3+prxFZMdbw++515tb8MfsFMHHgUxd\nn7MmLf6Vu5zPusc7wacoK1+h0fg0j1ijUBppbGn12hDu54F
p35CBHRI9btvALyHVw\nnvBrz+7hdSclvVFNA84RQOtEwxS+StXEgP+7JGv69a7r8fWF6b057MRp5ixXdo
-----END CERTIFICATE-----"
```

```
70G\ndRxZuYxg LHwbPHUGLFC3LKJd3OGBFO06nDUWqeaql215LzFO8/DWcPK//XuAeKfE\ncF3dc0WqBb
iK3czy7gYGMa2KEL79/0ZmTF/e4efe7j9vfqByPocMZWJWViIMpspr\nYDFr5KhKyLVpmUX7ZEpVA0JsZ
oWlja6kjgblF3Q3XcvK8o3oaWKRKuc3eYzq+hCU\nHXrwXOHLfVfw7nQNgQjHFgumhPo+rRGIDr8hJzWA
c/8oMeEreL5ZJF8R583j31lw\n5GFtSB1JgSD0+NRYLa73bryL\n-----END CERTIFICATE-----",
    "external_cm_id": "33a9ba2a-638a-46ef-b13d-521634c22d0d",
    "subject_dn": "CN=CipherTrust Root CA,O=Thales,L=Austin,S
T=TX,C=US",
    "expires_at": "2033-01-21T11:39:54Z"
  }
}
```

The response returns `external_cm_id`. In the above example, `33a9ba2a-638a-46ef-b13d-521634c22d0d` is the `external_cm_id`.

After the external CM server is created, you can manage these connections using `ksctl`.

Establish Connection between Client CM and External CM

This section describes the steps required to establish a connection between a client CM and an external CM.

Prerequisites

Before creating a CM connection, **register client** on external CM that includes the following steps:

1. Create CSR on client CM.

```
ksctl connectionmgmt connections csr --cn <Common-Name> --csr-outfile <Filename>
```

2. Generate registration token on external CM.

```
ksctl clientmgmt tokens create --ca-id <CA-ID>
```

Note

If you are generating the registration token with an external CA, you first need to upload the external CA. Use the following command.

```
ksctl ca externals upload --cert-infile <Filename>
```

3. Register client on external CM.

```
ksctl clientmgmt clients register --reg-token <Reg-Token> --csr <CSR>
```

The response will return the following resources:

- client id (`client-id`)
- client certificate (`client-cert-file`)

Steps

To create a CM connection on Client CM, run:

Syntax

```
ksctl connectionmgmt cm create --name <Connection-Name> --products <Products-Name> --meta <Key-Values> --client-cert-file <Client-Certificate-File> --client-id <Client-ID> --external-cm-id <External-CM-ID>
```

Here,

- `name` - unique name for CM connection
- `client-cert-file` - client certificate (generated in the [Prerequisites](#) section)
- `client-id` - client-id (generated in the [Prerequisites](#) section)
- `external-cm-id` - external-cm-id (generated in the [Create External CM Server](#) section)

Note

If the external CipherTrust Manager is in a clustered environment, the external CipherTrust Manager administrator adds the client to the group (Cluster Info Viewers), so that client CipherTrust Manager can read the cluster information.

Example Request

```
ksctl connectionmgmt cm create --name "connection_1_CM" --products "cckm" --client-cert-file ./clientCert.pem --client-id "008c52be-4d26-40b2-a41f-a86da534107e" --external-cm-id "33a9ba2a-638a-46ef-b13d-521634c22d0d"
```

Example Response

```
{
  "id": "fc01d709-7b2a-4235-87fe-6cdd20f1f22a",
  "external_cm_id": "33a9ba2a-638a-46ef-b13d-521634c22d0d",
  "uri": "kylo:kylo:connectionmgmt:connections:connection-1-cm-fc01d709-7b2a-4235-87fe-6cdd20f1f22a",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-04-19T10:20:16.124632296Z",
  "updatedAt": "2023-04-19T10:20:16.122916547Z",
}
```

```
"service": "CM",
"category": "KeyManager",
"last_connection_ok": null,
"last_connection_at": "0001-01-01T00:00:00Z",
"name": "connection_1_CM",
"products": [
    "cckm"
],
"client_certificate": "-----BEGIN CERTIFICATE-----\nMIIFGzCCA2ugAwIBAgIRAIYB4
m8ecQQtl4eP3xNseBUwDQYJKoZIhvcNAQELBQAww\nWjELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAIRYMq8w
DQYDVQQHEwZBdXN0aw4xDzAN\nBGNVBAoTB1RoYWxlczEcMBoGA1UEAxMTQ2lwaGVyVHJlc3QgUm9vdCB
DQTAEFw0y\nnMZAOMTGxmDEONDJaFw0yNTAOMTGxmDEONDJaMITGMQswCQYDVQQGEwJVUzELMAKGA1UE
CBMCTUQxEADAObGNVBACB0JlbGNhbXAxFtATBgNVBAOTDFRvYWxlcyBHcm91\nncDEMMAoGA1UECXMdUm5
EMREwDwYDVQQDEWhreWxvLmNvbTEtMCsGA1UELhMkMDA4\nYzYyYmUtNGQyNi00MGIIyLWE0MWYtYTg2ZG
ElMzQxMDdlMIIBIjANBgkqhkiG9w0B\nAQEFAAOCAQ8AMIIBCgKCAQEAAur6M7t4Z7cGTMQP0qJik4LgsA
0Erf1RuJWwVDElJ\nAV3B1BL3mlnPrZm1Mu6U5stU5eTLlT+HE+nRxWYPJ4Wed5GtZ/cBrWh4Z7IcP5W
1\nq9qXiOIYL7hKnfGRHGJ4+WpuetzR6HYoZSX/G5svEyD+sDi/4073Flvx8pIkD0WR\nNH4a74EQO7y2S
sDKGwnndQFJCQJZlizukBBQENFi+pTU70AdDiEazkt6/sIxFeVS\nPhjIcq/HW9lzGYxGtIfcznsTQAY
OjKn6wc/hC3c33ICNRQPg+MPpyjZJswyGdju\nnnlZNNQXRyrDV2HS+DOOPJS+bTxEx8v68ejeHbuLvskO
UQqwIDAQABO4IBCDCCAQww\nDgYDVR0PAQH/BAQDAgoIMBMGA1UdJQMMAoGCCSGAQUFBwMCMawGA1UdE
wEB/wQC\nMAAWHwYDVR0jBBgwFoAUWdyFoqjVth/meouD25TsPe1pfEwTgYDVR0RBECwRYIR\nNKi50aG
FsZXNncm91cC5jb22CESoudGhhbGVzZ3JvdXAubmV0gRdjB250YWN0QHRo\nYXlwclc2dyb3VwLmNvbYcEA
QEBATBeBgNVHR8EVzBVMFogUaBPhklodHRwoi8vY2lw\naGVydHJlc3RtYW5hZ2VyLmxvY2FsL2NybmhmV
NDA2MmVlNWmtZTMwNi00ODkzLWI5\nYjItMzMzZWZiY2NlYjY4LmNybdANBgkqhkiG9w0BAQsFAAOCAgE
AVFXx69fNsMoB\nCLEflrfaAB2uHJRvdjslnkWrNT5fwZD37ig85w70aYLH3yfdglHP4XilKEqMoX2E\nn
kC2NFSSa9JpwpYV3zkZk9Q6oJBxUwg9ploxF3KJ/t6TkHR53ZY4qhak+fVXFjrBq\nnHAoxusc4inIGYPc
6x9OowaQJJzBJTHICoxOux4VD9ihTrOGrMCTSRkG+WFBU3Zp\nn2DcwufzoSTNV/I8xHhqfSHiaTaeOJo
zG2Aoq3P06AT0JaphbJltJBtNu4ceA+/h\nnJjwsClXuc6fHu+OQL2MxOC8DrCeP2tQc95KWrvb7WJJCe
Vu6To+ztoOm/Jc6rrne\nnmuD9Tnv/cmDfjPQk1Evf/IsL+baHpEDf9399ZI6w8WvU45IanYYBSMtRaHn
Kesa\nn8klbjWHznWvG4ZtbOawX0q5LMkB+lEe/8YxDvy3RlCRcGQGh+sPwgjOL/XzdAvK6\nne9mjs2nzp
F37Ns4UiaHUo44XZJDikes8cqUpJhiqjPo/YPPeUqMGjucpCeIqKlh\nnZoio6d8h3WddvsIIv+ygkAnU
eVtiq0M2lwUKtKEBo+i8iXL3rctGCyPF5hQkRXWS\nnu5hiVl9frMKTuE4X7FQIEPrZlcPwsnfcsMpZf5
Mwbobwm/Ych9rrqWanYNaq+PVg\nn7U4C7GwJZYDD3349pO7g4CasiYSCorI=\n-----END CERTIFICAT
E-----\n",
"client_id": "008c52be-4d26-40b2-a41f-a86da534107e",
"nodes": [
    {
        "id": "efec26ad-f9f1-44d0-850f-be2cc3ed54d0",
        "uri": "kylo:kylo:connectionmgmt:cm-node:test2-efec26ad-f9f1-44d0-850
f-be2cc3ed54d0",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2023-04-19T10:03:02.532367Z",
        "hostname": "54.173.227.108",
        "external_cm_id": "33a9ba2a-638a-46ef-b13d-521634c22d0d"
    }
],
"trusted_ca": [
    {
        "id": "fe92910f-98c4-4dd4-bdf0-f39a2160d5b5",
        "uri": "kylo:kylo:connectionmgmt:cm-ca:test2-fe92910f-98c4-4dd4-bdf
0-f39a2160d5b5",
        "account": "kylo:kylo:admin:accounts:kylo",
        "createdAt": "2023-04-19T10:03:02.533998Z",
        "certificate": "-----BEGIN CERTIFICATE-----\nMIIFFrzCCA5egAwIBAgIRAIDg
uiVeBLRGPRQUeERSkYEwDQYJKoZIhvcNAQELBQAww\nWjELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAIRYMq8w
DQYDVQQHEwZBdXN0aw4xDzAN\nBGNVBAoTB1RoYWxlczEcMBoGA1UEAxMTQ2lwaGVyVHJlc3QgUm9vdCB
BDQTAEFw0y\nnMZAOMTcwMjE5MTFaFw0zMzA0MTUwMjE5MTFaMFoxCzAJBgNVBAYTAlVTMQswCQYD\nnVQQ
IEWJUWDEPMAG1UEBxMGQXVzdgLUmq8wDQYDVQQKEwZUaGFsZXNMxHDAaBgNV\nnBAMTEONpcGhlclRydX
N0IFJvb3QgQ0EwgqgiIMA0GCSqGSIb3DQEBAAUAA4ICDwAw\nngqIKAoICAQCwz0QtPo+cFY2qlDdsuXMHk
```

```

rVgv0bMGf9eXqgLOQ+8PsAoDvLg43AU\nAx7dDKvd73L1DsQmOwqeslx3bS3deBR2zyCfhy6tPlilL2iIL
sPuzJZqTzyu0+Sfs\nrnwplwM1oZE0E0QkkCM8vx017gn2plhx8CU9cu/s9Dbe1fcBFbqLURkDN3U3pTe
o\nmACHjTQSVQwN6ZcpIEdAycRaUL9+AKt6/k9DjusQpNJkMtJjY4OYp5j6JGAgFdcK\nufzjsVzb16EE
Mu6yyXi/P/QXT8+7zwJdaocT80kMRrnIYvmLwjCNo3kZpvXLsE1G\nHPgNbzdQgVT2afLeGKtMI1sRdo
KII8gvEBdeO9NMMyD/+VeY0Yop+hOSqPP/XQbF\ndtVengKUG/6mq3Z6cz92R0R/Sw9KNnDixFa0cy4KAZ
hS2mn9E8HrTfwzn6mZiWbl\n3zehiyMN2eyOB1sAfJwL04D6bZYDWD1Nnf/XXLChhkzUFImeA5QkYiPg0
ZgNOGaD\nkveQbaog26ui4REaJm0cTjOnzIyRwl14iQ0OdDlRbcasgpDFMMNzsYEFmunPfLxK\n90+rgz
lXXcowl7ZYIYwjcoc1IqPPNhjx2pHlkuYe8XXusRsuBEuHmg3eL+SpfsHE\nHnJn64ddofDqSGZ/4lJGY
vcCVTE02jAa5olhOC16fcV+y44xc/yZwwIDAQABo3Aw\nnbjAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/
BAUwAwEB/zAdBgNVHQ4EFgQUWdYF\nnoqvVth/meouD25TsPelpfEwLAYDVR0RBCUwI4EhdGVjaG5pY2F
sLnN1cHBvcnRA\nndGhhbGVzZ3JvdXAuY29tMA0GCSqGSIb3DQEBCwUAA4ICAQBz8a2KZo1MnkmD5w4I\nH7XpHsQ6cGlzYH72P5jik+6EHEkHHPXITvRj8nAW7nsXAZPLjjaJLJy46r94GYZN\neIY7PGgoRTdwRsS
TPTmTc6cw/ZgJd5hWukKe0D1C15Lo7k87KV3CVh9aH28R1sEe\nFxyY52b44qa/dZFAxUU5meXYFI/5lrz
MvDZ7gTVQWhR0FFz5b8OcG0HupJQe/xson\nnpERSDYLM4R6eDQ/kqQelFDEa3s8EUob3BQtPYWNMRPow
lVOWir74XtXk9K4cPml\nnz3tYZm4CqPLNtDiRmp3Qob/5EXzkgD7HdVzY7UkcZg5vTGZhIxHPZ4dCDE1x
gEBS\nn2IIP005IEp3LBG9bqps4UlfmN7MjQf2mGlAoLhHA0mvVosAeawRmPQh8Ydvtk6mQ\n9HxmMk+Wg
OVu06V/eYocXX0D7DYwJXWreFjRcpCtnpe0zOyzEdic025NoCM8yNa1\n3b889t5aXYFeZEI4bHu7IKYh
0poGQyIPsN775UdmU3QU1XfghLx7s+faeoo0lt18\ncnfbvrwZm++85AwblcaoOjuEeJj2h6xv5eRDpzC
6F57pka80hNlmhCbMmbQFgjre\nnveC9pkn8Yf79RYC3HzRGgsOnoEdadlmibfF0M018M21+aEubChK1Bt
b1Z6F4Azje\nn8RccyyfNBjhXQa73QAoaKH02A==\nn-----END CERTIFICATE-----",
    "external_cm_id": "4df54fb4-d3e6-432d-a917-4b587b180b05",
    "subject_dn": "CN=CipherTrust Root CA,O=Thales,L=Austin,ST=TX,C=US",
    "expires_at": "2033-04-15T02:12:11Z"
  }
]
}

```

After the CM connection is created, you can manage these connections using `ksctl`.

Managing External CM Server Connections using ksctl

The following operations can be performed:

- Get/Delete an external CM Server
- List all external CM Server connections
- Add/delete/get/update a node in an existing external CM Server
- Add/delete a trusted CA in an existing external CM Server

Getting Details of an External CM Connection

To get details, run:

Syntax

```
ksctl connectionmgmt external-cm get --id <Connection-Name/ID>
```

Example Request

```
{
    "id": "33a9ba2a-638a-46ef-b13d-521634c22d0d",
    "uri": "kylo:kylo:connectionmgmt:external-cm:test3-33a9ba2a-638a-46ef-b13d-521634c22d0d",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2023-02-21T07:02:43.483599Z",
    "name": "ext-cm"
    "nodes": [
        {
            "id": "47096049-c256-4e5e-b2ff-2ac9eac0731d",
            "uri": "kylo:kylo:connectionmgmt:cm-node:test3-47096049-c256-4e5e-b2ff-2ac9eac0731d",
            "account": "kylo:kylo:admin:accounts:kylo",
            "createdAt": "2023-02-21T07:02:43.488838Z",
            "hostname": "3.85.100.8",
            "external_cm_id": "33a9ba2a-638a-46ef-b13d-521634c22d0d"
        }
    ],
    "trusted_ca": [
        {
            "id": "ac663bce-cc96-4d52-8dd0-16a894513d12",
            "uri": "kylo:kylo:connectionmgmt:cm-ca:test3-ac663bce-cc96-4d52-8dd0-16a894513d12",
            "account": "kylo:kylo:admin:accounts:kylo",
            "createdAt": "2023-02-21T07:02:43.491888Z",
            "certificate": "-----BEGIN CERTIFICATE-----\nMIIFRjCCA5agAwIBAgIQQMRyHIYatHEkSVPPtLSLGL1jANBgkqhkiG9w0BAQsFADBa\n\\nmQswCQYDVQQGEwJVUzELMAkGA1UECBMCVFgxZDZANBgnVBACTBkFlc3RpbjEPMAOG\n\\na1UEChMGVGHhbGVzMRwwGgYDVQQDExNDaxBoZXJUCnVzdCBSb290IENBMb4XDTIz\n\\nMDEyMzExMzE1NFoXDTEzMDEyMTExMzE1NFowWjELMAkGA1UEBhMCVVMxCzAJBgNV\n\\nBAGTAIRYMq8wDQYDVQQHEwZBdXN0aW4xDzANBgNVBAOTBlRoYWxlczEcMBoGA1UE\n\\nAaxMTQ2lwaGVyVHJlc3QgUm9vdCBDQTCCAiIwdQYJKoZIhvcNAQEBBQADggIPADCC\n\\nAgoCggIBAMkCWtjYYILCsZ+Goh4vB7AUUnwKfZAHRqySAMBUIkzPsk5wMrqXwD2kc\n\\nmFdTeYda8o9qp6vbTnkeM50SmC7Zn+cEeIGnj1cpAw3gVEuzex/mFcU7MYJSos7m\n\\n9J/RqmaaLugUxoM2GbrIzaHREJVOsR0Hp9RH6Fue0fxgGWUVTPegSrzaPeVw/yOl\n\\nLZUW56FPet67CUmAfmGTSymb619EKZ7UP8KwgMXkK7JASCD+NwGoUmMqAF7uRElu\n\\netTJQ6dj2vdASrVKVWuU2Kobn7SqTTZcvLcYixis1zHcJ2/mOaXE65rKTb99hmFH\n\\nwTgt1U81Em2TkQkXq29YFB33ED90fYY+i4jlFKt+qBZ+f2kr8Tk4pi8EQHBs9/102\n\\nto/T8L6I8zUIxgQqPJXlaasaJ/NURKPMEFn6zBWbUAjhOCdY/S396n29feYC0sGI\n\\nm2VZdXB/SOWoSnyEEo/322+gKF6wj2d8S6H6gt48V9hWT+3Cp/XNuopXATVa5YbNQ\n\\ngIB30j1JLBw5lpOn88rfL5G5ggE8WT/RbVfx/5U0Pl3bFZNklDzL5MVCF eJPsfq\n\\nnNZQZW2Jp7cAZhwV1AfNUARm7o8R84pxEOle+7fnGDuInev9u8CB3Igbqp++uK+g\n\\n40vhjmsN3uzJIXHF9hSGqGq3kZ58s2yNmGSY2s2p4I1M18BKhzfAgMBAAgJCDBu\n\\nMA4GA1UdDwEB/wQEAWIBBjAPBgNVHRMBAf8EBTADAQH/MBOGA1UdDgQWBBS3Stjc\n\\n79rueuMtXSUGv64f3mctczAsBgNVHREEJTAjgSF0ZWNobmljYWWuc3VwcG9ydEB0\n\\naGFsZXNNcm91cC5jb20wDQYJKoZIhvcNAQELBQADggIBABTnfZEHQwVjKLUBoniJ\n\\nPLt9S6B8t1KRw4tbDvskyPL9Dr1fCicIUao7QGwWVpe0wyEE/p26i/DNFyORsdVl\n\\noajg6Ln7ss6rmsjMLLWS4FHfMnmwlyVjySJ5oe+7nasUCTm7oYNJ8JlQgayH2Ian\n\\nXX/iY6KaR4xsWuNTH4iC3zfOa6zL4CP4AiX15ulqBBtu7Aillqmr2uws+F1+zRBa\n\\n9REsGplsXtf6mEPUDdo6Y6r3BsK8hcndP3+prxFZMdww++515tb8MfsFMHHGUxD\n\\n7MmLf6Vu5zPusc7waccK1+h0fg0jj1ijUBppbGn12hDu54Fp35CBHRI9btvAlYHVw\n\\nvBrzz+7hdSc1vVFNA84RQOtEwxS+StXEGP+7JGv69a7r8fWF6b057MRp5ixXdo70G\n\\ndRxZuyxgLHwbPHUGLFC3LKJd3OGBFO06nDUWqeal215LzFO8/DWCpk//XuAEkFe\n\\ncF3dc0WQBbiK3czy7gyGma2KEL79/OzmTF/e4efe7j9vfqByPocMZWJWViImpspr\n\\nyDFr5KhKyLVpmUX7ZEpVA0JsZoWlja6kjgblF3Q3XcvK8o3oaWKRKuc3eYzq+hCU\n\\nhXRwXOHLfvfw7nQNqQjHFGumhPo+rRGIDr8hJzWA c/8oMeEreL5ZF8R583j31lw\n\\n5GftSB1JgSD0+NRyLa73bryL\n\\n-----END CERTIFICATE-----",
            "external_cm_id": "33a9ba2a-638a-46ef-b13d-521634c22d0d",
```

```

    "subject_dn": "CN=CipherTrust Root CA,O=Thales,L=Austin,S
T=TX,C=US",
    "expires_at": "2033-01-21T11:39:54Z"
  }
]
}

```

Deleting an External CM Connection

To delete a connection, run:

Syntax

```
ksctl connectionmgmt external-cm delete --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt external-cm delete --id "33a9ba2a-638a-46ef-b13d-521634c22d0d"
```

There will be no response if CipherTrust Manager connection is deleted successfully.

Getting List of External CM Connections

To list all the connections, run:

Syntax

```
ksctl connectionmgmt external-cm list
```

Example Request

```
ksctl connectionmgmt external-cm list
```

Example Response

```

{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "id": "33a9ba2a-638a-46ef-b13d-521634c22d0d",
      "uri": "kylo:kylo:connectionmgmt:external-cm:test3-33a9ba
2a-638a-46ef-b13d-521634c22d0d",

```

```
"account": "kylo:kylo:admin:accounts:kylo",
"createdAt": "2023-02-21T07:02:43.483599Z",
"name": "ext-cm"
"nodes": [
{
"id": "47096049-c256-4e5e-b2ff-2ac9eac0731d",
"uri": "kylo:kylo:connectionmgmt:cm-node:test3-47096049-c256-4e5e-b2ff-2ac9eac0731d",
"account": "kylo:kylo:admin:accounts:kylo",
"createdAt": "2023-02-21T07:02:43.488838",
"hostname": "3.85.100.8",
"external_cm_id": "33a9ba2a-638a-46ef-b13d-521634c22d0d"
},
],
"trusted_ca": [
{
"id": "ac663bce-cc96-4d52-8dd0-16a894513d12",
"uri": "kylo:kylo:connectionmgmt:cm-ca:test3-ac663bce-cc96-4d52-8dd0-16a894513d12",
"account": "kylo:kylo:admin:accounts:kylo",
"createdAt": "2023-02-21T07:02:43.491888",
"certificate": "-----BEGIN CERTIFICATE
E-----\nMIIFrjCCA5agAwIBAgIQQMRyHIYAtHEkSVPPtLSLGljANBgkqhkiG9w0BAQsFADBAnMQswCQYDVQQGEwJVUzELMAkGA1UECBMCFVgxDzANBGNVBACTBKf1c3RpbjEPMAO0G\nA1UEChMGVGhhbGVzMRRwwGgYDVQQDExNDaXBoZXJUCnVzdCBSb290IENBMDB4XDTIz\nnMDEyMzExMzk1NFoXDTMzMDEyMTEzMzk1NFowWjELMAkGA1UEBhmCVVMxCzAJBgNV\nnBAGTAIRYMq8wDQYDVQQHEwZBdXN0aW4xZDZANBGNVBAAOTBlRoYWxlczEcMBoGA1UE\nnAxMTQ21waGVyVHJlc3QgUm9vdCBDQTCCAIwDQYJKoZIhvcNAQEBBQADggIPADCC\nnAgoCggIBAMkCWtjYYILCsZ+Goh4vB7AUnwKfZAHRQySAMBUIkzPsk5wMrqXwD2kc\nnMFdTeyda8o9qp6vbTnkeM50SmC7Zn+cEeIGnj1cpAw3gVEuZex/mFcU7MYJsos7m\nn9J/RqmaaLugUxoM2GbrIzaHREJVOsR0Hp9RH6Fue0fxgGUWVTpEgSrzaPeVw/yOl\nnLZUW56FPet67CUmAfmGTSymb619EKZ7UP8KwgMXkK7JASCD+NwGoUmMqAF7uRElu\nnnetJTQ6dj2vdASrVKVWuU2Kobn7SqTTzcVlcyixis1zhCcJ2/mOAxe65rKTb99hmFH\nnWtg1U81Em2TkQkXq29YFB33ED90fYY+i4jlFkT+qBZ+f2kr8Tk4pi8EQHBS9/1O2\nntOt/T8L6I8zUIxgQqPJXlaasaJ/NURkPMEFn6zbWBuAjHOcDY/S396n29feYCOSGI\nnm2VZdXB/SOWoSnyEEo/322+gKF6wjds6H6gt48V9hWT+3Cp/XNuopXATVa5YbNQ\nnGiB30jlJ1Bw5lpon88rfL5G5ggE8WT/RbVfx/5U0Pl3bfZNklDzLs5MVCFEJPSfq\nnnNZQZW2Jp7caZhWVlAfNUARm7o8R84pxEole+7fnGDuInev9u8CB3Igbbq++uK+g\nn4OvhjmsN3uzJIXHF9hSGqGq3kZ58s2yNmngSYs2p4tlM18BKhzfAgMBAAgjcdBu\nnMA4GA1UdDwEB/wQEAWIBBJAPBGNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBT3STjC\nnn79rueuMtXSUGv64f3mctczAsBgNVHREEJTAjgSF0ZWNobmljYWwuc3VwcG9ydEB0\nnaGFsZXNNcm9lc5jb20wDQYJKoZIhvcNAQELBQADggIBABTNfZEHQwVjKLUBoniJ\nnPLt9S6B8t1KRw4tbDvskyPL9Dr1fCicIUao7QGwWVpe0wyEE/p26i/DNFyORsdVl\nnoajg6Ln7ss6rmsjMLLWS4FHfMnmwlvyvjySJ5oe+7naSUctm7oyNJ8JlQgayH2Ia\nnnXX/iY6KaR4xsWuNTH4iC3zfOa6zL4cP4AiX15ulqBBtu7Aillqmr2uws+Fll+zRba\nnn9ResGplsXtf6mEPudDdo6Y6r3BsK8hcndP3+prxFZMdbw++515tb8MfsFMHHGuxD\nn7MmLf6Vu5zpusc7wacoK1+h0fg0jlijUBppbGn12hDu54Fp35CBHRI9btvALyHVwnvBrz+7hdScLvVFNA84RQOtEwxS\nStXEgP+7JGv69ar8r8WF6b057MRp5ixXd070G\nndRxZuyxgLHwbPHUGLF3LKJd30GBFO06nDUWgeaqL215LzFO8/DWCpK//XuAekFe\nncF3dc0WqBbiK3czy7gYGMa2KEL79/OzmTF/e4efe7j9vfqByPocMZWJWViImpspr\nnYDFr5KhKyLVpmUX7ZEpVA0JsZoWljA6kjgb1f3Q3XcvK8o3oaWKRKuc3eYzq+hCU\nnHXrwXOHLfvfw7nQNNgQjHFgumhPo+rRGIDr8hJzWAc/8oMeEreL5ZJF8R583j31lw\nn5GFtSB1JgSD0+NRYLa73bryL\nnn-----END CERTIFICATE-----",
"external_cm_id": "33a9ba2a-638a-46ef-b13d-521634c22d0d",
"subject_dn": "CN=CipherTrust Root CA,O=Thales,L=Austin,ST=TX,C=US",
```



```

        "expires_at": "2033-01-21T11:39:54Z"
    }
}
},
{
    "id": "15548b96-6463-4950-b0fe-f9116b543301",
    "uri": "kylo:kylo:connectionmgmt:external-cm:test2-15548b
96-6463-4950-b0fe-f9116b543301",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2023-02-16T10:38:39.602117Z",
    "name": "ext-cm"
    "nodes": [
        {
            "id": "33158e1f-eb1a-407a-b376-7450a8959e
5e",
            "uri": "kylo:kylo:connectionmgmt:cm-nod
e:test2-33158e1f-eb1a-407a-b376-7450a8959e5e",
            "account": "kylo:kylo:admin:accounts:kyl
o",
            "createdAt": "2023-02-16T10:38:39.604913
Z",
            "hostname": "54.227.55.99",
            "external_cm_id": "15548b96-6463-4950-b0f
e-f9116b543301"
        }
    ],
    "trusted_ca": [
        {
            "id": "30323c67-8216-4c0e-9745-877ed20785
8c",
            "uri": "kylo:kylo:connectionmgmt:cm-ca:te
st2-30323c67-8216-4c0e-9745-877ed207858c",
            "account": "kylo:kylo:admin:accounts:kyl
o",
            "createdAt": "2023-02-20T07:53:29.270068
Z",
            "certificate": "-----BEGIN CERTIFICAT
E-----\nMIIFrjCCA5agAwIBAgIQQMRYHIyAtHEkSVPPeL
SLG1jANBgkqhkiG9w0BAQsFADBa\nMQswCQY
DVQQGEwJVUzELMAkGA1UECBMCFVgxDzANBgNVBACTBkF1c3RpbjEPMA0G\nA1UEChMGVGVGhhbGVzMWRwwGg
YDVQQDExNDaXBoZXJUCnVzdCBSb290IENBMB4XDTIz\nnMDEyMzExMzY1NFoXDTMzMDEyMTEyMzY1NFowW
jELMAkGA1UEBhMCVVMxMzY1NFoXDTMzMDEyMTEyMzY1NFowW
czEcMBoGA1UE\nAAMTQ21waGVyVHJ1c3QgUm9vdCBDQTCcA1IwDQYJKoZIhvcNAQEBBQADggIPADCC\nnA
goCggIBAMkCwTjYYILCsZ+Goh4vB7AUnwKfZAHRRqySAMBUIkzPsk5wMrgXwD2kc\nnMfdTeYda8o9qp6vb
TnkeM50SmC7Zn+cEeIGnjlcpAw3gVEuZex/mFcu7MYJsoS7m\nn9J/RqmaaLugUxoM2GbrIzaHREJVOsR0
Hp9RH6Fue0fxgGWUVTPeGsrzapeVw/yOl\nnLZUW56FPeT67CUmAfmgTsvmb619EKZ7UP8KwgMXkK7JAsC
D+NwGoUmMqAF7uRE1u\nneTJQ6dj2vdASrVKVWuU2Kobn7SqTTZcvLcYixislzHcJ2/mOaXE65rKtb99h
mFH\nnWtg1U81Em2TkQkXq29YFB33ED90fYY+i4jlFkT+qBZ+f2kr8Tk4pi8EQHBS9/1O2\nntO/T8L6I8z
UIxgQqPjXlaasaJ/NURkPMEFn6zBwBUajHocDY/S396n29feYC0sGI\nnm2VZdXB/SOWoSNyEEo/322+gK
F6wjd8S6H6gt48V9hWT+3Cp/XNuopXATVa5YbNQ\nnGiB30j1J1Bw5lpon88rfl5G5ggE8WT/RbVfx/5U0
P13bFZNk1DzLs5MVCFeJPSfq\nnNZQZW2Jp7cAZhWV1AFn0UArm7o8R84pxEole+7fnGDuInev9u8CB3Ig
bqp++uK+g\nn4OvhjmsN3uZJIXHF9hSGqGq3kZ58s2yNmnGSY2s2p4I1M18BKHzfAgMBAAGjCDBu\nnMA4G
A1UdDwEB/wQEAwIBBjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBbT3STjC\nn79rueuMtXSuGv64f3mc
tczAsBgNVHREEJTAjgSF0ZWNobmljYWwuc3VwcG9ydEB0\nnaGFsZXNncm91cC5jb20wDQYJKoZIhvcNAQ
ELBQADggIBABTnFzEHQwVjKLUB0niJ\nnPLt9S6B8t1KRw4tbDvskyPL9Dr1fCicIUao7QGgWVpe0wyEE/
p26i/DNFy0RsdVl\nnoajg6Ln7ss6rmsjMlLWs4FHfMnmwlyVjySJ5oE+7naSUCtm7oYNJ8JlQgayH2Ia
n\nnXX/iY6KaR4xsWuNTH4iC3zfOa6zL4cP4AiX15ulqBBtu7Ai1lqmr2uws+F1+zRBA\nn9RESgplsXtf6
mEPUDdDo6Y6r3BsK8hcndP3+prxFZMdbw++515tb8MfsFMHHGuxD\nn7MmLf6Vu5zPusc7wacoK1+h0fg0
j1ijUBppbGn12hDu54Fp35CBHRI9btvALyHVw\nnvBrz+7hdSc1vVFNA84RQOtEwxS+StXEgP+7JGv69a7
r8fWF6b057MRp5ixXdo70G\nndRxZuyxg LHwbPHUGLFC3LKJd3OGBFO06nDUWqeaql215LzFO8/DWcP

```

```

K//XuAeKfE\ncF3dc0WqBbiK3czy7gYGMa2KEL79/0ZmTF/e4efe7j9vfqByPocMZWJWViIMpspr\nYDF
r5KhKyLVpmUX7ZEpVA0JsZoWlja6kjgblF3Q3XcvK8o3oaWKRKuc3eYzq+hCU\nHXrwXOHLfVfw7nQNgQ
jHFgumhPo+rRGIDr8hJzWAc/8oMeEreL5ZJF8R583j31lw\n5GFtSB1JgSD0+NRYLa73bryL\n-----EN
D CERTIFICATE-----",
                                "external_cm_id": "15548b96-6463-4950-b0f
e-f9116b543301",
                                "subject_dn": "CN=CipherTrust Root CA,O=T
hales,L=Austin,ST=TX,C=US",
                                "expires_at": "2033-01-21T11:39:54Z"
                                }
                        ]
                }
        ]
}

```

Adding Node to an Existing External CM Connection

To add a node, run:

Syntax

```

ksctl connectionmgmt external-cm node add --id <Connection-Name/ID> --hostname <S
erver-Hostname>

```

Note

Nodes must be from the same CipherTrust Manager cluster.

Example Request

```

ksctl connectionmgmt external-cm node add --id "15548b96-6463-4950-b0fe-f9116b543
301" --hostname "54.227.55.29"

```

Example Response

```

{
  "id": "63566624-acdc-441d-99ed-9c6d42e55aad",
  "uri": "kylo:kylo:connectionmgmt:cm_node:test2-63566624-acdc-441d-99ed-9c
6d42e55aad",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-02-22T05:23:31.710244034Z",
  "hostname": "54.227.55.29",
  "external_cm_id": "15548b96-6463-4950-b0fe-f9116b543301"
}

```

Getting Node Details in an External CM Connection

To get node details, run:

Syntax

```
ksctl connectionmgmt external-cm node get --id <Connection-Name/ID> --node-id <External-CM-Node-ID>
```

Example Request

```
ksctl connectionmgmt external-cm node get --id "15548b96-6463-4950-b0fe-f9116b543301" --node-id "63566624-acdc-441d-99ed-9c6d42e55aad"
```

Example Response

```
{
  "id": "63566624-acdc-441d-99ed-9c6d42e55aad",
  "uri": "kylo:kylo:connectionmgmt:cm_node:test2-63566624-acdc-441d-99ed-9c6d42e55aad",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-02-22T05:23:31.710244Z",
  "hostname": "54.227.55.29",
  "external_cm_id": "15548b96-6463-4950-b0fe-f9116b543301"
}
```

Updating a Node in an External CM Connection

To update a node, run:

Syntax

```
ksctl connectionmgmt external-cm node modify --id <Connection-Name/ID> --node-id <Server-Node-ID> --hostname <Server-Hostname>
```

Example Request

```
ksctl connectionmgmt external-cm node modify --id "15548b96-6463-4950-b0fe-f9116b543301" --node-id "63566624-acdc-441d-99ed-9c6d42e55aad" --hostname "54.227.55.59"
```

Example Response

```
{
```

```
{
  "id": "63566624-acdc-441d-99ed-9c6d42e55aad",
  "uri": "kylo:kylo:connectionmgmt:cm_node:test2-63566624-acdc-441d-99ed-9c6d42e55aad",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-02-22T05:23:31.710244Z",
  "hostname": "54.227.55.59",
  "external_cm_id": "15548b96-6463-4950-b0fe-f9116b543301"
}
```

Deleting a Node from an External CM Connection

To delete a node, run:

Syntax

```
ksctl connectionmgmt external-cm node delete --id <Connection-Name/ID> --node-id <External-CM-Node-ID>
```

Example Request

```
ksctl connectionmgmt external-cm node delete --id "15548b96-6463-4950-b0fe-f9116b543301" --node-id "63566624-acdc-441d-99ed-9c6d42e55aad"
```

There will be no response if node is deleted successfully.

Adding Trusted CA in an Existing External CM Connection

To add a trusted CA, run:

Syntax

```
ksctl connectionmgmt external-cm trusted-ca add --id <Connection-Name/ID> --trusted-ca-cert-file <Trusted-CA-Certificate-File>
```

Example Request

```
ksctl connectionmgmt external-cm trusted-ca add --id "9fb8effc-56ac-49c9-9a70-2bf87f54f1c8" --trusted-ca-cert-file ./cert.pem
```

Example Response

```
{
  "id": "6e3300bb-d7e0-40c6-a8e7-ac8d4c1970ed",
  "uri": "kylo:kylo:connectionmgmt:cm-ca:test4-6e3300bb-d7e0-40c6-a8e7-ac8d4c1970ed",
}
```

```

"account": "kylo:kylo:admin:accounts:kylo",
"createdAt": "2023-04-19T10:08:57.483723806Z",
"certificate": "-----BEGIN CERTIFICATE-----\nMIIFrjCCA5agAwIBAgIQOO08kiKVKK41
1V2GCXIp7jANBgkqhkiG9w0BAQsFADBa\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCFgxDzANBgNVBAC
TBkF1c3RpbjEPMA0G\nnA1UEChMGVGVhbnBzMRwwGgYDVQQDEXdDaXBoZXJUCnVzdCBSb290IENBMB4XDT
Iz\nnMDIyNTEwMzgwNl0XDTMzMDIyMzEwMzgwNl0wWjELMAkGA1UEBhMCVVMxCzAJBgNV\nnBAGTA1RYMQ8
wDQYDVQQHEwZBdXN0aW4xZDZANBgNVBAoTB1RoYXxlc3EcmBoGA1UE\nnAAMTQ21waGVyVHJ1c3QgUm9vdC
BDQTCcAiIwDQYJKoZIhvcNAQEBBQADggIPADCC\nnAgoCggIBAOF1Ujy10hXiN3sPl0jCXGUglQM04MebP
4koqA/vaHnZWDLzmBnw+irN\nnk5jJG1I6rIAqPom8gVZR+RevxAn5XMKzwdk6tFaYtTJFwweH73TwtjHs
174fGGFu\nnkTTcLM7++IJOuULGDmChk9SQLtldmNwXx3LMqqtJIrOHu8XTdo9H1fWhpmnvIeez\nnaodEO
DNGptsPiz51Am5XcNRbPOL/c7X9XiBOJ2qSOteaX5/8RXCdz+29n2sMrBlA\nn7VtE2VN04DhAz1FjHYjw
WSC+1Pd11o7yrhjNgbEqreoh/Tlr78hLHqksFUCDE4XB\nnWFnEXdEP6/ub0oQNMGAwR1LW6av2QuR2e
Cjs8EXzXoiLV+c+2zKPtLZTO0pIP9\nn01bo+bNXyJAnF5uI2YmJTEIkqCBPaRpNt9D79BqkHSmtRWigwD
OsbSsn01Wky1kj\nnullIKFDpH+MumefkMN4yY3PlI8u/L8AKICz1+kb0aR51KpLTJqPoqjtKVjGVIDRp\nna0WJ/kXinYzdD+n/RznCtha2R3tekdPjlsWC78wEMTbBRPnF0nW8OkF/EIWgLB\nnMt5mLB8sB7zIXJ
hIWkQd3rSSmCDXWK5aVmCGvXLF/tJwYPJXFoSbkm16wdm0AHe3\nn1HTTmrKNLZRYffsDDuhDtbLVsKMAO
ObGbLMCGPTVAOKFor+zFMwvAgMBAAGjCDBU\nnMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMBAf8EBTADAQH/
MB0GA1UdDgQWBBrAw4R8\nnk9v7fsgix7nx1U1Urd7AUTAsBgNVHREEJTAjgSF0ZWNobmljYWwuc3VwcG9
ydEB0\nnaGFsZXNncm91cC5jb20wDQYJKoZIhvcNAQELBQADggIBANzyGEIYOMnB7pli3LJT\nnnkC5oAjX
ic9/9e5sYC1DE3kcwcGHrQs0kgTd974DHIJPSOjpBuZSW0giOAKetKfy\nnMr9CbiJoiFcCeRE/228BKAS
12nhH1Lwe1BO+slv5IjFQWAsfuvtgL9mC2ntrDbKF\nnQPrNBsl/xI7D83ljAEqXiPlJU38WBI3foekC2l
CfPzvNite48j7Rto8AnkzfQvUT\nnGyfpz11YadHzrP8MQF9glPE43/79Y0DSTaGBCp9rvt3wRTTgl5dYc
IfLhA2yEKs3\nnwD7jppqnl4VOo4GAA/WecerBILUfwtmXTdNiFslvM8VVtfJc9LS24H3n3TUcY3Cm\nneq
fIPmLYYB/GIav1t2pqElexWdjF/4On1aHRXZwgT3G4fPP8O03yQDWdEq7WdN90\nn+J9pbGX48iFaTqkjZ
N4M0voK4nEIXUJSwOfcU0esuBjK62uNRb5XHnB3prxOMVuy\nnb6b0+v1vjQFS1MxG1Ia7tPLUG3aFf9y0
5qXJxhAWGbhfWUCrQFPozArqQ3+2i/Xq\nnFzzRZZ8v723VXkPm+LglWVmMPM8/R4WCTSPdTONMZJerQTA
Lw9JyngkkG5C267Wb\nnFu8ishEmeJwd72JaGWCLh2kW87CFZwTzJcWUC/kAkkcRkZh9211TssA4Y2P4RZ
rE\nnI8EMLrdrDft2cX4tjQTlmzy2\nn-----END CERTIFICATE-----\n",
"external_cm_id": "9fb8effc-56ac-49c9-9a70-2bf87f54f1c8",
"subject_dn": "CN=CipherTrust Root CA,O=Thales,L=Austin,ST=TX,C=US",
"expires_at": "2033-02-23T10:38:06Z"
}

```

Deleting a Trusted CA from an Existing External CM Connection

To delete a trusted CA, run:

Syntax

```

ksctl connectionmgmt external-cm trusted-ca delete --id <Connection-Name/ID> --c
a-id <external-cm-ca-id>

```

Example Request

```

ksctl connectionmgmt external-cm trusted-ca delete --id 15548b96-6463-4950-b0fe-
f9116b543301 --ca-id 2f4f4758-ef0c-4879-ae54-24782b50f6b8

```

There will be no response if a trusted CA is deleted successfully.

Managing CM Connections using ksctl

The following operations can be performed:

- Get/delete/update a CM connection
- List all CM connections
- Test an existing CM connection
- Test the connection parameters before establishing the connection

Parameter Details

Parameter	Mandatory/Optional	Description
name	Mandatory	Unique connection name.
description	Optional	Connection description.
client-cert-file	Mandatory	File containing certificate of the client registered at the external CM.
client-id	Mandatory	ID of the client registered at the external CM.
external-cm-id	Mandatory	ID of the external CM resource that contains the Hostname and Root CA of the external CM.
json-file	Optional	Connection information passed in JSON format via a file. Parameters passed via the command line will override values in the JSON file. For example, <code>--json-file <filename></code> .
meta	Optional	Meta information in json format. For example, <code>--meta '{"color": "blue", "foo": "bar"}'</code> .
products	Optional	List of the CipherTrust products associated with the connection.

Getting Details of a CM Connection

To get details, run:

Syntax

```
ksctl connectionmgmt cm get --id <Connection-Name/ID>
```

Example Request

```
ksctl connectionmgmt cm get --id "fc01d709-7b2a-4235-87fe-6cdd20f1f22a"
```

Example Response

```

{
  "id": "fc01d709-7b2a-4235-87fe-6cdd20f1f22a",
  "external_cm_id": "4df54fb4-d3e6-432d-a917-4b587b180b05",
  "uri": "kylo:kylo:connectionmgmt:connections:connection-1-cm-fc01d709-7b2a-4235-87fe-6cdd20f1f22a",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-04-19T10:20:16.124632Z",
  "updatedAt": "2023-04-19T10:20:16.122917Z",
  "service": "CM",
  "category": "KeyManager",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "connection_1_CM",
  "products": [
    "cckm"
  ],
  "client_certificate": "-----BEGIN CERTIFICATE-----\nMIIFgzCCA2ugAwIBAgIRAIYB4m8ecQQtl4eP3xNseBUwDQYJKoZIhvcNAQELBQAw\nnWJELMAkGA1UEBhMCVVMxChZABgNVBAgTAlRlYMQ8wDQYDVQQHEwZBdXN0aW4xZDAN\nnBgNVBAoTB1RoYXxlcEcmBoGA1UEAxMTQ21waGVyVHJlc3QgUm9vdCB\nDQTAeFw0y\nnMzA0MTgxMDE0NDJhFw0yNTA0MTgxMDE0NDJhMIGTMQswCQYDVQQGEwJVUzELMAkG\nnA1UECBMCTUQxEDAOBgNVBAcTB0JlbGNhbXAxFTATBgNVBAoTDFRoYXxlcEcmBoGA1UEC\nxMDU5EMREwDwYDVQQDEwhreWxvLmNvbTEtMCsGA1UELhMkMDA4\nnYzUyYmUtNGQyNi00MGlyLWE0MmWYtYTg2ZGE1MzQxMDdlMIIBIjANBgkqhkiG9w0B\nnAQEFAAOCAQ8AMIIBCgKCAQEaur6M7t4Z7cGTMQP0qJik4LgsA0Erf1RujWwVDE1J\nnAV3B1BL3mlnPrZm1Mu6U5sTU5eTL1T+HE+nRxWYPJ4Wed5GtZ/cBrWh4Z7IcP5W\nl\nnq9qXioiYL7hKnfGRHGJ4+WpueezR6HYoZSX/G5svEYd+sDi/4073F1vx8pIkd0wR\nnH4a74EQ07y2SsDKGwnndQFJCQJZlizuKBBQeNFi+pTU70AdDiEaZkt6/sIxFeVSn\nnPhjIcq/HW9lzGYxGtIfcznsTQAyOjKn6wc/hC3c33IcNRRQPg+MPPYjZJswyGdju\nnnlZNNQxRyrDV2HS+DO0PjS+bTxE8v68ejeHbuLvskOUQqWIDAQAB\no4IBCDCCAQQw\nnDgYDVR0PAQH/BAQDAgOIMBMGA1UdJQQMMAoGCCsGAQUFBwMCMMAwGA1UdeEB/wQC\nnMAAwHwYDVR0jBBGwFoAUWdYFoqjVth/meouD25TsPelpfEwTgYDVR0RBECwRYIR\nnKi50aGFsZXNncm91cC5jb22CESoudGhhbGVzZ3JvdXAubmV0grdjb250YWN0QHRO\nnYXxlc2dyb3VwLmNvbYCEAqEBATBeBgNVHR8EVzBVMFogUaBPhklodHRwOi8vY2lw\nnaGVydHJlc3RtYW5hZ2VyLmNvbY2FsL2NybmhmVnDA2MmVlNWMTZTMwNi00ODkzLWI5\nnYjItMzMzFmNWZiY2NiYjY4LmNybDANBgkqhkiG9w0BAQsFAAOCAgEAVFXX69fNsMoB\nnCLEf1rfaAB2uHjRvdjS1nkWrNT5fWzD37ig85w70aYlH3yfdglHP4XilKEqMoX2E\nnkc2NFSsA9JpwpYV3ZkZk9Q6oJBxUwg9ploxF3KJ/t6TkHR53ZY4qhak+fVXfjrBq\nnHAoXusC4inIGYPc6x9OoWAqJjZBJTHICoxOux4VD9ilhTrOGrMCTsRkG+WFBU3Zp\nn2DcwufZoSTNV/I8xHhqfSHiaTaeOJoZG2Aoq3P06AT0JaphbJ1tJBTnua4ceA\n/h\nnJjwsC1Xuc6fHu+OQL2MxOC8DrCeP2tQc95KWrVb7WJJCeVu6To+Zto0m/Jc6rrne\nnmu5D9Tnv/cmDfjPQk1EVf/IsL+baHpEDf9399ZI6w8WvU45IanYYBSMtRaHnKesa\nn8klbjWHZnWvG4ZTbOawX0q5LMkB+1Ee/8YxvDy3RlCRcGQGH+sPwgj0L/XzdAvK6\nne9mjs2nzpZF37Ns4UiaHUo44XZJDikes8cqUpJhiqjPo/YPPeUqMGjucpCeIqKlh\nnZoio6d8h3WddvsIIv+ygkAnUeVtiq0M2lwUKtKEBo+i8iXL3rctGCyPF5hQkRXWS\nnu5hiVl9frMKTuE4X7FQIEPrZ1cPwsnfcQsMpZf5Mwbbwm/Ych9rrqWanYNAq+PVg\nn7U4C7GwJZYDD3349p07g4CasiYSCorI=\nn-----END CERTIFICATE\nE-----\nn",
  "client_id": "008c52be-4d26-40b2-a41f-a86da534107e",
  "nodes": [
    {
      "id": "efec26ad-f9f1-44d0-850f-be2cc3ed54d0",
      "uri": "kylo:kylo:connectionmgmt:cm-node:test2-efec26ad-f9f1-44d0-850f-be2cc3ed54d0",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-04-19T10:03:02.532367Z",
      "hostname": "54.173.227.108",
      "external_cm_id": "4df54fb4-d3e6-432d-a917-4b587b180b05"
    }
  ],
  "trusted_ca": [
    {
      "id": "fe92910f-98c4-4dd4-bdf0-f39a2160d5b5",
      "uri": "kylo:kylo:connectionmgmt:cm-ca:test2-fe92910f-98c4-4dd4-bdf

```

```

0-f39a2160d5b5",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2023-04-19T10:03:02.533998Z",
    "certificate": "-----BEGIN CERTIFICATE-----\nMIIFrzCCA5egAwIBAgIRAIDg
uiVeBLRGPRQUEerSkYEwDQYJKoZIhvcNAQELBQAw\nWjELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAIRYMQ8
wDQYDVQQHEwZBdXN0aW4xDzAN\nBgNVBAoTB1RoYWxlczEcMBoGA1UEAxMTQ21waGVyVHJlc3QgUm9vdC
BDQTAeFw0y\nmMzA0MTcwMjEyMTFaFw0zMzA0MTUwMjEyMTFaMFOxCzAJBgNVBAYTA1VTMQswCQYD\nIEwJUWDEPMA0GA1UEBxMGQXVzdGluMQ8wDQYDVQQKEwZUaGFsZXNMxHDAaBgNV\nnBAMTE0NpcGhlclRydX
N0IFJvb3QgQ0EwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAw\nnggIKAoICAQCwz0QtPo+cFY2glDdusXMhk
rVgv0bMGf9eXqgLOQ+8PsAoDvLg43AU\nnAx7dDKvd73L1DsQmOwqeslx3bS3deBR2zyCfhy6tPliL2iIL
sPuzJZqTzyu0+Sfs\nnrnwplwM1oZE0E0QkkCM8vx017gn2plhx8CU9cu/s9Dbe1fcBFbqLURkDN3U3pTe
o\nnmACHjTQSVQwN6ZcpIEdAycRaUL9+AKt6/k9DjusQpNjKmtJjY4OYp5j6JGAGFdcK\nnufzjsVzb16EE
Mu6yyXi/P/QXT8+7zwJdaocT80kMRrnIYvmLwjCNo3kZpvXLsE1G\nnHPgNbzdzbQgVT2afLeGKtMI1sRdo
KII8gvEBdeO9NMMyD/+VeY0Yop+hOSqPP/XQbF\ndtVenqKUG/6mq3Z6cz92R0R/Sw9KNnDixFa0cy4KAZ
hS2mn9E8HrTfwzn6mZiWbl\nn3zehiyMN2eyOB1sAfJwL04D6bZYDWD1Nnf/XXLChhkzUFImeA5QkYiPg0
ZgNOGaD\nnkveQbaog26ui4REaJm0cTjOnzIyRwl14iQ0OdDlrBcasgpDFMNMzsYEFmunPfLxK\nn90+rgz
lXXcowl7ZYIYwjcoc1IqPPNhjx2pHlkuYe8XXusRsuBEuHmg3eL+SpfsHE\nnHnJn64ddofDqSGZ/4lJGY
vcCVTE02jAa5olhOC16fcV+y44xc/yZwwIDAQABo3Aw\nnbjAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/
BAUwAwEB/zAdBgNVHQ4EFgQUWdYF\nnoqjVth/meouD25TsPelpfEwLAYDVR0RBCUwI4EhdGVjaG5pY2F
sLnN1cHBvcnRA\nndGhhbGVzZ3JvdXAuY29tMA0GCSqGSIb3DQEBCwUAA4ICAQBz8a2KZo1MnkmD5w4I\nn
H7XpHsQ6cG1zYH72P5jik+6EHEkHHPXITvrj8nAW7nsXAZPLjjaJLJy46r94GYZN\nneIY7PGgoRTdwRsS
TPTmTc6cw/ZgJd5hWukKe0D1C15Lo7k87KV3CVh9aH28R1sEe\nnFxY52b44qa/dZFAxUU5meXYFI/5l rz
MvDZ7gTVQWhR0FFz5b8OcG0HupJQe/xson\nnppERSDYLm4R6eDQ/kqQelFDEa3s8EUob3BQtPYWNMRPow
lVOWir74XtXk9K4cPml\nnz3tYZm4CqPLNtDiRmp3Qob/5EXzkgD7HdVzY7UkcZg5vTGZhIxHPZ4dCDE1x
gEBS\nn2IIP005IEp3LBG9bqps4U1fMN7MjQf2mG1AoLhHA0mvVosAeawRmPQh8Ydvtk6mQ\nn9HxmMk+Wg
OVu06V/eYocXX0D7DYwJXWreFjRcpCtnpe0zOyzEdic025NoCM8yNa1\nn3b889t5aXYFeZEI4bHu7IKYh
0poGQyIPsN775UdmU3QU1XfghLx7s+faeoo0lt18\nncnfbvrwZm++85Awb1caoOjuEeJj2h6xv5eRDpzC
6F57pka80hNlhmCbMmbQFgjrE\nnveC9pkn8Yf79RYC3HzRGgsOnoEdadlmibfF0M018M21+aEubChK1Bt
b1Z6F4Azje\nn8RccyyfNBjhXQa73QAoaKH02A==\nn-----END CERTIFICATE-----",
    "external_cm_id": "4df54fb4-d3e6-432d-a917-4b587b180b05",
    "subject_dn": "CN=CipherTrust Root CA,O=Thales,L=Austin,ST=TX,C=US",
    "expires_at": "2033-04-15T02:12:11Z"
  }
]
}

```

Updating a CM Connection

To update a connection, run:

Syntax

```

ksctl connectionmgmt cm modify --id <Connection-Name/ID> --products <Products-Nam
es> --meta <Key:Values> --client-cert-file <Client-Certificate-File> --client-id
<Client-ID> --external-cm-id <External-CM-ID>

```

Example Request

```

ksctl cm modify --id "fc01d709-7b2a-4235-87fe-6cdd20f1f22a" --client-cert-file
./clientCert.pem --client-id "1cd6f437-53e1-4993-a391-1c503fb67eb0"

```

Example Response


```

{
  "id": "fc01d709-7b2a-4235-87fe-6cdd20f1f22a",
  "external_cm_id": "4df54fb4-d3e6-432d-a917-4b587b180b05",
  "uri": "kylo:kylo:connectionmgmt:connections:connection-1-cm-fc01d709-7b2a-4235-87fe-6cdd20f1f22a",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-04-19T10:20:16.124632Z",
  "updatedAt": "2023-04-19T10:57:31.751032799Z",
  "service": "CM",
  "category": "KeyManager",
  "last_connection_ok": false,
  "last_connection_error": "Connection error for hostname 54.173.227.108 : Post
\"https://54.173.227.108/api/v1/auth/tokens/\": x509: cannot validate certificate
for 54.173.227.108 because it doesn't contain any IP SANs\n",
  "last_connection_at": "2023-04-19T10:26:55.486235Z",
  "name": "connection_1_CM",
  "products": [
    "cckm"
  ],
  "client_certificate": "-----BEGIN CERTIFICATE-----\nMIIFGjCCA2qgAwIBAgIQdS2Uo
iGtJ4+T2hZA/w63ZzANBgkqhkiG9w0BAQsFADBA\nnMQswCQQYDVQQGEwJVUzELMAkGA1UECBMCFVgxDzAN
BgNVBACtBkF1c3RpbjEPMA0G\nnA1UEChMGVGVhbnGVzMRwwGgYDVQQDExNDaXBoZXJUCnVzdCBSb290IEN
BMB4XDTEz\nnMDQxODEwNTQxM1oXDTI1MDQxODEwNTQxM1owgZMxCzAJBgNVBAYTAlVTMQswCQQYD\nnVQQIEwJNRDEQMA4GA1UEBxMHQmVsY2FtcDEVMBMGA1UEChMMVGhbnGVzIEdyb3Vw\nnMQwwCgYDVQQLEwNSbkQ
xETAPBgNVBAMTCGt5bG8uY29tMS0wKwYDVQQQEyQxY2Q2\nnZjQzNy01M2UxLTQ5OTMtYTM5MS0xYzUwM2
ZiNjdldYjAwggEiMA0GCSqGSIb3DQEB\nnAQUAA4IBDwAwggEKAoIBAQQDDH6fIayjJUVT/LVDF3qCJHZ7Mf
hWGvgKG1ZnD4LqL\nnPvHZR5IUvuQO2zUsCKwPi4ilaF4nNlcvDlflCjLEyboct01LsBaQOgmpza7Hmq3
s\nnEKhUtYQBCbjSznunpuDEJYFkDAEbJ2+kitj1mGwltryN28TeBodaZxampN6BpGjN\nnDimKwK7JprA8
Lt69lhGFJmZiJQzV8phuD7wazCF6ey73NWq8PQfYb0oShzg00eT\nnMXx9cbt0AEeZ3Q8ZA+0st8bDleL
ShMdDI+MFgYzoQdelRlrKSMYXMXcjTuk9vBZ8\nn5H+SsU69+ovt0v66Uty+eYF5hkxyrECwprvaB9xKIu
PdAgMBAAGjggEIMIIBDDAO\nnBgNVHQ8BAf8EBAMCA4gwEwYDVR01BAwwCgYIKwYBBQUHAWIwDAYDVR0TA
QH/BAIw\nnADAFBgNVHSMEGDAwGBRZ1gWiqNW2H+Z6i4Pb1Ow97W1+cTBOBgNVHREERzBFghEq\nnLnRoYW
xlc2dyb3VwLmNvbYIRKi50aGFsZXNncm91cC5uZXSBF2NvbnRhY3RAdGhh\nnbGVzZ3JvdXAuY29thwQBA
QEBMF4GA1UdHwRXMFUwU6BRoE+GTWh0dHA6Ly9jaXBo\nnZXJ0cnVzdG1hbmFnZXIubG9jYWwvY3Jscy80
MDYyZWU1Yy1lMzA2LTQ4OTMtYjli\nnMi0zMWY1ZmJjY2JiNjguY3JsmA0GCSqGSIb3DQEB\nnCwUAA4ICAQA
nrnCNPIurCg0\nnTe+Gbt01n73AdPeNcyi5XDUSySmTBo5RqbGYJxJWNK3MnUxH4CGP+HCQx3ngJmXg\nn
2rq+WjcxhToSQ3U5oGHUksXungAjWIPjeohbOQ1CRX7SuoXZGaYtoLODiocGbzoA\nnTW2r0HewbFw5HED
jRTdgObDeO2tkxzb123rHDYB9Q2NbZvw04+cqtnWk0YmdnXf\nnMt+FeK5bej3m0ysaByvm0rwCoF3F4T
HX8BDca7CSdYjDRfVrg20mK8gl9XGQjMCh\nnTLC/Alna+JyNV/f7v1Uyzyld5jtaGYwoYSZRnTZlow8XJ
Uj1CpUDTXi5BnP+0zqW\nneGzm0DRHNS8xzQig0avIG8wMmlH0FXCIdoZHUX+mxSXG0RjLZ86VqM4joQ/A
Qc31\nnbNTfSVA0krkbj6maxOonv8yWu7nMAF917L7VFfvcYKABiZrwrP5V8WxB7w7rvMxs\nnMa3GKYvq7
tUTyE0qN+un6eRTctSPNdbia8pl07z6IGC14plXCMQeZa+7B1QubAWZ\nnhWywZQdQ6MSjh3CH8BeMQS1J
5qEO4iK+le2kWhpfYRvzVQLzfT6P3bpm9zeOFTCv\nnYQDfL2RCDtXAfGxBXp4ty+b8Ju2AAz38Cs6mkM
pvBm6APgJ1V2Y003pUm1ECgqV\nnIFmsJviCW1F9OYsnC1w8K0cfbk3A8w==\nn-----END CERTIFICAT
E-----\nn",
  "client_id": "1cd6f437-53e1-4993-a391-1c503fb67eb0",
  "nodes": [
    {
      "id": "efec26ad-f9f1-44d0-850f-be2cc3ed54d0",
      "uri": "kylo:kylo:connectionmgmt:cm-node:test2-efec26ad-f9f1-44d0-850f-be2cc3ed54d0",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-04-19T10:03:02.532367Z",
      "hostname": "54.173.227.108",
      "external_cm_id": "4df54fb4-d3e6-432d-a917-4b587b180b05"
    }
  ],
  "trusted_ca": [

```

To delete a connection, run:

```
ksctl connectionmgmt cm delete --id <Connection-Name/ID>
```

```
ksctl connectionmgmt cm delete --id "fc01d709-7b2a-4235-87fe-6cdd20f1f22a"
```

There will be no response if CipherTrust Manager connection is deleted successfully.

Getting List of CM Connections

To list all the connections, run:

Syntax

```
ksctl connectionmgmt cm list
```

Example Request

```
ksctl connectionmgmt cm list
```

Example Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "resources": [
    {
      "id": "fc01d709-7b2a-4235-87fe-6cdd20f1f22a",
      "external_cm_id": "4df54fb4-d3e6-432d-a917-4b587b180b05",
      "uri": "kylo:kylo:connectionmgmt:connections:connection-1-cm-fc01d709-7b2a-4235-87fe-6cdd20f1f22a",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-04-19T10:20:16.124632296Z",
      "updatedAt": "2023-04-19T10:20:16.122916547Z",
      "service": "CM",
      "category": "KeyManager",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "connection_1_CM",
      "products": [
        "cckm"
      ],
      "client_certificate": "-----BEGIN CERTIFICATE-----\nMIIFGzCCA2ugAwIBAgTAlRYMQ8wDQYDVQQHEwZBdXN0aW4xDzAN\nBgNVBAoTB1RoYWxlczEcMBoGA1UEAxMTQ21waGVyVHJlc3QgUm9vdCBDQTAeFw0y\nnMZA0MTgxMDE0NDJaFw0yNTA0MTgxMDE0NDJAMIGTMQswCQYDVQQGEwJVUzELMAkG\nA1UECBMCTUQxEDAOBgNVBAcTB0JlbGhXbXAxFTATBgNVBAoTDFRoYWxlcyBHcm91\nncDEMMAoGA1UECjMDUm5EMREwDwYDVQQDEWhreWxvLmNvbTEtMCsGA1UEIhMkMDA4\nnYzUyYmUtNGQyNi00MGlyLWE0MWYtYTg2ZGE1MzQxMDdlMIIBIjANBgkqhkiG9w0B\n\nnAQEFAAOCAQ8AMIIBCgKCAQEAur6M7t4Z7cGTMQP0qJIK4LgsA0Erf1RujWwVDE1J\n\nnAV3B1BL3mlnPrZm1Mu6U5sTU5eTLlT+HE+nRxWYPJ4Wed5GtZ/cBrWh4Z7IcP5W1\n\nnq9qXioiYL7hKnfGRHGJ4+WpueezR6HYoZSX/G5svEYd+sDi/4073F1vx8pIkd0wR\n\nnH4a74EQ07y2SsDKGwnndQFJCQzJlizuKBBQeNFi+pTU70AdDiEaZkt6/sIxFeVS\n\nnNpPhjIcq/HW91zGYxGtIfcznsTQAyOjKn6wc/hC3c33IcNRRQPg+MPPyJzJswyGdju\n\nnlnLZNNQxRyrDV2HS+DO0PjS+bTxE8v68ejeHbuLvskOUQqWIDAQABo4IBCDCCAQQw\n\nnDgYDVDR0PAQH/BAQDAgoIMBMGA1UdJQQMMAoGCCsGAQUFBwMCMAwGA1UdEwEB/wQC\n\nnMAAwHwYDVR0jBBgwFoAUWdYFogjVth/meouD25TsPelpfEwTgYDVR0RBECwRYIR\n\n\nnKi50aGFsZXNncm91cC5jb22CESoudGhhbGVzZ3JvdXAubmV0gRdjb250YWN0QHRO\n\n\nnYWxlczdyb3VwLmNvbYcEAQEBAQEBBgNVHR8EVzBVMFOgUaBPhk1odHRwOi8vY2l1w\n\n\nnaGVydHJlc3RtYW5hZ2VzLmNvbY2FsL2NybmhmVnNDA2MmVlNWMTZTMwNi00ODkzLWI5\n\n\nnYjItMzMzFmNWZiY2NiYjY4LmNybDANBgkqhkiG9w0BAQsFAAOCAgEAVFXX69fNsMoB\n\n\nnCLEflrfAB2uHjRvdjSlnkWrNT5fWzD37ig85w70aYLH3yfdglHP4XilKEq
```

```

MoX2E\nkC2NFSsA9JpwpYV3ZkZk9Q6oJBxUwg9ploxF3KJ/t6TkHR53ZY4qhak+fVXfjrBq\nnHAoXusC4
inIGYPc6x9OoWAqJjZBJTHICoXouX4VD9i1hTrOGrMCTsRkG+WFBU3Zp\n2DcwufZoSTNV/I8xHhqfSHi
aTaeOJozG2Aoq3P06AT0JaphbJ1tJBTnua4ceA+/h\nJjwsC1Xuc6fHu+OQL2MxOC8DrCeP2tQc95KwRv
b7WJJCeVu6To+Zto0m/Jc6rrne\nnmu5D9Tnv/cmDfjPQk1EVf/IsL+baHpEDf9399ZI6w8WvU45IanYYB
SMtRaHnKesa\n8k1bjWHZnWvG4ZTbOawX0q5LMkB+1Ee/8YxDvy3RlCRcGQGH+sPwgj0L/XzdAvK6\ne9
mjs2nzpZF37Ns4UiaHUo44XZJDikes8cqUpJhiqjPo/YPPeUqMGjucpCeIqKlh\nnZoio6d8h3WddvsII
v+ygkAnUeVtiq0M2lwUKtKEBo+i8iXL3rctGCyPF5hQkRXWS\nnu5hiV19frMKtUE4X7FQIEPrZ1cPwsnf
cQsMpZf5Mwbbwm/Ych9rrqWanYNAq+PVg\nn7U4C7GwJZYDD3349pO7g4CasiYSCorI=\n-----END CER
TIFICATE-----\n",
  "client_id": "008c52be-4d26-40b2-a41f-a86da534107e",
  "nodes": [
    {
      "id": "efec26ad-f9f1-44d0-850f-be2cc3ed54d0",
      "uri": "kylo:kylo:connectionmgmt:cm-node:test2-efec26ad-f9f
1-44d0-850f-be2cc3ed54d0",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-04-19T10:03:02.532367Z",
      "hostname": "54.173.227.108",
      "external_cm_id": "4df54fb4-d3e6-432d-a917-4b587b180b05"
    }
  ],
  "trusted_ca": [
    {
      "id": "fe92910f-98c4-4dd4-bdf0-f39a2160d5b5",
      "uri": "kylo:kylo:connectionmgmt:cm-ca:test2-fe92910f-98c4-4d
d4-bdf0-f39a2160d5b5",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-04-19T10:03:02.533998Z",
      "certificate": "-----BEGIN CERTIFICATE-----\nMIIFrzCCA5egAwIB
AgIRAIDguiVeBLRGPRQUEerSkYEwDQYJKoZIhvcNAQELBQAwnWjELMAkGA1UEBhMCVVMxZzA5BGNVBAg
TAlRYMQ8wDQYDVQQHEwZBdXN0aW4xDzAN\NBGNVBAoTB1RoYXxlczEcMBoGA1UEAxMTQ2lwaGVyVHJlc3
QgUm9vdCBDQTAeFw0y\nmZaA0MTcwMjEyMTFaFw0zMzA0MTUwMjEyMTFaMFoxZzA5BGNVBAYTA1VTMQswC
QYD\nnVQqIEwJUWDEPMA0GA1UEBxMGQXVzdGluMQ8wDQYDVQQKEwZUaGFsZXN0aW45BGNVBAoTB1R0YX
hlclRydXN0IFJvbn3QgQ0EwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAw\nnggIKAoICAQCwz0QtPo+cFY2gl
DdusXMhkrVgv0bMGf9eXqgL0Q+8PsAoDvLg43AU\nnAx7dKvd73L1DsQmOwgeslx3bS3deBR2zyCfhy6t
Plil2iILsPuzJZqTzyu0+Sfs\nnrnwplwM1oZE0E0QkckM8vx017gn2plhx8CU9cu/s9Dbe1fcBFbqLURk
DN3U3pTeo\nnmAchjTQSVQwN6ZcpIEdAycRaUL9+AKt6/k9DjusQpNjKmtJjY4OYp5j6JGAgFdcK\nnufzj
sVzb16EEMu6yyXi/P/QXT8+7zwJdaocT80kMRrniYvmLwjCNO3kZpvXLsE1G\nnHPgNbzdBQgVT2afLeGK
tMI1sRdoKII8gvEBdeO9NMMyD/+VeY0Yop+hOSqPP/XQbF\ndtVenqKUG/6mq3Z6cz92R0R/Sw9KNnDixF
a0cy4KAZhS2mn9E8HrTfwzn6mZiWbl\n3zehiyMN2eyOB1sAfJwL04D6bZYDWD1Nnf/XXLChhkzUFImeA
5QkYiPg0ZgNOGaD\nnkveQbaog26ui4REaJm0cTjOnzIyRwl14iQ0OdDlRbcasgPDFMNMZsYEFmunPflX
K\n90+rgzlXXcowl7ZYIYwjcoc1IqPPNhjx2pHlkuYe8XXusRsuBEuHmg3eL+SpsfHE\nnHnJn64ddofDq
SGZ/41JGYvcCVTE02jAa5olhOC16fcV+y44xc/yZwwIDAQAB03Aw\nnbjA0BgNVHQ8BAf8EBAMCAQYwDwY
DVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUWdYF\nnoqjVth/meouD25TsPe1pfnEwLAYDVR0RBCUwI4EhdG
VjaG5pY2FsLnN1cHBvcnRA\nndGhhbGVzZ3JvdXAuY29tMA0GCSqGSIb3DQEBChwUAA4ICAQBz8a2KZo1Mn
kmD5w4I\nnH7XpHsQ6cGlzYH72P5jik+6EHEkHHPXITvRj8nAW7nsXAZPLjjaJLJy46r94GYZN\neIY7PG
goRTdwRsSTPTmTc6cw/ZgJd5hWukKe0D1C15Lo7k87KV3CVh9aH28R1sEe\nnFxY52b44qa/dZFAxUU5me
XYFI/5lRzmvDZ7gTVQWhR0FFz5b8OcG0HupJQe/xson\nnppERSDYLM4R6eDQ/kqQelFDEa3s8EUob3BQt
PYWNMRPowlVOWir74XtXk9K4cPml\nnz3tYZm4CqPLNtDiRmp3Qob/5EXzkgD7HdVzY7UkcZg5vTGZhIxH
PZ4dCDE1xgEBS\n2IIpO05IEp3LBG9bqps4U1fMN7MjQf2mGLA0LhHA0mvVosAeawRmPQh8Ydvtk6mQ\nn
9HxmMk+WgOVu06V/eYocXX0D7DYwJXWreFjRcpCtnpe0zOyzEdic025NoCM8yNa1\n3b889t5aXYFeZEI
4bHu7IKYh0poGQYIPsN775UdmU3QU1XfghLx7s+faeoo01t18\nncnfbrvwZm++85Awb1caoOjuEeJj2h6
xv5eRDpzC6F57pka80hNlmhCbMmbQFgjre\nnveC9pkn8Yf79RYC3HzRGgsOnoEdadlmibfF0M018M21+a
EubChK1Btb1Z6F4Azje\nn8RccyyfNBjhxQA73QAoaKH02A==\n-----END CERTIFICATE-----",
      "external_cm_id": "4df54fb4-d3e6-432d-a917-4b587b180b05",
      "subject_dn": "CN=CipherTrust Root CA,O=Thales,L=Austin,ST=T
X,C=US",
      "expires_at": "2033-04-15T02:12:11Z"
    }
  ]
}

```

```

    ]
  }
]
}

```

Testing an Existing CM Connection or Connection Parameters

To test an existing connection or connection parameters, run:

Note

For a successful TLS connection, the external CipherTrust Manager web server certificate SAN field should not be empty. For example, if the hostname connection is IP, the same IP should be present in the SAN field in the web server certificate of external CipherTrust Manager.

Syntax

```

ksctl connectionmgmt cm test --id <Connection-Name/ID>

*Or*

ksctl connectionmgmt cm test --client-cert-file <client-certificate-file> --client-id <client-id> --external-cm-id <external-cm-id>

```

Example Request

```

ksctl connectionmgmt cm test --id "14ff5e03-5fb5-4f74-bda5-99f70ba14da8"

*Or*

ksctl connectionmgmt cm test --client-cert-file ./clientCert.pem --client-id "fc797b46-5470-474c-8797-fdd25eb011a7" --external-cm-id "f8946eb8-bc0f-439a-958e-0238b30f9272"

```

Example Response

```

{
  "status": true,
  "nodes":
  [
    {
      "hostname": "3.87.253.205",
      "connection_ok": true
    }
  ]
}

```

Akeyless Gateway

The Akeyless Gateway connection is required for [Secrets Management](#).

You can create and manage Akeyless Gateway Connections through the CipherTrust Manager GUI or CLI.

Creating Akeyless Gateway Connections using GUI

Note

An Akeyless Gateway Connection is automatically created as part of the [initial configuration](#). We strongly recommend that you configure CipherTrust Manager with this autogenerated connection. Only manually create Akeyless Gateway connections for recovery or testing purposes.

To manually create an Akeyless Gateway connection, you require the **Access Key ID** and **Access Key** of the Gateway-Admin authentication method. This is obtained through [resetting the Gateway-Admin credentials](#), which invalidates the initial autogenerated Akeyless connection.

Click the **Test Credentials** button to check whether CipherTrust Manager can connect to Akeyless Vault Platform using the **Gateway-Admin** credentials.

Note

This connection is not associated with a product, so there is no **Add Products** selection.

Managing Akeyless Gateway Connections using ksctl

You can perform the following operations on an Akeyless Gateway connection:

- View one Akeyless Gateway connection
- Update
- Delete
- List all Akeyless Gateway connections
- Test an existing Akeyless Gateway Connection
- Create
- Test parameters for a new Akeyless Gateway connection

Viewing Details for an Akeyless Gateway connection

To view details for an Akeyless Gateway connection run:

Syntax

```
ksctl connectionmgmt akeyless get --id <connection_name_or_id>
```

Example Request

```
ksctl connectionmgmt akeyless get --id my-akeyless-connection
```

Example Response

```
{
  "id": "da862727-157a-40f7-8847-061cd1e82839",
  "uri": "kylo:kylo:connectionmgmt:connections:my-akeyless-connection-da862727-157a-40f7-8847-061cd1e82839",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-08-02T15:51:48.354687Z",
  "updatedAt": "2023-08-02T15:51:48.353866Z",
  "service": "akeyless",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
  "name": "my-akeyless-connection",
  "access_key_id": "p-123456abcdef"
}
```

Updating an Akeyless Gateway connection

You can update the Gateway Admin access key ID, the Gateway Admin access key, or the meta information associated with an Akeyless Gateway connection. The example updates the meta information.

To update an Akeyless Gateway connection run:

Syntax

```
ksctl connectionmgmt akeyless modify --id <connection_name_or_id> [--akeyless-key-id <gateway_admin_key_id>] [--akeyless-key <gateway_admin_key>] [--meta <comma_separated_key:value>]
```

Example Request

```
ksctl connectionmgmt akeyless modify --id my-akeyless-connection --meta "{\"color\":\"blue\"}"
```

Example Response

```
{
```

```

    "id": "da862727-157a-40f7-8847-061cd1e82839",
    "uri": "kylo:kylo:connectionmgmt:connections:my-akeyless-connection-da862727-157a-40f7-8847-061cd1e82839",
    "account": "kylo:kylo:admin:accounts:kylo",
    "createdAt": "2023-08-02T15:51:48.354687Z",
    "updatedAt": "2023-08-02T16:05:46.798878173Z",
    "service": "akeyless",
    "category": "cloud",
    "last_connection_ok": null,
    "last_connection_at": "0001-01-01T00:00:00Z",
    "name": "my-akeyless-connection",
    "meta": {
      "color": "blue"
    },
    "access_key_id": "p-123456abcdef"
  }

```

Deleting an Akeyless Gateway connection

To delete an Akeyless Gateway connection run:

Syntax

```
ksctl connectionmgmt akeyless delete --id <connection_name_or_id>
```

Example Request

```
ksctl connectionmgmt akeyless delete --id my-akeyless-connection
```

There will be no response if the connection is deleted successfully.

Listing all Akeyless Gateway connections

To list all Akeyless Gateway connections run:

Syntax

```
ksctl connectionmgmt akeyless list
```

Example Request

```
ksctl connectionmgmt akeyless list
```

Example Response

```

{
  "skip": 0,
  "limit": 10,
  "total": 3,
  "resources": [

```



```

    {
      "id": "da862727-157a-40f7-8847-061cd1e82839",
      "uri": "kylo:kylo:connectionmgmt:connections:my-akeyless-connection-da862727-157a-40f7-8847-061cd1e82839",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-08-02T15:51:48.354687Z",
      "updatedAt": "2023-08-02T16:05:46.798878Z",
      "service": "akeyless",
      "category": "cloud",
      "last_connection_ok": null,
      "last_connection_at": "0001-01-01T00:00:00Z",
      "name": "my-akeyless-connection",
      "meta": {
        "color": "blue"
      },
      "access_key_id": "p-123456abcdef"
    },
    {
      "id": "cff708c4-34df-4cc5-8f4a-06b22bell1c87",
      "uri": "kylo:kylo:connectionmgmt:connections:akeyless-connection-2-cff708c4-34df-4cc5-8f4a-06b22bell1c87",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-08-01T15:50:18.247739Z",
      "updatedAt": "2023-08-01T15:50:22.890931Z",
      "service": "akeyless",
      "category": "cloud",
      "last_connection_ok": true,
      "last_connection_at": "2023-08-01T15:50:22.890733Z",
      "name": "akeyless-connection-2",
      "access_key_id": "p-123456abcdef"
    },
    {
      "id": "b625f1c0-542c-4f3c-b467-78c7b80becb0",
      "uri": "kylo:kylo:connectionmgmt:connections:akeyless-b625f1c0-542c-4f3c-b467-78c7b80becb0",
      "account": "kylo:kylo:admin:accounts:kylo",
      "createdAt": "2023-08-01T15:22:01.218978Z",
      "updatedAt": "2023-08-01T15:22:05.004709Z",
      "service": "akeyless",
      "category": "cloud",
      "last_connection_ok": true,
      "last_connection_at": "2023-08-01T15:22:05.004562Z",
      "name": "akeyless",
      "access_key_id": "p-123456abcdef"
    }
  ]
}

```

Testing an Existing Akeyless Gateway connection

To test an existing Akeyless Gateway connection run:

Syntax

```
ksctl connectionmgmt akeyless test --id <connection_name_or_id>
```

Example Request

```
ksctl connectionmgmt akeyless test --id my-akeyless-connection
```

Example Response

```
{
  "connection_ok": true
}
```

Creating an Akeyless Gateway Connection

Note

An Akeyless Gateway Connection is automatically created as part of the [initial configuration](#). We strongly recommend that you configure CipherTrust Manager with this autogenerated connection. Only manually create Akeyless Gateway connections for recovery or testing purposes.

To manually create an Akeyless Gateway connection, you require the **Access Key ID** and **Access Key** of the Gateway-Admin authentication method. This is obtained through [resetting the Gateway-Admin credentials](#), which invalidates the initial autogenerated Akeyless connection.

To create an Akeyless Gateway connection run:

Syntax

```
ksctl connectionmgmt akeyless create --name <connection_name> --akeyless-key-id
<gateway_admin_key_id> --akeyless-key <gateway_admin_key> [--meta <comma_separate
d_key:value>]
```

Example Request

```
ksctl connectionmgmt akeyless create --name my-akeyless-connection --akeyless-ke
y-id p-123456abcdef --akeyless-key mG6yXCUmGKCCGDKnQecDxmVdWjCzLGbRQJbnztQ7Tn9e
```

Example Response

```
{
  "id": "da862727-157a-40f7-8847-061cd1e82839",
  "uri": "kylo:kylo:connectionmgmt:connections:my-akeyless-connection-da86272
7-157a-40f7-8847-061cd1e82839",
  "account": "kylo:kylo:admin:accounts:kylo",
  "createdAt": "2023-08-02T15:51:48.354687497Z",
  "updatedAt": "2023-08-02T15:51:48.353865994Z",
  "service": "akeyless",
  "category": "cloud",
  "last_connection_ok": null,
  "last_connection_at": "0001-01-01T00:00:00Z",
}
```

```
{
  "name": "my-akeyless-connection",
  "access_key_id": "p-123456abcdef"
}
```

Testing Parameters for a new Akeyless Gateway connection

To test the Gateway Admin credentials before creating an Akeyless Gateway connection run:

Syntax

```
ksctl connectionmgmt akeyless test --akeyless-key-id <gateway_admin_access_key_id> --akeyless-key <gateway_admin_access_key>
```

Example Request

```
ksctl connectionmgmt akeyless test --akeyless-key-id p-123456abcdef --akeyless-key mG6yXCUmGKCCGDKnQecDxmVdWjCzLGbRQJbnztQ7Tn9e
```

Example Response

```
{
  "connection_ok": true
}
```

Browsing LDAP Users and Groups

The CipherTrust Manager allows you to browse the LDAP users and groups created in the LDAP connection manager. Before browsing, make sure that LDAP connection must be configured. Refer to [LDAP](#) for details.

Note

LDAP browsing is not applicable to the [LDAP](#) connection available through Access Management.

Browsing LDAP Users

To browse LDAP users, run:

Syntax

```
ksctl ldap-browse users --connection-id <connectionID/name> --limit <limit> --use-username <username> --groupname <groupname> --server-maximum-page-size <server-maximum-page-size> --early-listing-stop <early-listing-stop>
```

Here,

- `connection-id` - connection ID of the ldap server.
- `limit` - maximum number of connection information structures that can be returned by this query.
- `username` - username of the ldap user.
- `groupname` - groupname of the ldap group.
- `server-maximum-page-size` - Sets the page size which determines the number of users/groups to be fetched from the LDAP server using pagination.
- `early-listing-stop` - When set to `false` displays the total number of users/groups that are present on the LDAP server. When set to `true`, the value of `total` in the response is displayed as `-1` indicating that more records are present on the ldap server than the limit specified. The default value of `early-listing-stop` is `false`. Let's take a scenario where we have set `early-listing-stop` to `true` and the LDAP server contains total 5000 users. On the CipherTrust Manager, the `limit` parameter is set to 1000, then total number of entries in response will 1000 and the value of `total` will be `-1`.

Example Request 1

```
ksctl ldap-browse users --connection-id 19f0228c-a928-4ba0-87d2-bcef829f77eb --limit 2
```

Example Response 1

```
{
  "skip": 0,
  "limit": 2,
  "total": 7,
  "users": [
    {
      "DN": "cn=Amy Wong+sn=Kroker,ou=people,dc=planetexpress,dc=com",
      "attributes": [
        {
          "name": "cn",
          "values": [
            "Amy Wong"
          ]
        },
        {
          "name": "uid",
          "values": [
            "amy"
          ]
        }
      ]
    },
    {
      "DN": "cn=Bender Bending Rodríguez,ou=people,dc=planetexpress,dc=com",
      "attributes": [
        {
          "name": "cn",
```

```

        "values": [
            "Bender Bending Rodríguez"
        ]
    },
    {
        "name": "uid",
        "values": [
            "bender"
        ]
    }
]
}

```

Example Request 2

```

ksctl ldap-browse users --connection-id 5ca9eaae-f319-4d62-b448-21c1dd537cc2 --li
mit 1 --username fry

```

Example Response 2

```

{
    "skip": 0,
    "limit": 1,
    "total": 1,
    "users": [
        {
            "DN": "cn=Philip J. Fry,ou=people,dc=planetexpress,dc=com",
            "attributes": [
                {
                    "name": "cn",
                    "values": [
                        "Philip J. Fry"
                    ]
                },
                {
                    "name": "uid",
                    "values": [
                        "fry"
                    ]
                }
            ]
        }
    ]
}

```

Example Request 3

```

ksctl ldap-browse users --connection-id 5ca9eaae-f319-4d62-b448-21c1dd537cc2 --li
mit 1 --username bender --groupname *crew

```

Example Response 3

```
{
  "skip": 0,
  "limit": 10,
  "total": 1,
  "users": [
    {
      "DN": "cn=Bender Bending Rodríguez,ou=people,dc=planetexpress,dc=com",
      "attributes": [
        {
          "name": "cn",
          "values": [
            "Bender Bending Rodríguez"
          ]
        },
        {
          "name": "uid",
          "values": [
            "bender"
          ]
        }
      ]
    }
  ]
}
```

Example Request 4

```
ksctl ldap-browse users --connection-id 19f0228c-a928-4ba0-87d2-bcef829f77eb --limit 2 --server-maximum-page-size 10000 --early-listing-stop true
```

Example Response 4

```
{
  "skip": 0,
  "limit": 2,
  "total": -1,
  "users": [
    {
      "DN": "uid=arsh,ou=people,dc=example,dc=com",
      "attributes": [
        {
          "name": "uid",
          "values": [
            "2000"
          ]
        }
      ],
      "DN": "cn=Member4,ou=people,dc=example,dc=com",
      "attributes": [
        {
          "name": "uname",
          "values": [
            "Member4"
          ]
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

Browsing LDAP Groups

To browse LDAP groups, run:

Syntax

```

ksctl ldap-browse groups --connection-id <connectionID/name> --limit <> --group_id_attribute <group-id-attribute> --groupname <group-name> --user_id <user-id> --server-maximum-page-size <server-maximum-page-size> --early-listing-stop <early-listing-stop>

```

Here,

- `connection-id` - connection ID of the ldap server.
- `limit` - maximum number of connection information structures that can be returned by this query.
- `group_id_attribute` - attribute inside the group object which contains the group identifier (name). This value should be unique.
- `groupname` - groupname of the ldap group.
- `user_id` - username of the ldap user.
- `server-maximum-page-size` - sets the page size which determines the number of users/groups to be fetched from the LDAP server using pagination.
- `early-listing-stop` - when set to `false` displays the total number of users/groups that are present on the LDAP server. When set to `true`, the value of `total` in the response is displayed as `-1` indicating that more records are present on the ldap server than the limit specified. The default value of `early-listing-stop` is `false`. Let's take a scenario where we have set `early-listing-stop` to `true` and the LDAP server contains total 5000 users. On the CipherTrust Manager, the `limit` parameter is set to 1000, then total number of entries in response will 1000 and the value of `total` will be `-1`.

Example Request 1

```

ksctl ldap-browse groups --connection-id 5ca9eaae-f319-4d62-b448-21c1dd537cc2 --limit 1

```

Example Response 1

```

{
  "skip": 0,
  "limit": 1,

```

```

    "total": 2,
    "groups": [
      {
        "DN": "cn=admin_staff,ou=people,dc=planetexpress,dc=com",
        "attributes": [
          {
            "name": "cn",
            "values": [
              "admin_staff"
            ]
          }
        ]
      }
    ]
  }
}

```

Example Request 2

```

ksctl ldap-browse groups --connection-id 5ca9eaae-f319-4d62-b448-21c1dd537cc2 --l
imit 1 --group_id_attribute cn --groupname admin*

```

Example Response 2

```

{
  "skip": 0,
  "limit": 1,
  "total": 1,
  "groups": [
    {
      "DN": "cn=admin_staff,ou=people,dc=planetexpress,dc=com",
      "attributes": [
        {
          "name": "cn",
          "values": [
            "admin_staff"
          ]
        }
      ]
    }
  ]
}

```

Example Request 3

```

ksctl ldap-browse groups --connection-id 5ca9eaae-f319-4d62-b448-21c1dd537cc2 --l
imit 1 --group_id_attribute cn --user_id fry

```

Example Response 3

```

{
  "skip": 0,
  "limit": 1,
  "total": 2,

```



```

    "groups": [
      {
        "DN": "cn=admin_staff,ou=people,dc=planetexpress,dc=com",
        "attributes": [
          {
            "name": "cn",
            "values": [
              "admin_staff"
            ]
          }
        ]
      }
    ]
  }
}

```

Example Request 4

```

ksctl ldap-browse groups --connection-id 19f0228c-a928-4ba0-87d2-bcef829f77eb --limit 2 --server-maximum-page-size 10000 --early-listing-stop true

```

Example Response 4

```

{
  "skip": 0,
  "limit": 2,
  "total": -1,
  "groups": [
    {
      "DN": "cn=boys,ou=groups,dc=example,dc=com",
      "attributes": [
        {
          "name": "gname",
          "values": [
            "boys"
          ]
        }
      ]
    },
    {
      "DN": "cn=testGrp8,ou=groups,dc=example,dc=com",
      "attributes": [
        {
          "name": "gname",
          "values": [
            "testGrp8"
          ]
        }
      ]
    }
  ]
}

```

System Upgrade/Downgrade

You can upgrade/downgrade your CipherTrust Manager firmware to and from 2.22 by securely downloading and applying a new/older system archive file.

Note

Consult the [Release Model](#) and [Upgrade Paths](#) pages for high-level overviews of upgrade support across more versions of CipherTrust Manager.

Specific upgrade steps vary depending on whether the CipherTrust Manager is [clustered](#) or [unclustered](#).

[Downgrade](#) is only supported on unclustered appliances.

Upgrading a k160 Thales TCT Device

The k160 Thales Standard model TCT device has a different supported upgrade path than all other CipherTrust Manager models. For this device, the supported upgrade path are:

- 2.11.2-tct to 2.15.0 to 2.16.x to 2.20.x to 2.21.x to 2.22.x
- 2.11.2-tct to 2.15.0 to 2.16.x to 2.20.x to 2.22.x
- 2.11.2-tct to 2.15.0 to 2.20.x to 2.22.x

The High Assurance (HA) model is not supported with this release.

The supported upgrade methods are [unclustered/standalone](#) and [cluster remove/rebuild](#).

System Upgrade for an Unclustered Appliance

Caution

Please read this section carefully before performing a system upgrade.

Refer to [Cluster Upgrade](#) for details on upgrading a CipherTrust Manager which is part of a cluster of devices.

For 2.22.0, we tested upgrade from 2.21.1, 2.20.1, and 2.19.1 for the CipherTrust Manager k570 and k470 physical appliances, and Virtual CipherTrust Manager k170v and k470v instances. For the CipherTrust Manager k160 small form factor appliance, consult the [Upgrade Paths](#) page for supported paths.

Note

Upgrades from other versions have not been tested and may not work correctly.

You require `ksadmin` [level access](#) with an SSH key.

1. Obtain the signed archive file for the upgrade from the Support Portal. The file has the format `ks_upgrade_<major.minor.patch+build_number>.tar.gz.gpg`.
2. On CipherTrust Manager create and download a backup with corresponding backup key, in case there are any problems.

Note

Upgrades keep all the data and may migrate the data and configuration. Therefore, as a precaution, it is recommended to take a [backup](#) before upgrading.

Consult the [backups](#) page for details on forward compatibility for backups. Restoring a newer backup to an older version is never supported.

Caution

If local database audit logging is enabled by restoring a backup, system operation performance can degrade. We recommend [disabling local database audit logging](#) before taking a backup.

3. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

Source Operating System	Example Command Syntax	Required Private Key Format
Linux	<pre>scp -i <path_to_private_SSH_key> <archive_file_name> ksadmin@<ip>:.</pre>	OpenSSH format
Windows	<pre>pscp -i <path_to_private_SSH_key> <archive_file_name></pre>	PuTTY PPK format

Source Operating System	Example Command Syntax	Required Private Key Format
	<code>ksadmin@<ip>:.</code>	

4. `ssh` into the CipherTrust Manager as `ksadmin` and ensure there is sufficient disk space available. The required disk space is 5 times the size of the upgrade file. For example, if the upgrade file is 5 GB, 25 GB of free space is required. Use `df -h /` to view available space.

5. Run the following command:

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

Here, `<archive_file_path>` specifies the CipherTrust Manager path to the signed archive file.

The signature of the archive file is verified and the upgrade is applied.

6. Reboot the appliance when prompted. The following output indicates the upgrade has completed:

```
Upgrade succeeded
Reboot the instance to finish the upgrade
```

Note

If you have lost the SSH session, you can access upgrade logs in `/home/ksadmin/`. Upgrade log files have the prefix `ks_upgrade`. Search for "Upgrade succeeded" to confirm the upgrade completed, and that you must reboot. Take care to verify the upgrade has completed before rebooting. You must always reboot to fully apply the upgrade.

7. Ensure the CipherTrust Manager services have started. From the `ksadmin` session, run `systemctl status keysecure`. Alternatively, you can visit the CipherTrust Manager web console or attempt to connect with the `ksctl` CLI.
8. If you need to verify the new version, look for the version displayed in CipherTrust Manager banner in the SSH session on reboot, as shown in the screenshot. Alternatively, log in the web console UI and click the information icon at the top right to display version information.

[illegible]

9. **Re-download and re-install ksctl** to access commands and settings introduced in the new CipherTrust Manager software version.

Note

Previous versions of ksctl still work with new CipherTrust Manager software for existing features and settings.

System Downgrade

An unclustered CipherTrust Manager can be downgraded to the previous minor version. For release-specific upgrade/downgrade information, refer to the release notes for your release.

Downgrades perform a CipherTrust Manager reset, which wipes all CipherTrust Manager data except the backup files that already exist.

As well, the PCI HSM drivers on k570 models, and base operating system packages are not changed during downgrade.

Warning

As we cannot guarantee stability, we strongly recommend using downgraded systems for test environments only. Do not use a downgraded CipherTrust Manager in a production environment.

To return to a production environment to a previous version:

1. Take a backup.
2. Perform a [system factory reset](#).
3. Upgrade the CipherTrust Manager to the desired version.
4. Restore the backup.

To downgrade your CipherTrust Manager

1. SSH into the CipherTrust Manager as "ksadmin".
2. Downgrade the CipherTrust Manager:

```
$ sudo /opt/keysecure/ks_downgrade.sh -f <~/filename>
```

Usage: `ks_downgrade.sh -f <FILE> [-o]`

```
*  -f: Path to the signed CipherTrust Manager installer file.
*  -o: Clustered node cannot be downgraded. Use this flag to override this behavior. -->
```

Cluster Upgrade

You can only upgrade a cluster to 2.22 using the [Cluster Remove/Rebuild Method](#), due to better lifecycle management for system critical components.

Prerequisites

- You require `ksadmin` [level access](#) with an SSH key.
- Obtain the signed archive file for the upgrade from the Support Portal. The file has the format `ks_upgrade_<major.minor.patch+build_number>.tar.gz.gpg`.

Upgrade Using the Cluster Remove/Rebuild Method

Supported upgrade paths are the same as for [system upgrade of an unclustered appliance](#).

Prerequisite for Cluster Remove/Rebuild Method

Plan for downtime when clients cannot make requests to the cluster and manually block client access to the CipherTrust Manager IP addresses/hostnames in your network before starting.

Caution

If any client request comes into a CipherTrust Manager while the cluster is broken or during the cluster rebuild, the resulting data changes can be lost.

To Perform an Upgrade Using Cluster Remove/Rebuild Method

1. On one of the cluster nodes, create and download a backup with corresponding backup key, in case there are any problems.
2. Remove all nodes from the cluster except one.
3. Delete the cluster configuration on the remaining node.

```
$ ksctl cluster delete
```

4. Perform the upgrade on that remaining node.
 - a. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

Source Operating System	Example Command Syntax	Required Private Key Format
Linux	<pre>scp -i <path_to_private_SSH_key> <archive_file_name> ksadmin@<ip>:.</pre>	OpenSSH format
Windows	<pre>pscp -i <path_to_private_SSH_key> <archive_file_name> ksadmin@<ip>:.</pre>	PuTTY PPK format

- b. SSH into the CipherTrust Manager as `ksadmin` and ensure there sufficient disk space available. The required disk space is 5 times the size of the upgrade file. For example, if the upgrade file is 5 GB, 25 GB of free space is required. Use `df -h /` to view available space.
- c. Run the following command to upgrade the device.

```
$ sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

Here, `<archive_file_path>` specifies the CipherTrust Manager path to the signed archive file.

The signature of the archive file is verified and the upgrade is applied.

- d. Reboot the appliance as prompted. The following output indicates the upgrade has completed:

```
Upgrade succeeded  
Reboot the instance to finish the upgrade
```

Note

If you have lost the SSH session, you can access upgrade logs in `/home/ksadmin/`. Upgrade log files have the prefix `ks_upgrade`. Search for "Upgrade succeeded" to confirm the upgrade completed, and that you must reboot. Take care to verify the upgrade has completed before rebooting. You must always reboot to fully apply the upgrade.

- e. From the `ksadmin` session, run `systemctl status keysecure` to ensure that the CipherTrust Manager services have started. Alternatively, you can visit the CipherTrust Manager web console or attempt to connect with the `ksctl` CLI.
- f. If you need to verify the new version, look for the version displayed in CipherTrust Manager banner in the SSH session on reboot, as shown in the screenshot. Alternatively, log in the web console UI and click the information icon at the top right to display version information.

[illegible]

5. After the appliance reboots, re-build the cluster by creating a new cluster on this node.
6. Perform the upgrade on all other removed nodes, as outlined in step 4.

Note

If a previously used node is to be re-used, the cluster must first be deleted from that system.

7. Join new instances to the cluster.
8. Re-establish client access to the cluster.
9. **Re-download and re-install ksctl** to access commands and settings introduced in the new CipherTrust Manager software version.

Note

Previous versions of ksctl still work with new CipherTrust Manager software for existing features and settings.

Upgrade Troubleshooting

The following section lists some commonly encountered errors during upgrade and their solutions.

Occasional Error for Devices Upgraded from NextGen KeySecure 1.6 or 1.7

CipherTrust Manager appliances that at one time ran NextGen KeySecure software versions 1.6 or 1.7 occasionally fail to upgrade on future versions. The error reported is `E: Unable to locate package docker-compose-plugin`.

Solution: Contact customer support to resolve this state.

Configuring DNS Hosts

The CipherTrust Manager allows the Admin users to specify local DNS host entries, which eliminates the need to remember the IP addresses. While specifying a hostname, its IP address is required so that CipherTrust Manager can connect directly to that address whenever its hostname is specified for any outbound connection such as SMTP and Syslog Server.

Multiple hostnames can be assigned to an IP address. To do so, you need to create multiple records.

Example

```
ksctl dnshosts create --name 'example.com' --ip '172.168.59.22'
```

```
ksctl dnshosts create --name 'example1.com' --ip '172.168.59.22'
```

In the above example, IP `172.168.59.22` is assigned to multiple hostnames.

Similarly, multiple IP addresses can be assigned to a hostname.

Example

```
ksctl dnshosts update --id 'example.com' --ips '172.168.59.22' --ips  
'172.168.12.22' --type 'host-record'
```

In the above example, multiple IPs are assigned to the `example.com` hostname.

The DNS hostnames are unique in the system.

Note

Currently, local DNS-Host entries are not resolved while configuring SNMP and NTP servers.

The following operations can be performed:

- Create/get/update/delete DNS record
- List all DNS records

Creating DNS records

To create a DNS record, you need to define the following parameters:

- `name` - Hostname of the domain.
- `ip` - Host IP. This field is deprecated. Therefore, it is recommended to use `ips`.
- `ips` - Host IPs. You can provide single or multiple IPs if required.
- `type` - Type of the DNS host record. Possible values are:
 - `address` (default) - supports domain redirection. For example, if a DNS entry for the domain "example.com" exists with an IP address, the subdomain "abc.example.com" will also be redirected to the same IP as defined for the "example.com".
For entries with multiple IPs, round-robin is not supported. For example, if a DNS record is created with multiple IPs, only a specific IP will be returned every time during the domain name resolution process.
 - `host-record` - supports round-robin over multiple IPs, but doesn't support domain redirection. Therefore, the resolution only takes place when there's an exact match for the specified hostname.

If you don't provide `type` explicitly, the default value (`address`) is used.

Example 1

Syntax

```
ksctl dnshosts create --name <hostname> --ip <host-ip>
```

Request

```
ksctl dnshosts create --name 'example.com' --ip '52.82.16.43'
```

Response

```
{
  "id": "fbc4530f-519b-4a74-b7a3-edaec53afe3c",
  "uri": "kylo:kylo:solo:dns:fbc4530f-519b-4a74-b7a3-edaec53afe3c",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-05-10T08:18:35.455866Z",
  "name": "example.com",
  "ip": "52.82.16.43",
  "type": "address",
  "updatedAt": "2023-05-10T08:18:35.455866Z"
}
```

In the above example, the default `type` that is `address` is returned in the response.

Example 2 (`type` set to `address`)

Request

```
ksctl dnshosts create --id 'example.com' --ips '172.168.59.22' --type 'address'
```

Response

```
{
  "id": "c77e2053-8a00-4c84-b437-d3516b3907aa",
  "uri": "kylo:kylo:solo:dns:c77e2053-8a00-4c84-b437-d3516b3907aa",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-05-10T08:15:56.873007Z",
  "name": "example.com",
  "ips": [
    "172.168.59.22"
  ],
  "type": "address",
  "updatedAt": "2023-05-10T08:17:10.253454Z"
}
```

Example 3 (type set to host-record)

Request

```
ksctl dnshosts create --name 'example.com' --ips '172.168.59.22' --ips '172.168.12.22' --type 'host-record'
```

Response

```
{
  "id": "c77e2053-8a00-4c84-b437-d3516b3907aa",
  "uri": "kylo:kylo:solo:dns:c77e2053-8a00-4c84-b437-d3516b3907aa",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-05-10T08:15:56.873007Z",
  "name": "example.com",
  "ips": [
    "172.168.59.22",
    "172.168.12.22"
  ],
  "type": "host-record",
  "updatedAt": "2023-05-10T08:15:56.873007Z"
}
```

Getting details of DNS records

To get details of a DNS record, run:

Request

```
ksctl dnshosts get --id example.com
```

Response

```
{
  "id": "c77e2053-8a00-4c84-b437-d3516b3907aa",
  "uri": "kylo:kylo:solo:dns:c77e2053-8a00-4c84-b437-d3516b3907aa",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-05-10T08:15:56.873007Z",
  "name": "example.com",
  "ips": [
    "172.168.59.22",
    "172.168.12.22"
  ],
  "type": "host-record",
  "updatedAt": "2023-05-10T08:17:10.253454Z"
}
```

Updating DNS records

To update a DNS record, run:

Request

```
ksctl dnshosts update --id "example.com" --ips '52.63.25.46' --ips '172.34.12.62'
```

Response

```
{
  "id": "b0da914e-3c80-4d29-8f84-35b8384955be",
  "uri": "kylo:kylo:solo:dns:b0da914e-3c80-4d29-8f84-35b8384955be",
  "account": "kylo:kylo:admin:accounts:kylo",
  "application": "ncryptify:gemalto:admin:apps:kylo",
  "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
  "createdAt": "2023-05-11T08:20:48.823011Z",
  "name": "example.com",
  "ips": [
    "52.63.25.46",
    "172.34.12.62"
  ],
  "type": "host-record",
  "updatedAt": "2023-05-11T08:24:56.419976Z"
}
```

Getting list of DNS records

To get a list of all DNS records, run:

Request

```
ksctl dnshosts list
```

Response

```
{
  "skip": 0,
  "limit": 10,
  "total": 2,
  "resources": [
    {
      "id": "55a7e5d6-c1e1-40ba-a471-cfc32b700e31",
      "uri": "kylo:kylo:solo:dns:55a7e5d6-c1e1-40ba-a471-cfc32b700e31",
      "account": "kylo:kylo:admin:accounts:kylo",
      "application": "ncryptify:gemalto:admin:apps:kylo",
      "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
      "createdAt": "2023-05-11T08:22:31.025336Z",
      "name": "sample.com",
      "ip": "52.23.98.120",
      "type": "address",
      "updatedAt": "2023-05-11T08:22:31.025336Z"
    },
    {
      "id": "b0da914e-3c80-4d29-8f84-35b8384955be",
      "uri": "kylo:kylo:solo:dns:b0da914e-3c80-4d29-8f84-35b8384955be",
      "account": "kylo:kylo:admin:accounts:kylo",
      "application": "ncryptify:gemalto:admin:apps:kylo",
      "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
      "createdAt": "2023-05-11T08:20:48.823011Z",
      "name": "example.com",
      "ips": [
        "173.52.23.124",
        "52.12.25.23"
      ],
      "type": "host-record",
      "updatedAt": "2023-05-11T08:20:48.823011Z"
    }
  ]
}
```

Deleting DNS records

To delete a DNS record, run:

Request

```
ksctl dnshosts delete --id 'example.com'
```

There will be no response if DNS record is deleted successfully.

CLI Toolkit

This section describes how to install the CLI toolkit `ksctl` and use it to perform typical administrating tasks on CipherTrust Manager VM or instance.

To access CLI commands for any features or settings introduced in a new CipherTrust Manager software version, re-download and re-install ksctl after upgrade.

CLI Toolkit Installation

CipherTrust Manager includes a CLI toolkit, named **ksctl**, that can be downloaded and run locally to control a remote CipherTrust Manager.

Note

ksctl is designed to be run from a remote system, not on the CipherTrust Manager itself.

ksctl exclusively uses the REST API to communicate with CipherTrust Manager, so anything you can do with the CLI 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 adding users and groups, and end-user functions, such as creating keys.

To get started with **ksctl** you must do the following:

1. Download and unzip the **ksctl_images.zip** file.
2. Setup the *ksctl-os* file for your system. This includes setting some basic configuration variables for ksctl authentication.
3. (Optional) Further set **ksctl configuration variables**, if desired.
4. Run ksctl.

Download and unzip the ksctl_images.zip file

1. Enter the IP address of your CipherTrust Manager system in your browser.
2. If you are logged into CipherTrust Manager, select the **API** link at the top right. If you are logged out, select **API & CLI Documentation** at the bottom right.
3. Select the **** CLI Guide page and click on Download CLI**** button:



4. Unzip the **ksctl_images.zip** file.

Example set of available images in **ksctl_images.zip** file:

```
ksctl-darwin-amd64
```

```
ksctl-linux-amd64  
ksctl-win-amd64.exe
```

Setup the *ksctl-os* for your system

The `ksctl` utility can be set up on [Windows](#), [Linux](#), and [macOS](#).

Windows

1. After unzipping, rename `ksctl-win-amd64.exe` to `ksctl.exe`.

Note

ksctl.exe is a single executable with no dependencies, so it can be run from anywhere.

2. Move the **ksctl.exe** file to a folder easily accessed by the local command prompt.

Note

You can also add the location of the **ksctl.exe** file to your PATH variable or copy **ksctl.exe** to a location already in your PATH.

3. Create a **ksctl** configuration file using the local command prompt **cmd.exe**:

- a. Create a **.ksctl** directory in your **%HOMEPATH%** directory:

```
mkdir %HOMEPATH%\ksctl  
cd %HOMEPATH%\ksctl
```

- b. In the **%HOMEPATH%.ksctl** directory, create a **config.yaml** file with the following items:

```
KSCTL_USERNAME: admin  
KSCTL_PASSWORD: <your password>  
KSCTL_URL: https://<ciphertrust manager ipaddress> or https://<ciphertrust_hostname>  
KSCTL_NOSSLVERIFY: true
```


Tip

Enter `set homedir` to find or confirm your **%HOMEPATH%** directory.

Linux

1. After unzipping, rename the file of choice to **ksctl** and move it to a directory within your PATH.

Note

ksctl can be run from any directory.

2. Create a **ksctl** configuration file:
 - a. Create a **.ksctl** directory in your **\$HOME** directory:

```
mkdir $HOME/.ksctl
cd $HOME/.ksctl
```

- b. In the **\$HOME/.ksctl** directory create a **config.yaml** file with the following items:

```
KSCTL_USERNAME: admin
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<ciphertrust manager ipaddress> or https://<ciphertrust_hostname>
KSCTL_NOSSLVERIFY: true
```

3. Bash completions can be generated by typing `ksctl bashcomp` at the bash prompt and following the instructions.

macOS

1. After unzipping, rename **ksctl-darwin-amd64** to **ksctl** and move it to a directory within your PATH.
2. In Finder, right-click or control-click the **ksctl** executable and select **Open**. You cannot open the executable by double-clicking.

Note

For some versions of macOS, a prompt appears indicating that macOS cannot verify the developer. You need to dismiss the prompt and navigate into Security & Privacy settings to manually allow **ksctl**.

3. On the prompt, click **Open** to proceed.
4. Create a **ksctl** configuration file:
 - a. Create a **.ksctl** directory in your **\$HOME** directory:

```
mkdir $HOME/.ksctl
cd $HOME/.ksctl
```

- b. In the **\$HOME/.ksctl** directory create a **config.yaml** file with the following items:

```
KSCTL_USERNAME: admin
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<ciphertrust manager ipaddress> or https://<ciphertrust_hostname>
KSCTL_NOSSLVERIFY: true
```

Tip

Enter `env | grep HOME` to find your **\$HOME** directory.

ksctl Configuration Variables

Summary of **ksctl** configuration variables for the `config.yaml` configuration file:

```
KSCTL_USERNAME: <your username>
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<ciphertrust manager ipaddress> or https://<ciphertrust_localhost>
KSCTL_NOSSLVERIFY: <true or false. The default value is false.>
KSCTL_CONNECTION: <LDAP connection name>
KSCTL_TRUSTED_CA_FILE: <Filename of the Trusted CA. Required when KSCTL_NOSSLVERIFY is set to false.>
KSCTL_SERVER_NAME: <Used to verify the hostname on the server certificate. Required when KSCTL_NOSSLVERIFY is set to false, the hostname is not present on the client workstation `etc/hosts` file, or the hostname is not mapped on the local DNS
```

```
server>
KSCTL_VERBOSITY: <true or false for verbosity>
KSCTL_JWT: <JWT>
KSCTL_RESPFMT: <response output format, the default "json" is the only supported
value at present>
KSCTL_TIMEOUT: <Time limit in seconds to wait for a response. Set to `0` for no t
imeout. The default value is 30, if not set.>
KSCTL_AUTH_DOMAIN: <The ${cm} domain where the user is authenticated>
```

ksctl is easiest to use with the configuration file (config.yaml). However, those parameters can be overridden by using these flags on the command line:

```
--user (replaces KSCTL_USERNAME)
--password (replaces KSCTL_PASSWORD)
--url (replaces KSCTL_URL)
--connection (replaces KSCTL_CONNECTION)
--nosslverify (replaces KSCTL_NOSSLVERIFY=false)
--nosslverify=false (replaces KSCTL_NOSSLVERIFY=true)
--trusted-ca-file (replaces KSCTL_TRUSTED_CA_FILE)
--server-name (replaces KSCTL_SERVER_NAME)
--verbose (replaces KSCTL_VERBOSITY=false)
--verbose=false (replaces KSCTL_VERBOSITY=true)
--jwt (replaces KSCTL_JWT)
--respfmt (replaces KSCTL_RESPFMT)
--timeout int (replaces KSCTL_TIMEOUT)
```

Note

- If `KSCTL_VERBOSITY` and `KSCTL_NOSSLVERIFY` are set to true in the config.yaml file, you must use `--verbose=false` or `--nosslverify=false` flags to override them. This is because these two settings are boolean values.
- We recommend setting `KSCTL_TRUSTED_CA_FILE` and `KSCTL_SERVER_NAME` to verify the CipherTrust Manager [web interface](#) server certificate. In most cases, `ksctl` communicates with the web interface. When the [disk is encrypted and in the preboot stage](#), `ksctl` communicates with the preboot interface. In this state, we recommend using the `--trusted-ca-file` and `--server-name` flags to provide values matching the preboot interface certificate. It is also possible but not recommended to use `--nosslverify` to skip certificate verification.

Authenticating to ksctl

Setting the `config.yaml` as indicated in [Setup the ksctl OS for your system](#) means that most `ksctl` commands authenticate you as the admin user.

If you wish to authenticate as another user you can either:

- Run `ksctl login --user <username> --password <password>` command. All subsequent commands in the session keep the authentication context, and are run as the logged in user.

- Use `--user <username>` and `--password <password>` flags on each individual command.

As well all commands are run in the root domain by default. If you want to authenticate to a **child domain**, use the `--domain <domain_name>` flag. You can either:

- Run `ksctl login --domain <domain_name>` command. All subsequent commands in the session keep the authentication context, and are run in the specified domain.
- Use `--domain <domain_name>` flags on each individual command.

Run ksctl

Note

This is an example only; not all supported CLI commands and flags are shown.

1. At the command line prompt, enter **ksctl**. You will get the following example output:

```
ksctl
```

```
This command line utility can be used to interface with a CipherTrust Manager device.
```

```
Configuration parameters can be used to pass in the username, URL, etc. These parameters can be passed to the application in several ways.
```

```
When the configuration parameters are available from multiple sources, the order of precedence (highest to lowest) is:
```

- ```
1. Command line arguments
2. Configuration file passed via the command line
3. Environmental variables
4. Configuration file in the default location
```

```
The parameters passed in as command line arguments override everything else.
```

```
The structure of the configuration file must be based on 'config_example.yaml'. The name of the default configuration file is 'config.yaml'. It should be located in the '.ksctl' subdirectory of the user's home directory.
```

```
On Linux, this file should be located in: '~/.ksctl/config.yaml'.
```

On Windows, this file should be located in: 'C:/ksctl/config.yaml'.

If defined, proxy environment variables (i.e. 'http\_proxy') will be honored for all functions except 'init getpw'. 'init getpw' internally uses ssh which means that ssh must be configured to be proxy aware for the users local operating system.

#### Usage:

```
ksctl [command]
```

#### Available Commands:

|                |                                                                  |
|----------------|------------------------------------------------------------------|
| akeyless       | Akeyless Gateway                                                 |
| alarms         | Manage Alarms. List, state and clear commands.                   |
| backup         | Perform system or domain database backups.                       |
| backupkeys     | Manage Backup keys. Used for securing backups.                   |
| banners        | Manage Login Banners.                                            |
| bashcomp       | Generate bash completion script file for all commands and flags. |
| ca             | Manage Certificate Authorities and Certificates.                 |
| cckm           | Manage CipherTrust Cloud Key Manager.                            |
| changepw       | Change a user password.                                          |
| client-records | Get client record(s).                                            |
| clientmgmt     | Create Client Tokens and Register Clients.                       |
| cluster        | Manage Clusters.                                                 |
| .              |                                                                  |
| .              |                                                                  |
| .              |                                                                  |

#### Flags:

|                                   |                                                                                             |
|-----------------------------------|---------------------------------------------------------------------------------------------|
| <code>--auth-domain string</code> | The CipherTrust Manager Domain Where the user is created.                                   |
|                                   | By default the auth domain is root or provide the domain where user created.                |
| <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 except 'tokens create' command.      |
|                                   | By default the command will operate in the root domain or the domain the user is logged-in. |

```

-h, --help help for ksctl
--jwt string The JSON Web Token (JWT) - access token can be passed instead
 of user/password (optional). 'ksctl tokens create' creates a JWT.
--nossllverify Do not verify the certificate for SSL/HTTPS authentication (not recommended)
--password string CipherTrust Manager Server User Password. Do not use this flag to enter the password (masked) from terminal.
--respfmt string Response Output format (json is the only supported value at present, optional) (default "json")
--server-name string Name of the Server which is used to verify the hostname on the returned certificates unless nossllverify is given.
--timeout int Timeout in seconds for TCP connection attempts
--token string The refresh token returned from the login command to be passed instead of user/password (optional).
 'ksctl login' creates a token and writes it to the config file.
--trusted-ca-file string Filename of the Trusted CA to be read from.
--url string CipherTrust Manager Server URL
--user string CipherTrust Manager Server User Name
-v, --verbose Provide verbose output while executing command (optional)

Use "ksctl [command] --help" for more information about a command.

```

2. Try creating a key by entering the command `ksctl keys create --autoname`:

```
ksctl keys create --autoname
```

This will create output similar to the following:

```

{
 "id": "b979d9c7-1374-4b97-a274-dfaa20f8139b",
 "uri": "kylo:kylo:vault:keys:keys-f0f3e1lhqh5r7xavk3l0-rx-ktkmwgzf3-rudppxr4-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2016-12-15T15:35:51.322996318Z",
 "name": "Keys-f0F3eLlhQh5R7XAVK3l0_rx-KtkmWGzF3-rUDPpXRW4",

```

```

"updatedAt": "2016-12-15T15:35:51.322996318Z",
"material": "",
"usage": "blob",
"meta": null,
"version": 0,
"algorithm": "AES",
"size": 256,
"format": "raw",
"unexportable": false,
"undeletable": false,
"publickey": ""
}

```

## CLI Usage with JSON parameters

When providing JSON parameters to `ksctl` commands, the syntax differs between Windows and Linux.

### Linux/macOS CLI usage

On Linux/macOS, you can use standard JSON format for key-value pairs without any modifications.

**Example:**

```
ksctl backup create --scope domain --filters '[{"resourceType": "Keys"}, {"resourceType": "cte_policies"}]'
```

### Windows CLI usage

On Windows, you must escape the double quotation marks within the JSON string using a backslash (`\`) to ensure the command is correctly interpreted.

**Example:**

```
ksctl backup create --scope domain --filters "[{\"resourceType\": \"Keys\"}, {\"resourceType\": \"cte_policies\"}]"
```

## Batching Commands with Tokens

As described in [Tokens](#), a token is returned after logging in to the CipherTrust Manager, for example, using the `config.yaml` file or the `--user` and `--password` command line parameters. This token, when used with subsequent commands, can improve performance with the CipherTrust Manager, since verifying the token is much faster than looking up the user and password on each invocation.

If the user wishes to perform batched commands and desires greater performance, you can use this token as described in this section.

## Using tokens to perform operations



```
$ export KSCTL_JWT=$(ksctl tokens create --user sarah --password Sarah_pw1 | jq -r ".jwt")
```

### Note

If you have user sarah defined in your config.yaml file, the --user and --password variables are not needed.

For the purposes of this example, Sarah's `user_id` must be retrieved; it is used in the creation of a key.

### Get Sarah's ID

1. To retrieve Sarah's `user_id`, enter:

```
$ ksctl users list --name sar
```

Returns the following response:

```
{
 "skip": 0,
 "limit": 10,
 "total": 1,
 "resources": [
 {
 "created_at": "2017-06-22T16:25:56.441357Z",
 "email": "",
 "last_login": "2017-06-22T18:34:15.835436Z",
 "logins_count": 2,
 "name": "sarah",
 "nickname": "sarah",
 "updated_at": "2017-06-22T18:34:15.835436Z",
 "user_id": "local|77bf124c-c4f2-4a89-a777-be9f8c66d099",
 "username": "sarah",
 "failed_logins_count": 0,
 "account_lockout_at": null,
 "failed_logins_initial_attempt_at": null,
 "last_failed_login_at": null,
 "password_changed_at": "2017-06-22T16:25:56.441806Z",
 "password_change_required": false
 }
]
}
```

```
}
```

2. To assign Sarah's `user_id` to an environment variable, enter:

```
$ export SARAH=$(ksctl users list --name sarah | jq -r ".resources[0]" | jq -r ".user_id")
```

### Create a key

```
$ ksctl keys create --autoname --ownerid $SARAH --jsonfile keyparams.json
```

This command can be run in a loop to automatically create as many keys as desired. If your loop runs longer than the lifetime of your JWT, you can periodically refresh the token with a new create. The lifetime of your JWT is returned in the create call; see the `duration` field. The value is in seconds.

### Refresh the token

```
$ export KSCTL_JWT=$(ksctl tokens create --user sarah --password Sarah_pw1 | jq -r ".jwt")
```

As you can see, there is some setup effort involved in these commands. However, the payoff in ease-of-use and performance is worth the time spent.

## Example CLI User Set Up

The following example will create a non-admin user named **sarah** and create a key. **sarah** will own the key and will be able to do things with it.

We will create a second user, named **bob**, and show he does not have access to **sarah's** key.

Next, we will add **bob** to a group. **sarah** will add a permission to the key, granting members of the group some permissions to the key. We'll then show that **bob** now has these permissions.

You can try these steps using the API playground, curl, or by using the CLI commands.

This exercise assumes that the user has already installed the CLI tool (`ksctl`) and has created a `config.yaml` file that is specific to their installation. Note that we will use the built-in user in some cases and override them in others. Any command without user/password credentials is using the built-in (admin) user.

1. Create a new user called **sarah**.

```
$ ksctl users create -n sarah -p Sarah_pw1
```

The response to the create user command is a user resource that looks like:

```
{
```

```

"created_at": "2018-04-27T21:10:53.191577Z",
"email": "",
"last_login": null,
"logins_count": 0,
"name": "bob",
"nickname": "bob",
"updated_at": "2018-04-27T21:10:53.191577Z",
"user_id": " local|0c617800-58f0-4cf9-85bd-8d75037d8bfb",
"username": "sarah",
"failed_logins_count": 0,
"account_lockout_at": null,
"failed_logins_initial_attempt_at": null,
"last_failed_login_at": null,
"password_changed_at": "2018-04-27T21:10:53.192083Z",
"password_change_required": false
}

```

2. Now create an AES key, which will be owned by **sarah**.

We'll need to create a file containing the attributes of the key we want to create, then pass that file to the CLI tool:

### Windows

```
$ echo {"name":"aeskey"} > keyparams.json
```

### Linux

```
$ echo "{\"name\":\"aeskey\"}" > keyparams.json
```

### All systems

```
$ ksctl --user sarah --password sarah_pw1 keys create -j keyparams.json
```

The response to the create key command is a key resource that looks like:

```

{
 "id": "cf8c8c215e8077e11a7af5304fe77082791af13e3906d52ba7237eb6fd2df868",
 "uri": "kylo:kylo:vault:keys:aeskey-v0-HEU",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2018-04-27T20:56:11.731123593Z",

```

```

"name": "aeskey",
"updatedAt": "2018-04-27T20:56:11.731123593Z",
"usage": "blob",
"usageMask": 12,
"meta": {
 "ownerId": "local|0c617800-58f0-4cf9-85bd-8d75037d8bfb"
},
"algorithm": "AES",
"size": 256,
"format": "raw",
"unexportable": false,
"undeletable": false,
"defaultIV": "5bec41b53dd4a64aa25d64c175b05fb0",
"objectType": "Symmetric Key",
"activationDate": "2018-04-27T20:56:11.729121368Z",
"state": "Active"
}

```

3. **sarah** is the owner of the key, which grants her permission to read the key:

```
$ ksctl keys get -n aeskey
```

The response looks like:

```

{
 "id": "cf8c8c215e8077e11a7af5304fe77082791af13e3906d52ba7237eb6fd2df868",
 "uri": "kylo:kylo:vault:keys:aeskey-v0-HEU",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2018-04-27T20:56:11.731124Z",
 "name": "aeskey",
 "updatedAt": "2018-04-27T20:56:11.731124Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|0c617800-58f0-4cf9-85bd-8d75037d8bfb"
 },
 "algorithm": "AES",
 "size": 256,
 "format": "raw",
 "unexportable": false,
 "undeletable": false,

```

```

 "defaultIV": "5bec41b53dd4a64aa25d64c175b05fb0",
 "objectType": "Symmetric Key",
 "activationDate": "2018-04-27T20:56:11.729121Z",
 "state": "Active"
 }

```

4. Now we'll create a second user, named **bob**:

```
$ ksctl users create -n bob -p bob_password1
```

The response looks like:

```

{
 "created_at": "2018-04-27T21:10:53.191577Z",
 "email": "",
 "last_login": null,
 "logins_count": 0,
 "name": "bob",
 "nickname": "bob",
 "updated_at": "2018-04-27T21:10:53.191577Z",
 "user_id": "local|c7560d06-3628-4fb0-81e3-5881d2f8d7b3",
 "username": "bob",
 "failed_logins_count": 0,
 "account_lockout_at": null,
 "failed_logins_initial_attempt_at": null,
 "last_failed_login_at": null,
 "password_changed_at": "2018-04-27T21:10:53.192083Z",
 "password_change_required": false
}

```

5. We'll also create a new group **eng** and put **bob** in it:

```
$ ksctl groups create -i "local|c7560d06-3628-4fb0-81e3-5881d2f8d7b3" -n eng
```

The response looks like:

```

{
 "name": "eng",
 "created_at": "2018-04-27T21:15:36.644959Z",
 "updated_at": "2018-04-27T21:15:36.644959Z"
}

```

6. We will connect as **bob** and try to read the key. This will fail, since **bob** has no access to the key.

```
$ ksctl --user bob --password bob_password1 keys get -n aeskey
```

The response looks like:

```
{
 "code": 16,
 "codeDesc": "NCERRResourceNotFound: Resource not found"
}
```

7. Let us connect as **sarah** and give the **eng** group, of which **bob** is a member, permission to get info about the key.

### Windows

```
$ echo {"meta":{"permissions":{"ReadKey":["eng"]}}} > keypermissions.json
```

### Linux

```
$ echo "{\"meta\":{\"permissions\":{\"ReadKey\":[\"eng\"]}}}" > keypermissions.json
```

### All systems

```
$ ksctl --user sarah --password sarah_pw1 keys modify -n aeskey -j keypermissions.json
```

The response looks like:

```
{
 "id": "cf8c8c215e8077e11a7af5304fe77082791af13e3906d52ba7237eb6fd2df868",
 "uri": "kylo:kylo:vault:keys:aeskey-v0-HEU",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2018-04-27T20:56:11.731124Z",
 "name": "aeskey",
 "updatedAt": "2018-04-27T21:23:05.713305295Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
```

```

 "ownerId": "local|0c617800-58f0-4cf9-85bd-8d75037d8bfb",
 "permissions": {
 "ReadKey": [
 "eng"
]
 },
 "algorithm": "AES",
 "size": 256,
 "format": "raw",
 "unexportable": false,
 "undeletable": false,
 "defaultIV": "5bec41b53dd4a64aa25d64c175b05fb0",
 "objectType": "Symmetric Key",
 "activationDate": "2018-04-27T20:56:11.729121Z",
 "state": "Active"
 }
}

```

8. Now **bob** is able to get info about the key.

```
$ ksctl -user bob -password bob_password1 keys get -n aeskey
```

The response looks like:

```

{
 "id": "cf8c8c215e8077e11a7af5304fe77082791af13e3906d52ba7237eb6fd2df868",
 "uri": "kylo:kylo:vault:keys:aeskey-v0-HEU",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2018-04-27T20:56:11.731124Z",
 "name": "aeskey",
 "updatedAt": "2018-04-27T21:24:29.898123Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|0c617800-58f0-4cf9-85bd-8d75037d8bfb",
 "permissions": {
 "ReadKey": [
 "eng"
]
 }
 }
},

```

```

 "algorithm": "AES",
 "size": 256,
 "format": "raw",
 "unexportable": false,
 "undeletable": false,
 "defaultIV": "5bec41b53dd4a64aa25d64c175b05fb0",
 "objectType": "Symmetric Key",
 "activationDate": "2018-04-27T20:56:11.729121Z",
 "state": "Active"
 }

```

## Support CLI

The Support CLI provides root access to the appliance, and is intended for advanced troubleshooting in limited scenarios with customer support. If customer support indicates that the Support CLI is necessary, you use **ks\_support** (the Support CLI tool) to request an admin support token and then use that token to start a shell as the support user. The Support CLI comes with the CipherTrust Manager appliance and does not require installation.

### Warning

Only request an admin support token and open the support shell under direction from a Thales support engineer.

Customer Support can initiate a Support CLI shell with you:

- To reset the GUI admin password.
- To install Thales-approved scripts for issues that do not have any other workaround.
- To increase PED timeout values to use multifactor quorum feature (previously called M of N) when initializing the Thales k570 appliance.

## Request

1. Connect to your CipherTrust Manager instance via: `ssh ksadmin@ip.address`

The following prompt is displayed:

```
ksadmin@keysecure:~$
```

2. Start the `ks_support` tool, as follows:

```
sudo ks_support request
```



The following output is displayed:

```
Copy the following request to KeySecure support:
U3BhY2U6IHRoZSBmaW5hbCBmcm9udGllci4gVGhlc2UgYXJlIHRoZSB2b3lhZ2VzIG9mIHRoZSB
zdGFyc2hpcCBFbnRlcnByaXNlLiBjdmZmZl2ZS15ZWZyIG1pc3Npb246IHRvIGV4cGxvcmUgc3
RyYW5nZSBuZXcgc29ybGRzLiBUbyBzZWVrIG91dCBuZXcgbGlmZSBhbmQgbmV3IGNpdmlsaXphd
GlvbnMuIFRvIGJvbGRseSBnbyB3aGVyZSBubyBtYW4gaGFzIGdvcnUgYmVmb3JlLgo=
```

3. Select and copy all of the string after the `support:` (starting at `U3BhY2U6...` and ending with `...JlLgo=` in this example) to the clipboard.
4. Paste the string into an email that you send to your CipherTrust Manager support contact.

## Shell

After you have received a response from the CipherTrust Manager support:

1. Start the `ks_support` tool again:

```
sudo ks_support shell
```

The following prompt is displayed:

```
Paste text here, then press <ENTER> :
```

2. Copy the string you received from your CipherTrust Manager support contact.
3. Paste the string immediately after the prompt, as follows:

```
Paste text here, then press <ENTER>: SXQgaXMgYSBwZXJpb2Qgb2YgY2l2aWwgd2FyL
iBSZWJlbCBzcGFjZXNoaXBzLCBzdHJpa2luZyBmcm9tIGEGaGlkZGVuIGJhc2UsIGhhdmUgd29
uIHRoZWlyIGZpcnN0IHZpY3RvcnkgyWdhaw5zdCB0aGUgZXZpbCBHYWxhY3RpYyBFbXBpcmUuI
ER1cmlyB0aGUgYmF0dGx1LCBSZWJlbCBzcGllcyBtYW5hZ2VzIHRvIHN0ZWZsIHNlY3JldCB
wbGFucyB0byB0aGUgRWlwaXJl4oCZcyB1bHRpbWF0ZSB3ZWZwb24sIHRoZSBERUFUSCBTVEFSL
CBhbiBhcm1vcnVkbHNwYWNlIHN0YXRpb24gd2l0aCB1bm91Z2ggcG93ZXIgdG8gZGVzdHJveSB
hbiB1bnRpcmUgcGxhbmV0LiBQdXJzdWVrIGJ5IHRoZSBFbXBpcmXigJlziHNpbmlzdGVyIGFnZ
W50cywgUHVpbmNlc3MgTGVPYSByYWNlcyBob2l1IGFib2FyZCB0ZXIgc3RhcnoaXAsIGNlc3R
vZGlhbiBvZiB0aGUgc3RvbGVuIHBsYW5zIHRoYXQgY2FuIHNhdmUgaGVyIHB1b3BsZSBhbmQgc
mVzdG9yZSBmcmVlZG9tIHRvIHRoZSBnYWxheHkK
```

4. Press **Enter**.

## Success

If the string matches the necessary controls, the following support shell is displayed:

```
root@keysecure:~#
```

At this point, you can perform the steps as instructed by the CipherTrust Manager support personnel. When you log out from this shell, you are returned to the `ksadmin` shell prompt (as shown above). Note that the response that you receive can only be used once and is only valid for one day. Any attempted reuse or use after expiration results in an error message (as shown below).

## Failure

If the string does not match the required controls, the following message is returned:

```
Error: The challenge response was not accepted. Here are some possible reasons why:
* bad input - check the input for missing leading or trailing characters, additional whitespace, etc.
* expired challenge - challenges expire 1 day after being requested
* replayed - a challenge response can only be used once
```

If this occurs, relay these issues to your support personnel so that they can be remedied.

## Multiple Requests

In the event that you need support shell access multiple times in a single day (for example, if the appliance needs to be rebooted), then multiple requests may be generated and sent to your CipherTrust Manager support contact.

## Cleanup

The Support CLI tool automatically cleans up the files that it no longer needs, as they are used. However, it is possible that a number of requests may have been created that were not needed. To clean all of these requests from your system, run the following command as the `ksadmin` user:

```
sudo ks_support cleanup
```

This command returns no output except in the case of error. Any errors should be reported to your CipherTrust Manager support contact.

## Keys

CipherTrust Manager supports up to 1,000,000 keys total in the appliance or virtual instance, which can be distributed among multiple domains.

Keys can be created, modified, imported, exported, and deleted. Keys can be symmetric or asymmetric, variable sizes, and can be constrained to particular usages.

| Algorithm | Sizes (* = default) |
|-----------|---------------------|
| aria      | 128, 192, 256*      |
| seed      | 128*                |
| aes       | 128, 192, 256*      |
| tDES      | 128, 192*           |

| Algorithm   | Sizes (* = default)          |
|-------------|------------------------------|
| rsa         | 512, 1024*, 2048, 3072, 4096 |
| hmac-sha1   | >= 128, 160*                 |
| hmac-sha256 | 128, 192, 256*               |
| hmac-sha384 | 192, 288, 384*               |
| hmac-sha512 | 256, 384, 512*               |

| Supported Elliptic Curves (ec) (size is defined by the curve) |                 |                 |                 |
|---------------------------------------------------------------|-----------------|-----------------|-----------------|
| brainpoolP224r1                                               | brainpoolP224t1 | brainpoolP256r1 | brainpoolP256t1 |
| brainpoolP384r1                                               | brainpoolP384t1 | brainpoolP512r1 | brainpoolP512t1 |
| prime256v1                                                    | secp224k1       | secp224r1       | secp256k1       |
| secp384r1                                                     | secp521r1       | curve25519      | ed25519         |

Some of these curves have multiple names chosen by different standards organizations, as described in [RFC 4492](#), although CipherTrust Manager only supports the names listed above.

#### Note

A key name should not contain special characters such as angular brackets (<,>) and backslash (\).

#### Note

The CipherTrust Manager checks and corrects the parity bits (in each byte, the least significant bit is an odd-parity generated bit) of the triple DES key. Note that there can exist keys represented by different materials, producing identical ciphertext for the same plaintext input, as their parity bits can be modified without affecting the encryption bits.

## Supported Key Formats

Supported key formats for REST API.

| Object Type | Key Formats                                                                            |
|-------------|----------------------------------------------------------------------------------------|
| Symmetric   | <ul style="list-style-type: none"> <li>raw</li> <li>opaque</li> </ul>                  |
| Private Key | <ul style="list-style-type: none"> <li>pkcs1</li> <li>pkcs8</li> <li>pkcs12</li> </ul> |

| Object Type | Key Formats                                                                 |
|-------------|-----------------------------------------------------------------------------|
| Public Key  | <ul style="list-style-type: none"> <li>• pkcs1</li> <li>• pkcs8</li> </ul>  |
| Certificate | <ul style="list-style-type: none"> <li>• pkcs12</li> <li>• x.509</li> </ul> |
| Opaque      | <ul style="list-style-type: none"> <li>• raw</li> </ul>                     |
| Secret      | <ul style="list-style-type: none"> <li>• raw</li> </ul>                     |
| Split Key   | <ul style="list-style-type: none"> <li>• raw</li> </ul>                     |
| Template    | <ul style="list-style-type: none"> <li>• raw</li> </ul>                     |

**Note**

The supported formats for exporting an 'ed25519' private key are *PKCS8 (default)* and *PKCS12*, where the key material includes the base64-encoded value of the PFX file.

## Secrets

In addition, secret objects and opaque objects are also managed through the **Keys** menu in the UI. These managed objects are available through the `ksctl secret` commands in the CLI, and `v1/vault/secrets` endpoints in the REST API.

Secrets are managed objects that can store user defined data. This data can be a blob, a password, or a seed. There are two types of objects used to store secrets: a "Secret Object" and an "Opaque Object". Secrets support many of the same attributes that keys do.

## Create a Key

A key can be created using the GUI, KSCTL, or API.

**Note**

The CTE keys must be:

- AES128, AES-256, ARIA-128, or ARIA-256
- Shared with the CTE Clients group
- Set as exportable

Only the clients that support the CBC CS1 encryption mode can have GuardPoints that use a policy with a CBC CS1 key.

## Create a New Key Through ksctl CLI

To create a key using KSCTL, execute the command:

```
ksctl keys create -alg AES --name "testAESkey"
```

The response looks like:

```
{
 "id": "f0c2b40017dca0cdbbda90e2d224d0dcb4488ec412e156a594af3bc11a879b1a",
 "uri": "kylo:kylo:vault:keys:testaeskey-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2018-04-23T17:57:15.899060389Z",
 "name": "testAESkey",
 "updatedAt": "2018-04-23T17:57:15.899060389Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|ee328ec6-f37b-4e9c-85be-8867120b6b79"
 },
 "algorithm": "AES",
 "size": 256,
 "format": "raw",
 "unexportable": false,
 "undeletable": false,
 "defaultIV": "3ddc71eea888f8ec554b63759f371af9",
 "objectType": "Symmetric Key",
 "activationDate": "2018-04-23T17:57:15.886182496Z",
 "state": "Active",
 "keyCheckValue": "1f43a0"
}
```

## Create a New key Using a Template Through the ksctl CLI

Keys can be created using templates by providing the template id in the request. The key parameters present in the template are used to create a key. Users can also provide parameters in the key create request with the template id.

To create a key using the template:

### Syntax

```
ksctl keys create --name <template-name> --templateId <templateId>
```

### Example Request

```
ksctl keys create --name "TemplateKey" --templateId "63da49f4-422c-4a88-8af0-b66a15ebaad0"
```

### Example Response

```
{
 "id": "13b1564fced4f59b285fe323f383b32d6f088e7850b4f95bc49a2d6ee50f9a1",
 "uri": "kylo:kylo:vault:keys:templatekey-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2024-05-01T10:24:21.429179Z",
 "name": "TemplateKey",
 "updatedAt": "2024-05-01T10:24:21.429179Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|f718b511-0a3e-4a1f-b390-ec0d285b40d1"
 },
 "version": 0,
 "algorithm": "AES",
 "size": 256,
 "unexportable": false,
 "undeletable": false,
 "neverExported": true,
 "neverExportable": false,
 "emptyMaterial": false,
 "defaultIV": "d4ab6cc1c9c63900d0e8e14903986a5c",
 "sha1Fingerprint": "13f68100c120c731",
 "sha256Fingerprint": "8b45bf4ba07790c81b65fe5714a50c35d390a41264f727e6c3cc099ff319326c",
 "objectType": "Symmetric Key",
 "activationDate": "2024-05-01T10:24:21.422291Z",
 "state": "Active",
 "aliases": [
 {
 "alias": "TemplateKey",
 "type": "string",
 "index": 0
 }
],
 "uuid": "2c6cca78-195d-470f-a710-b80b80fc22b8",
 "muid": "2c6cca78-195d-470f-a710-b80b80fc22b842f6dae1-f235-471b-97f6-d772bf793cae",
 "keyCheckValue": "ea8324"
}
```

The parameters defined in the key create request and template takes precedence based on the access level of the user.

- Privileged users
- Restricted users

## Privileged Users

Privileged users are members of Key Users/Key Admins/admin groups. For privileged users, the precedence is given to the parameters defined in the key creation request over template parameters. The different parameters present in both key create request and template are merged. For Example, if a template is created for the AES Key and the key create request has the `algorithm` set to the RSA key. The preference will be given to the key create request parameters.

## Example Request

```
ksctl keys create --name "TemplateKey_PrivilegedUser" --templateId "63da49f4-422c-4a88-8af0-b66a15ebaad0" --alg "RSA" --size 2048
```

## Example Response

```
{
 "id": "e481e59d64394056ab35f8e9fe5b01cd9882d48a81424239b2e4682247eccc62",
 "uri": "kylo:kylo:vault:keys:templatekey-privilegeduser-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2024-05-01T10:26:51.105523Z",
 "name": "TemplateKey_PrivilegedUser",
 "updatedAt": "2024-05-01T10:26:51.105523Z",
 "usage": "sign",
 "usageMask": 3,
 "meta": {
 "ownerId": "local|651c9169-6923-44cb-9745-fcb1a76eae50"
 },
 "version": 0,
 "algorithm": "RSA",
 "size": 2048,
 "unexportable": false,
 "undeletable": false,
 "neverExported": true,
 "neverExportable": false,
 "emptyMaterial": false,
 "publickey": "-----BEGIN PUBLIC KEY-----\nMIIB...N3y2n0d+Yjo/2OT9cCqNd2\nZwIDAQAB\n-----END PUBLIC KEY-----\n",
 "defaultIV": "fa63563ad4cd74268791d1dde384e28f",
 "sha1Fingerprint": "97a31b5a79418acc",
 "sha256Fingerprint": "0edb2b708ff8318dfc34cdd4a186e4501ca195608a252487d29738e90680cd99",
 "objectType": "Private Key",
 "activationDate": "2024-05-01T10:26:50.855956Z",
 "state": "Active",
 "aliases": [
 {
 "alias": "TemplateKey_PrivilegedUser",
 "type": "string",
 "index": 0
 }
],
 "links": [
 {
 "id": "26268b74-6a98-47f4-95bb-9d8ea2cd615b",
 "uri": "kylo:kylo:vault:links:26268b74-6a98-47f4-95bb-9d8ea2cd615b",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2024-05-01T10:26:51.124271Z",
 "updatedAt": "2024-05-01T10:26:51.124271Z",
 "type": "publicKey",
 "source": "kylo:kylo:vault:keys:templatekey-privilegeduser-v0",
 "sourceID": "e481e59d64394056ab35f8e9fe5b01cd9882d48a81424239b2e46822"
```

```

47eecc62",
 "target": "kylo:kylo:vault:keys:templatekey-priviledgeduser-pub-v0",
 "targetID": "b0110e3865574ec6ab599dab42a40e6d4379b7f1a4894bfeaa5bfca3
4128258b",
 "index": 0
 }
],
"uuid": "c4d9f61d-4caf-4f63-b18e-93c6a504f124",
"muid": "c4d9f61d-4caf-4f63-b18e-93c6a504f124b2e99c95-308b-4e42-bb1f-641e23cd
9d34"
}

```

In the above example, the parameters defined in the key create request (alg "RSA", size 2048) take precedence over the template parameters.

## Restricted Users

Restricted users are members of the "Restricted Key Users" group. For restricted users, precedence is given to the template parameters over key create request parameters. The different parameters present in both key create request and template are merged. The key parameters allowed for restricted users in the key create request are:

- name
- description
- labels
- aliases
- material
- password
- meta (only ownerId)

For Restricted users, the "ownerId" defined in the template meta needs to be the same as provided in the key create request meta.

### Example Request

```

ksctl keys create --name "TemplateKey_RestrictedUser" --templateId "63da49f4-422
c-4a88-8af0-b66a15ebaad0" --labels "sales=HR"

```

### Example Response

```

{
 "id": "0b1b7fb3f006461a97bdc87aa4cf09613b3af0cda0ba4f7c8ac4518f24ee474e",
 "uri": "kylo:kylo:vault:keys:templatekey-restricteduser-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2024-05-01T10:30:42.867234Z",
 "name": "TemplateKey_RestrictedUser",

```



```

 "updatedAt": "2024-05-01T10:30:42.867234Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|622e0a4b-7bf9-4cfb-9226-ae716c716401"
 },
 "version": 0,
 "algorithm": "AES",
 "size": 256,
 "unexportable": false,
 "undeletable": false,
 "neverExported": true,
 "neverExportable": false,
 "emptyMaterial": false,
 "defaultIV": "03675c4382fd4e5c95b2643e8ea27723",
 "sha1Fingerprint": "61692c6192f36490",
 "sha256Fingerprint": "d974fc99c6492305902f03b73c779fc2278382bc2ca3bad05a5edb7
28c75cd29",
 "objectType": "Symmetric Key",
 "activationDate": "2024-05-01T10:30:42.866576Z",
 "state": "Active",
 "aliases": [
 {
 "alias": "TemplateKey_RestrictedUser",
 "type": "string",
 "index": 0
 }
],
 "uuid": "a3ea8fc4-6d58-4c1e-b46d-ac8926f82cbb",
 "muid": "a3ea8fc4-6d58-4c1e-b46d-ac8926f82cbb5461e345-45e6-45e5-8522-4ded5809
6d25",
 "labels": {
 "sales": "HR"
 },
 "keyCheckValue": "e749d7"
 }

```

## Create Template with keyAttributes Meta

### Example Request

```

ksctl templates create --name 'Test Template with Meta' --desc 'Symmetric' --la
bels 'sale=HR' --meta '{ "color":"Red"}' --key_attributes '{ "algorithm": "AES",
"size": 256 , "meta" : { "team": "finance", "customAttributes": [{ "name": "x-n
ae-attr", "value": "test" }] } }'

```

### Example Response

```

{
 "id": "5d24661e-a569-4c95-9891-1e7e56ddaee5",
 "uri": "kylo:kylo:vault:templates:test-template-with-meta-5d24661e-a569-4c9
5-9891-1e7e56ddaee5",
 "account": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2024-05-01T10:35:50.27364Z",
 "updatedAt": "2024-05-01T10:35:50.27364Z",
 "name": "Test Template with Meta",

```

```

 "description": "Symmetric",
 "meta": {
 "color": "Red"
 },
 "labels": {
 "sale": "HR"
 },
 "key_attributes": {
 "algorithm": "AES",
 "size": 256,
 "meta": {
 "team": "finance",
 "customAttributes": [
 {
 "name": "x-nae-attr",
 "value": "test"
 }
]
 }
 }
 }
}

```

## Restricted Key Users Meta Merge

### Example Request

```

ksctl keys create --name "TemplateKey_RestrictedUser_1" --templateId "5d24661e-a5
69-4c95-9891-1e7e56ddaee5" --labels "sales=finance"

```

### Example Response

```

{
 "id": "27f09b43042b466c84eeea39787d1ad9f3ef5ba3b9a941099d53f213d71fdb23",
 "uri": "kylo:kylo:vault:keys:templatekey-restricteduser-1-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2024-05-01T10:38:23.402756Z",
 "name": "TemplateKey_RestrictedUser_1",
 "updatedAt": "2024-05-01T10:38:23.402756Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "customAttributes": [
 {
 "name": "x-nae-attr",
 "value": "test"
 }
],
 "ownerId": "local|622e0a4b-7bf9-4cfb-9226-ae716c716401",
 "team": "finance"
 },
 "version": 0,
 "algorithm": "AES",
 "size": 256,
 "unexportable": false,
}

```

```

 "undeletable": false,
 "neverExported": true,
 "neverExportable": false,
 "emptyMaterial": false,
 "defaultIV": "a1baa294a74d8ed751052211db771b83",
 "sha1Fingerprint": "7cc7a011183f699c",
 "sha256Fingerprint": "840be93c6e77874ac99a40ae8bfa6c6705c392778b0fd2da73ebc3e
cbc49a34d",
 "objectType": "Symmetric Key",
 "activationDate": "2024-05-01T10:38:23.402178Z",
 "state": "Active",
 "aliases": [
 {
 "alias": "TemplateKey_RestrictedUser_1",
 "type": "string",
 "index": 0
 }
],
 "uuid": "6607ab8c-dfb3-484e-826b-a4cd841778ea",
 "muid": "6607ab8c-dfb3-484e-826b-a4cd841778ea691235e2-9af0-4ad1-8c66-da629be3
e5f2",
 "labels": {
 "sales": "finance"
 },
 "keyCheckValue": "2ba6d4"
 }
}

```

In the above example, you can see that meta is merged.

## Create a New Key Through the Web Console UI

1. Log in to the CipherTrust Manager domain.
2. Navigate to **Keys**.
3. Click **+ Add Key**.

The **Add Key** dialog displays.

4. (Optional) Provide a **Key Name**, **Description**, one or more **Aliases**, one or more **Alternative Names**, and one or more **Key Labels**. If you don't provide a key name, one will be auto-generated.
5. (Optional) Provide a key template, if desired. A key template must already exist.
  - a. Enable the **Create with key template** checkbox.
  - b. Click **Choose** to browse for key templates.
 

A window launches displaying the available templates.
  - c. Browse through the list or type the template name in the search box to find the desired template.
  - d. Select the radio button next to the desired template name, and click **Select**.

---

You are returned to the **Add Key** dialog.

The template auto-fills values for **Algorithm**, **Size/Curve**, **Key Properties**, and **Key Usage** in the rest of the **Key Usage** dialog.

6. Click **Next**.

The **Key Attributes** section is displayed.

7. Specify the **Object Type** to create.

- **Symmetric Key** includes AES, TDES, HMAC, ARIA, or SEED keys.
- **"Public/Private Key Pair"** includes RSA and EC keys.

8. Click **Generate Key Material**.

9. If you have not used a key template, or if you are signed in as a [privileged user](#) and want to override key template values, specify more values. Otherwise, skip to step 11.

#### Note

If you are signed in as a [restricted user](#) and have provided a key template, you cannot edit these values, and are immediately directed to review the key values. The CipherTrust Manager applies the key template parameters in this case.

- Select an **Algorithm** from the dropdown. For most algorithms, you must also select a **Size** from the dropdown. For **EC** key type, you select a **Curve**.
- You can optionally provide **Key Properties**.
- You can optionally provide **Key Usage**.

10. Click **Next**.

11. Review your selections. Click **Back** or **Edit** to modify any selection. Click **Add Key** to confirm.

The key details page is displayed.

## Importing an Existing Key or Certificate through the Web Console UI

Only [privileged users](#) can import existing keys or certificates into a CipherTrust Manager [domain](#). A [key template](#) is optional.

You can import plaintext keys/certificates (unencrypted), or keys/certificates wrapped (encrypted) with another key. Supported key types for import are AES, TDES, HMAC, RSA, EC, ARIA, or SEED. The supported certificate format for import is X.509 with DER or PEM encoding.

## Wrapping a Key for Import

If you choose to import a wrapped key, the wrapping key must be present on CipherTrust Manager. You can use a wrapping key with a supported wrapping key algorithm generated on CipherTrust Manager or generated externally. The high-level wrapping process varies based on your choice:

- **CipherTrust Manager-generated wrapping key:** Export the wrapping key to the location of the desired key, use the wrapping key to wrap the desired key, and import the wrapped key.
- **External wrapping key:** Use the wrapping key to wrap the desired key, import the wrapping key to CipherTrust Manager, and import the wrapped key.

Supported wrapping key algorithms are AES128, AES192, AES256, RSA2048, RSA3072, or RSA4096. Supported wrapping methods are AES key wrap, AES key wrap with padding, RSA encryption, and RSA AES key wrap with padding, depending on which key type you are wrapping.

| Wrapped Object Type                            | Supported Wrapping Methods                               |
|------------------------------------------------|----------------------------------------------------------|
| Symmetric key (AES, TDES, ARIA, SEED, or HMAC) | AES key wrap, AES key wrap with padding, RSA encryption  |
| RSA Key                                        | AES key wrap with padding, RSA AES key wrap with padding |
| EC Key                                         | AES key wrap with padding                                |
| Certificate                                    | AES key wrap with padding                                |

### Caution

We strongly discourage using RSA512 or RSA1024, as this algorithm and key length is no longer considered secure enough for this purpose, as outlined in [NIST Special Publication 800-131A Revision 2](#), section 6 Key Agreement and Key Transport Using RSA.

There are performance and convenience considerations for each wrapping method. In general, AES wrapping methods are faster and require less processing power than RSA encryption. However, there are more steps involved to securely transfer an AES key between CipherTrust Manager and an external system as compared to an RSA key.

We have the following guidelines to help you plan secure and efficient key wrapping outside of CipherTrust Manager.

- Generate the wrapping key on CipherTrust Manager. CipherTrust Manager's key generation uses high-quality, trusted entropy sources.
- Use CipherTrust Application Data Protection (CADP) to wrap keys on external systems. The following pages demonstrate key wrapping on each CADP client type.
  - [Wrapping keys with CADP for C](#)
  - [Wrapping keys with CADP for Java](#)

- 
- Wrapping keys with CADP for .NET Core
  - For a one-time import of less than a hundred symmetric keys, use RSA encryption. Generate an RSA key pair on CipherTrust Manager, export the public to the other system, wrap the symmetric keys, and import the wrapped keys. CipherTrust Manager can decrypt the key material with the private RSA key already present.
  - If you are importing more than a hundred keys, or you expect to import keys repeatedly, use AES key wrap or AES key wrap with padding. To securely share an AES key for wrapping between CipherTrust Manager and the other system, follow one of these two options:
    - **CipherTrust Manager-generated AES key:** Generate an RSA key pair on the other system, and import the public key to CipherTrust Manager. When you export the AES key from CipherTrust Manager, wrap it with RSA encryption using the public RSA key. Transfer the wrapped AES key to the other system, which can then unwrap the AES key with the private key.
    - **Externally-generated AES key:** Generate an RSA key pair in CipherTrust Manager, and export the public key to the other system. Wrap the AES key with RSA encryption on the other system. Import the wrapped key, so that CipherTrust Manager can unwrap with the RSA private key.

## Import a Key

To import a key:

1. Login to the desired CipherTrust Manager domain as a privileged user.
2. Navigate to **Keys**.
3. Click **+ Add Key**.

The **Add Key** dialog displays.

4. (Optional) Provide a **Key Name**, **Description**, one or more **Aliases**, one or more **Alternative Names**, and one or more **Key Labels**. If you don't provide a key name, one will be auto-generated.
5. (Optional) Provide a key template, if desired.
  - a. Enable the **Create with key template checkbox**.
  - b. Type the key template name into the text field or click **Choose** to select a key template from the list.
6. Click **Next**.

The **Key Attributes** section is displayed.

7. Specify the **Object Type** to import.
  - **Symmetric key:** This includes AES, TDES, HMAC, ARIA, or SEED keys.
  - **Certificate:** This includes X.509 certificates with PEM or DER encoding.

- **Public/Private Key Pair:** This includes RSA and EC keys.

8. Under **Key Material Origin**, select **Import Key Material** or **Import Wrapped Key Material**.
9. Specify related settings for the key you are importing.

#### Note

- No additional settings are required for certificate import.
- Ensure that curve ID 'ed25519' is specified while importing the 'ed25519' key, certificate, or PKCS#12 bundle. You can import either a PEM-encoded PKCS#8 format 'ed25519' private key or a base64-encoded PFX file in the PKCS#12 format. Additionally, you can import X509 PEM-encoded certificate or hex encoded X.509 DER-encoded certificate.

## Symmetric key

To import key material, you must select **HEX** or **Base64** encoding for symmetric keys.

To import wrapped key material, only Base64 encoding is supported. You do not make an encoding selection for wrapped key material.

## Public/Private Key Pair

If the key is in PKCS#12 format, toggle the **PKCS#12 format** to on.

10. Provide the **Key Material** as a file upload or text.
11. You can optionally provide **Key Properties** and **Key Usage**.

#### Note

These values are auto-filled if you specified a key template in step 5. You can change the values to override the template values.

12. Click **Next**.
13. (Public/Private Key PKCS#12 import only) Provide **Private Key Credentials**.
  - a. Select the **Credential Type**, **Use Password** or **Use Secret**.

## Password

Type the base64-encoded password in the **Password** text box.

---

## Secret

Search for a secret on the CipherTrust Manager by name. The search is conducted on all keys of object type "secret data". Enable the radio button for the desired secret.

b. Click **Next**.

14. (Import wrapped key material only) Select the wrapping method and wrapping key.

- a. For the **Wrapping Encryption Algorithm**, select **AES Key Wrap**, **AES Key Wrap with Padding**, **RSA Encryption**, or **RSA AES Key Wrap with Padding**. The options displayed depend on the format of the imported key. Refer to [Wrapping a Key for Import](#) for details of supported wrapping methods.
- b. For **RSA AES Key Wrap with Padding** method, you must also select **AES Key Size** and **RSA Padding** values from dropdowns.
- c. Search for the wrapping key's name in the list. There is a search bar provided.
- d. Enable the radio button for the desired wrapping key.
- e. Click **Next**.

15. Review the details of the key to import. Click **Back** or **Edit** to modify any selection. Click **Add Key** to confirm.

The key details page is displayed.

## Exporting a key in pkcs12/pfx format

### Exporting a private key for the first time

To export a private key from CipherTrust Manager to password protected PKCS#12/PFX format:

1. Log on to the CipherTrust Manager GUI as administrator.
2. Navigate to **Keys**.
3. Click the ellipsis icon corresponding to the key and select **Export**.

The Export Key (in pkcs12 format) wizard is displayed. Follow the steps to complete the setup.

- a. [Generate CSR](#)
- b. [Issue Certificate](#)
- c. [Download/Copy Certificate](#)
- d. [Enter Password](#)
- e. [Download Exported Key](#)



**Note**

The asymmetric key with algorithm – RSA and EC can only be exported. Make sure that the key is exportable.

**Generate CSR**

1. On the Generate CSR screen, provide the following details for the certificate signing request.

| Field       | Description                                                                   |
|-------------|-------------------------------------------------------------------------------|
| Common Name | Fully qualified domain name that you wish to secure. It is a mandatory field. |
| State       | Province, region, county, or state.                                           |
| Org Name    | legal name of a company or entity.                                            |
| Email       | The organization contact                                                      |
| Country     | The two-letter ISO code for the country where your organization is located.   |
| City        | City where your organization is located.                                      |
| Org Unit    | Internal organization department/division name.                               |

2. Click **Next** to move to the Issue Certificate screen.

**Issue Certificate**

1. On the issue certificate screen, specify:

| Field                | Description                                                                                                                  |
|----------------------|------------------------------------------------------------------------------------------------------------------------------|
| Local CA             | Issues and signs the certificate.                                                                                            |
| Certificate duration | Validity of the certificate in days. Its value can range from 1 to 365 days. The default certificate duration is "365 days". |
| Certificate Purpose  | Defines the purpose of the certificate. The default value is "client".                                                       |

2. Click **Next** to move to the Display/Save Certificate screen.

---

## Download/Copy Certificate

The Download/Copy Certificate screen shows whether the certificate to export is issued.

1. Click either **Download Certificate** to download the certificate in the .pem format or click **Copy Certificate to Clipboard** to copy the certificate.
2. Click **Next** to move to the Enter Password screen.

## Enter Password

On the Enter Password screen:

1. In **Password**, provide a password for the exported key in the plain text (ASCII) format. UI will do the base64 encoding before submitting it to the backend.
2. In **Confirm Password**, reenter the same password for the exported key.

The above fields are mandatory and should match exactly before continuing the wizard.

3. Click **Export Key** to export the key.

## Download Exported Key

The Download Exported Key screen shows the status of the key exported.

1. Select **Download/Copy Key in Format** as either **Base64** or **ASCII**.
2. Click either **Download Key** to download the exported key or click **Copy Key to Clipboard** to copy the exported key.
3. Click **Close**.

## Re-exporting an exported private key

To export a private key that has been exported before from CipherTrust Manager to password-protected PKCS#12/PFX format:

1. Log on to the CipherTrust Manager GUI as administrator.
2. Navigate to **Keys**.
3. Click the ellipsis icon corresponding to the key, select **Export**, and select the key that is exported before.

The Export Key Warning (in pkcs12 format) wizard is displayed. Read the warning carefully.

4. Select either **Use existing certificate and Update Password only** or **Proceed with the export key and replace the existing certificate**.

- **Use existing certificate and Update Password only**

Choosing this option will add a new password to the existing certificate. Follow the steps to complete the setup.

- a. Enter a password. Follow the steps mentioned [here](#).
- b. Download the exported key using the steps mentioned [here](#).

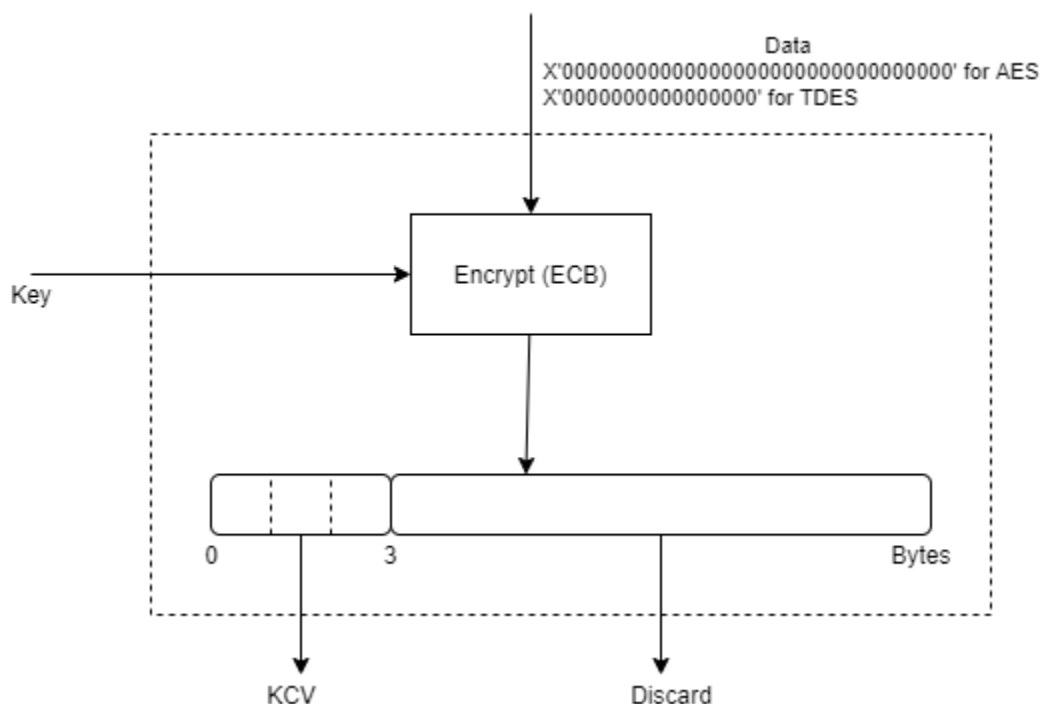
- **Proceed with the export key and replace the existing certificate**

Choosing this option will replace the existing certificate. Follow the steps mentioned in the [Exporting a private key for the first time](#) section to complete the setup.

## Computing Key Check Value on CipherTrust Manager

For AES and TDES keys, the key checksum value (KCV) is computed by performing an encryption operation on a set of empty bytes.

The key check value is computed by encrypting 8 bytes for DES and 16 bytes for AES, each with null (0x00) bytes, with the key to be checked using the default cipher and mode (for example, ECB) and retaining the 3 highest-order bytes of the encrypted result.



Use the following command to get the Key Check Value for any symmetric AES or TDES keys. The object is specified by the key name or ID.

```
ksctl keys generate-kcv --name mykey
```

## Configuring Key Rotation Frequency Days

The key rotation frequency days defines the number of days from key creation date to rotate the key. It should be

*greater than or equal to 0*. Default is an empty string. If `rotation-frequency-days` is set to "0", the `rotation-frequency-days` will set to an empty string and auto rotation of the key will be disabled.

The upper limit for key rotation frequency days is *99999 days*.

For more details, refer to [Default Schedulers](#).

### Example (Key Create)

```
ksctl keys create --name 'rotation_test' --rotation-frequency-days "5"
```

### Response

```
{
 "id": "a5b10dded7d7471c96d0a7cc7240ce365678389075c54f82aede433dbfe075ce",
 "uri": "kylo:kylo:vault:keys:rotation-test-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2024-01-19T09:30:42.640298Z",
 "name": "rotation_test",
 "updatedAt": "2024-01-19T09:30:42.640298Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|fc4915ff-3d3c-434f-8902-54a7f923a285"
 },
 "version": 0,
 "algorithm": "AES",
 "size": 256,
 "unexportable": false,
 "undeletable": false,
 "neverExported": true,
 "neverExportable": false,
 "emptyMaterial": false,
 "defaultIV": "1c2de986ac51f7731752fa3a2cf17da0",
 "sha1Fingerprint": "7ef3475b6c0d9891",
 "sha256Fingerprint": "98606d4b2e75ce40995c993bf1e2ecfb8343e06c5402883317bb763180daf6e7",
 "objectType": "Symmetric Key",
 "activationDate": "2024-01-19T09:30:42.638279Z",
 "scheduledRotationDate": "2024-01-24T09:30:42.638791218Z",
 "rotationFrequencyDays": "5",
 "state": "Active",
 "aliases": [
 {
 "alias": "rotation_test",
 "type": "string",
 "index": 0
 }
],
 "uuid": "c3d10d41-e65f-4e62-8381-c5df1004651e",
 "muid": "c3d10d41-e65f-4e62-8381-c5df1004651ed64e5c8d-3942-4593-ae54-9d80552dc078",
 "keyCheckValue": "4bb0ed"
}
```

## Example (Key Update)

```
ksctl keys modify --name 'rotation_test' --rotation-frequency-days "10"
```

## Response

```
{
 "id": "5406bfa581ed4059a4d11caf8d081d27465681a2b3ca45a1b30d4b8e4ed73a3b",
 "uri": "kylo:kylo:vault:keys:rotation-test-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2024-04-22T06:22:13.805753Z",
 "name": "rotation_test",
 "updatedAt": "2024-04-22T06:23:00.824459Z",
 "activationDate": "2024-04-22T06:22:13.800364Z",
 "state": "Active",
 "scheduledRotationDate": "2024-05-02T06:22:13.805753Z",
 "rotationFrequencyDays": "10",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|e72ec0c1-6e57-40ac-ace8-8a0bcf8db2ad",
 "permissions": {}
 },
 "objectType": "Symmetric Key",
 "aliases": [
 {
 "alias": "rotation_test",
 "type": "string",
 "index": 0
 }
],
 "sha1Fingerprint": "e4a574f3a53f114b",
 "sha256Fingerprint": "583480225a67b4c47e9aea74bb062aea5f4061fad299414b234ee2f69ed61bcc",
 "defaultIV": "cc4f89221c36f2286f569ff6d21176e6",
 "version": 0,
 "algorithm": "AES",
 "size": 256,
 "unexportable": true,
 "undeletable": true,
 "neverExported": true,
 "neverExportable": true,
 "emptyMaterial": false,
 "uuid": "dcfdefe6-388f-4078-97a3-513a50db1782",
 "muid": "dcfdefe6-388f-4078-97a3-513a50db17827db3f0a9-aad1-4b57-9fe8-f2cc87ef3764",
 "keyCheckValue": "b23bd4"
}
```

## Setting Key Permissions

User groups can be granted permissions to perform operations with a particular key such as read, encrypt, and decrypt data. You can set key permissions on the Key Access section of a key's detail page or in the Meta tab. This is done after creation.

Key permissions act at the *group* level. The **Key Usage** settings set at creation, and modifiable under the **KMIP** tab in key details act at the *key* level. This means that if a key's **Key Usage** does not allow encryption, *no* group can encrypt with that key, no matter what **Key Permissions** are granted to the group.

Key permissions can be set using the GUI, KSCTL, or API.

## To Set Key Permissions Using the GUI

1. Log on to the CipherTrust Manager console as administrator.
2. Open the **Keys** application.
3. Under **Key Name** column, click the key name link. The key details are displayed.
4. Expand the **Key Access** section.
5. Click **General** or **NAE** depending on the interface you want to update.
6. Change the **Key Owner**, if desired, from the dropdown.
7. Check or uncheck permissions for the listed groups. You can search for a group by name.
  - In the **General** tab you can check or uncheck **Read, Use, Decrypt, Sign, Verify** or **All**.
  - In the **NAE** tab you can check or uncheck **Decrypt, Encrypt**, or **All**.
8. Click **Update**.

## Making Keys Exportable or Deletable

You can control whether a key can be exported or deleted.

1. Log on to the CipherTrust Manager console as a user in the **Key Admin** or **admin group**, such as the `admin` user.
2. Open the **Keys** application.
3. Under **Key Name** column, click the key name link. The key details are displayed.
4. To enable or disable export for the key, click **Exportable** toggle near the top of the screen.
5. To enable or disable deletion for the key, click the **Deletable** toggle near the top of the screen.

In the example image below, the key is set to be both exportable and deletable.

|               |                                                                                   |             |                                                                                     |
|---------------|-----------------------------------------------------------------------------------|-------------|-------------------------------------------------------------------------------------|
| Owner         | local   admin                                                                     | Object Type | Symmetric Key                                                                       |
| Created       | 06 Mar 2021, 22:39                                                                | Algorithm   | AES                                                                                 |
| Last Modified | 06 Mar 2021, 23:03                                                                | Size        | 256                                                                                 |
| Exportable    |  | Deletable   |  |

## Versioned Keys

### KeySecure Classic

A KeySecure Classic versioned key is a key that maintains a single set of key metadata but contains multiple sets of key data. Each set of key data belongs to a unique version of the key.

A KeySecure Classic crypto operation using a versioned key prepends a header containing key version information to the resulting ciphertext. A crypto operation using an unversioned key does not prepend the ciphertext with this header.

### CipherTrust Manager

The CipherTrust Manager handling of versioned keys depends on the interface used and the setting of key flag (versionedKey). For backward compatibility with KeySecure Classic, when the NAE-XML interface is used **and** if the key flag (versionedKey = true), the key is handled as a versioned key and the crypto operation prepends a header to the resulting ciphertext. When the REST interface is used, crypto operations always produce unadorned ciphertext with no header, as KeySecure Classic produces when using unversioned keys.

Similarly, if you perform the inverse crypto operations on the ciphertext encrypted using a versioned key such as decrypt or verify, the operation must be done via the NAE-XML interface, since the REST interface does not handle these header bytes.

To maintain backward compatibility with KeySecure Classic, when you create a key through the NAE-XML interface with the **<Versioned/>** element present, CipherTrust Manager server adds a flag to the key (versionedKey = true). This flag indicates that when the key is used via the NAE-XML interface, the key should behave like a legacy NAE-XML versioned key (i.e. the produced ciphertext results that contain the legacy header bytes). This internal flag is only honored when the key is used via the NAE-XML interface; the flag has no effect on REST interface interactions.

#### Note

The (versionedKey = true) flag can be created through the REST interface. Detailed documentation is available in the "API playground". To create the (versionedKey = true) flag via the GUI, under the Key Details page > NAE tab, enable the Versioned Key checkbox.

### Note

When using NAE, 'VersionedKey' information stored in KeySecure Classic and CipherTrust Manager cannot be exported by or imported from either product. If 'VersionedKey' must be set, it must be noted by the operator that it exists and turned on in the GUI after the import has occurred. An alternative to export/import is to run a migration from KeySecure Classic to the CipherTrust Manager as this operation preserves 'VersionedKey' information.

## Key States

A key can be in any of the following states at any given time:

- **Pre-Active:** The key exists, but should not be used for any cryptographic purpose.
- **Active:** By default, the key can be used for any cryptographic purpose.

You can set the **Protect Stop Date** to a date in the past to restrict the key's usage to decryption, signature verification, unwrapping, and MAC verification operations. You can also [reactivate](#) to unset a **Protect Stop Date** that has passed and restore all cryptographic operations. These key state changes are available through the REST API and CLI interfaces.

- **Deactivated:** The key **cannot** be used for cryptographic operations. You can [reactivate](#) a deactivated key through the REST API and CLI interfaces.
- **Compromised:** The key should **not** be used for cryptographic operations such as encryption, signing, wrapping, MACing, and deriving. The key should only be used for decryption, signature verification, unwrapping, and MAC verification.
- **Destroyed:** The key should not be used for any cryptographic purpose.
- **Destroyed Compromised:** The key should not be used for any cryptographic purpose. Its compromised status should be retained for audit or security purposes.

You can set the **Protect Stop Date** to restrict or allow encryption, signing, wrapping, and MAC operations. In addition, you can [reactivate a key](#) to re-enable all or some cryptographic operations.

Besides reactivation, state transitions conform to the 1.4 KMIP specification.

## Equivalencies to NAE States

As of CipherTrust Manager version 2.5, all values for the `<KeyVersionState>` element in the NAE-XML interface map directly to the above key states. Changing a key state in the KMIP, REST API, GUI, or CLI results in a `<KeyVersionState>` change in NAE-XML. Similarly, changing `<KeyVersionState>` in NAE-XML results in a key state change in KMIP, REST API, GUI and CLI.

The following table shows the equivalent key states to the NAE-XML `<KeyVersionState>` element.



| NAE-XML KeyVersionState | Equivalent Key State(s)                                           | Allowed Operations                                                   |
|-------------------------|-------------------------------------------------------------------|----------------------------------------------------------------------|
| Pre-active              | Pre-active                                                        | None                                                                 |
| Active                  | Active with <b>Protect Stop Date</b> unset, or set in the future  | Any cryptographic operation                                          |
| Restricted              | Active with <b>Protect Stop Date</b> set in the past, Compromised | Decryption, signature verification, unwrapping, and MAC verification |
| Retired                 | Deactivated                                                       | None                                                                 |
| Wiped                   | Destroyed, DestroyedCompromised                                   | None                                                                 |

## Reactivate a Key

You can reactivate a key with the REST API `/v1/vault/keys2/{id}/reactivate` endpoint (described in the API playground), or the `ksctl keys reactivate` command.

### Note

You cannot reactivate a key through requests to the KMIP interface.

There are three state transitions that occur through reactivation.

- **Deactivated To Active:** This returns a Deactivated key to an Active state without setting a **Protect Stop Date**. All cryptographic operations are allowed.

### Example CLI command:

```
ksctl keys reactivate --name <key-id> --type name --reason toreactivate DeactivatedToActive
```

To cause the same state transition in the NAE-XML interface, change the `<KeyVersionState>` element from `Retired` to `Active`.

- **Active Protect Stop to Active:** This removes a **Protect Stop Date** for Active keys, if the **Protect Stop Date** has passed. All cryptographic operations are allowed.

### Example CLI command:

```
ksctl keys reactivate --name <key-id> --type name --reason toreactivate ActiveProtectStopToActive
```

To cause the same state transition in the NAE-XML interface, change the `<KeyVersionState>` element from `Restricted` to `Active`.

- **Deactivated to Active with Protect Stop:** This returns a Deactivated key to an Active state and sets the Protect Stop date to the current time. Decryption, signature verification, unwrapping, and MAC verification are allowed.

**Example CLI command:**

```
ksctl keys reactivate --name <key-id> --type name --reason to reactivate DeactivatedToActiveProtectStop
```

To cause the same state transition in the NAE-XML interface, change the `<KeyVersionState>` element from `Retired` to `Restricted`.

## Effects of Negative Actions on Keys

When you delete, destroy, revoke, or archive a key, the following happens:

- **Delete:** The key is deleted from the database altogether, including the key metadata and key material.

### Note

- Only CTE Administrators (users of the **CTE Admins** group) can delete the CTE keys.
- An in-use CTE key cannot be deleted. Before you delete a CTE key, make sure that no key rule is using the key.

- **Destroy:** The key material is deleted from the database, while the key metadata is kept.
- **Revoke:** The key state is set depending on the provided revocation reason. A revocation reason of "key compromise" or "CA compromise" results in a key state of "Compromised". Any other revocation reason results in a key state of "Deactivated". The key material is available on the server.
  - Deactivated keys cannot be used for any cryptographic operation.
  - Compromised keys can be used only for decryption, signature verification, unwrapping, and MAC verification. They **cannot** be used for encryption, signing, wrapping, MACing, and deriving.
- **Archive:** The key cannot be used for any cryptographic operations. Later the key can be recovered and used or modified, as required.

## Key Rotation

The key rotation operation generates a new version of a key with same key name and attributes, but with new key material. The operation provides an easy way to rotate keys in bulk periodically at the specified time.

**Note**

**Scheduler** itself is a generic service, so it provides granularity of a minute. A job configuration should take the nature of operation information into consideration. For example, rotating the same set of multiple keys every minute is a highly unlikely use case.

You must be in the Key Admins [group](#), and logged into the root [domain](#) to manage key rotation job configurations.

## Scheduling Key Rotation

The CipherTrust Manager can periodically run key rotation jobs in the background. This can be achieved by creating scheduler configurations for key rotation.

This section provides examples of key rotation scheduler configurations using the ksctl utility. Refer to the ksctl or API documentation for details.

To create a scheduler configuration for key rotation, run the command:

```
ksctl scheduler configs create key-rotation --name <config-name> --run-at
"<cron-expression-format>"
```

Here,

- **<config-name>**: Name for the job configuration.
- **<cron-expression-format>**: Time when the job runs. The format must have *five* fields. If the number of fields is not equal to five, the format becomes invalid. Refer to [Time Specification](#) for details.

Optionally, you can specify the following:

- **--key-query-json**: Query used for determining the keys to be rotated. By default, all keys are rotated. The query should be specified as a JSON blob. Refer to the API documentation for the JSON schema associated with the key query.
- **--metadata-json**: Metadata for the key that is replaced by the key rotation operation. The metadata should be specified as a JSON blob. The key metadata is also a JSON blob. The supplied metadata is added to the key metadata (existing data is overwritten). For example, this provides a mechanism to change the NAE state of a key by supplying `{"naeState": "Restricted"}`.
- **--deactivate-replaced-key**: Time interval in seconds ( $\geq 0$ ) after which the key that is replaced by the key rotation procedure is deactivated. By default, the key is not deactivated.

**Note**

Refer to the ksctl or API documentation for the complete list of parameters needed for scheduling key

rotation operations.

## Examples

### Note

The examples are provided for the Windows platform. To run these examples on Linux, see [CLI usage with JSON parameters](#).

To schedule a key rotation everyday at 9 AM, and modify the metadata of the key that is replaced, run the command:

```
ksctl scheduler configs create key-rotation --name change-meta --run-at "0 9 * * *" --metadata-json "{\"naeState\": \"Restricted\"}"
```

To rotate all AES keys every Monday at 9 AM:

```
ksctl scheduler configs create key-rotation --name rotate-aes --run-at "0 9 * * 1" --key-query-json "{\"algorithm\": \"AES\"}"
```

To activate the above key rotation on May 8, 2021 at 5:57:51 PM:

```
ksctl scheduler configs create key-rotation --name rotate-aes --run-at "0 9 * * 1" --start-time "May 8, 2021 5:57:51 PM"
```

To schedule a key rotation everyday at 9 PM, and rotate keys that were created at least a year ago:

```
ksctl scheduler configs create key-rotation --name rotate-old-keys --run-at "0 21 * * *" --metadata-json "{\"createdBefore\": \"-1y\"}"
```

## Replaced Key State

On KeySecure, the state of a rotated key can be set to one of the following: restricted, active, or retired. However, after migrating from KeySecure to CipherTrust Manager, the rotated key state can only be changed to retired.

As a result, performing decryption operations on the previous key version after rotation was not possible in earlier versions of the CipherTrust Manager.

There is a provision to set the state of the previous key version after rotation using an optional parameter, `replaced-key-state`. This parameter allows setting the previous key version's state to either "Deactivated" or "ProtectStop" following key rotation.

The "Deactivated" key state indicates that a key has become inactive on a specific date and time and can no longer be used for any cryptographic operations, whereas the "Protect Stop Date" is not a separate key state but a substate of the "Active" state. When the Protect Stop Date has passed, the key remains active but indicates that protection or support has expired. In this sub-state, encryption and wrapping operations are disallowed, while decryption is still permitted.

The `replaced-key-state` parameter should only be used with the `change_state_after_time` parameter. Also, ensure not to use `replaced_key_state` parameter when `deactivate_replaced_key` is used. The value of the `replaced-key-state` can be `Deactivated` or `ProtectStop` after key rotation.

#### Example Request 1 (`replaced-key-state` is set to `ProtectStop`)

```
ksctl scheduler configs create key-rotation --name "test1" --run-at "0 1 * * *"
--replaced-key-state "ProtectStop" --change-state-after-time 3
```

#### Example Response 1

```
{
 "id": "0c316329-9b81-4da5-8041-74271252552d",
 "uri": "kylo:kylo:scheduler:job_configs:0c316329-9b81-4da5-8041-74271252552d",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2023-04-27T04:37:55.984758701Z",
 "name": "test1",
 "updatedAt": "2023-04-27T04:37:55.984758701Z",
 "description": "",
 "operation": "key_rotation",
 "run_at": "0 1 * * *",
 "run_on": "any",
 "disabled": false,
 "job_config_params": {
 "replaced_key_state": "ProtectStop",
 "change_state_after_time": 3
 }
}
```

#### Note

- The `replaced-key-state` and `change-state-after-time` parameters must be used together.
- Don't use `replaced_key_state` parameter when `deactivate_replaced_key` is used.

The `change_state_after_time` parameter represents the time (number of seconds  $\geq 0$ ) after which the replaced key will change its state to the value defined in the `replaced_key_state` parameter.

#### Example Request 2 (`replaced-key-state` is set to `Deactivated`)

```
ksctl scheduler configs create key-rotation --name "test2" --run-at "0 1 * * *"
--replaced-key-state "Deactivated" --change-state-after-time 3
```

#### Example Response 2

```
{
```

```

 "id": "d0c48426-f5cc-416b-b935-e91b5872db8a",
 "uri": "kylo:kylo:scheduler:job_configs:d0c48426-f5cc-416b-b935-e91b5872db8a",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2023-04-27T04:39:57.848220415Z",
 "name": "test2",
 "updatedAt": "2023-04-27T04:39:57.848220415Z",
 "description": "",
 "operation": "key_rotation",
 "run_at": "0 1 * * *",
 "run_on": "any",
 "disabled": false,
 "job_config_params": {
 "replaced_key_state": "Deactivated",
 "change_state_after_time": 3
 }
 }
}

```

#### Note

The value of `change-state-after-time` cannot be set to negative, that is  $> = 0$ .

## Crypto Operations

The CipherTrust Manager supports the following crypto operations:

### Symmetric Encryption

Symmetric encryption uses AES, TDES, or AES-GCM (with or without authentication data).

A variety of modes and key sizes are supported. Operations can be chained to encrypt a stream of data, split over multiple requests.

| Algorithm | Mode     | Padding (* = default) |
|-----------|----------|-----------------------|
| AES       | GCM      |                       |
| AES       | ECB, CBC | none, PKCS7*          |
| TDES      | CBC      | none, PKCS5*          |
| AES       | CTR      | none                  |

### Asymmetric Encryption

Asymmetric encryption uses RSA keys. A variety of paddings are supported.

| Algorithm | Mode | Padding (* = default) |
|-----------|------|-----------------------|
| RSA       |      | OAEP, PKCS1*          |

## Asymmetric Sign and Verify

| Algorithm |
|-----------|
| SHA1      |
| SHA256    |
| SHA384    |
| SHA512    |

### Note

For HashEdDSA, the input must be a SHA-512 hashed message, encoded in a Base64 format.

## MAC and MAC Verify

| Algorithm |
|-----------|
| SHA1      |
| SHA256    |
| SHA384    |
| SHA512    |

## Format Preserving Encryption

Format-Preserving Encryption is a special type of encryption that keeps the output of the algorithm in the same format and length as the input. This can be important when the result is stored in a database that expects a certain format. A common use case is for Credit Card Numbers, zip codes, telephone numbers, names, addresses etc.

### Note

Using this service/api/product does not confer compliance with any regulation regarding the use, storage and processing of Credit Card data.

Normally FPE algorithms limit the size of the data that can be encrypted to a number based on the size of the alphabet. Selecting the "alphabet" determines two things - how long your input can be - and what characters will be encrypted and which ones will be left in place.

| Sample       | Alphabet     | sample Result  |
|--------------|--------------|----------------|
| 443-555-1055 | digit        | 231-601-4293   |
| 443-555-1055 | alphabet     | 443-555-1055   |
| 443-555-1055 | alphanumeric | ab4-56k-DG3e   |
| 443-555-1055 | printable    | &08yfh210f8\$5 |
| Hello World! | digit        | Hello World!   |
| Hello World! | alphabet     | djheF dkoRE!   |
| Hello World! | alphanumeric | 4JhR6 0e5w2!   |
| Hello World! | printable    | 3J;e6#0e^1&d   |

| Alphabet     | Maximum Size |
|--------------|--------------|
| digit        | 56           |
| alphabet     | 32           |
| alphanumeric | 32           |
| printable    | 28           |

The algorithm handles ASCII chars 0x20 through 0x7E. Apart from ASCII, other characters are supported through unicode character set.

The FPE implementation also supports inputs that are longer than the limits imposed above. However using the API this way is more complicated and additional parameters must be passed.

## Unicode

The Unicode charset must be present on the server and its size should be greater than 10 and less than 256 characters. The number of characters provided in the character set will be treated as cardinality for the input data.

This table provides the cardinality size and the corresponding IV/block-size for the FPE/AES/UNICODE usage. In FPE/AES/UNICODE, the cardinality and IV/block-size is defined in character length.

| Cardinality | IV/Block-size | Cardinality | IV/Block-size | Cardinality | IV/Block-size |
|-------------|---------------|-------------|---------------|-------------|---------------|
| 10          | 56            | 92          | 28            | 174         | 24            |
| 11          | 54            | 93          | 28            | 175         | 24            |
| 12          | 52            | 94          | 28            | 176         | 24            |
| 13          | 50            | 95          | 28            | 177         | 24            |
| 14          | 50            | 96          | 28            | 178         | 24            |
| 15          | 48            | 97          | 28            | 179         | 24            |
| 16          | 48            | 98          | 28            | 180         | 24            |
| 17          | 46            | 99          | 28            | 181         | 24            |
| 18          | 46            | 100         | 28            | 182         | 24            |



| Cardinality | IV/Block-size | Cardinality | IV/Block-size | Cardinality | IV/Block-size |
|-------------|---------------|-------------|---------------|-------------|---------------|
| 19          | 44            | 101         | 28            | 183         | 24            |
| 20          | 44            | 102         | 28            | 184         | 24            |
| 21          | 42            | 103         | 28            | 185         | 24            |
| 22          | 42            | 104         | 28            | 186         | 24            |
| 23          | 42            | 105         | 28            | 187         | 24            |
| 24          | 40            | 106         | 28            | 188         | 24            |
| 25          | 40            | 107         | 28            | 189         | 24            |
| 26          | 40            | 108         | 28            | 190         | 24            |
| 27          | 40            | 109         | 28            | 191         | 24            |
| 28          | 38            | 110         | 28            | 192         | 24            |
| 29          | 38            | 111         | 28            | 193         | 24            |
| 30          | 38            | 112         | 28            | 194         | 24            |
| 31          | 38            | 113         | 28            | 195         | 24            |
| 32          | 38            | 114         | 28            | 196         | 24            |
| 33          | 38            | 115         | 28            | 197         | 24            |
| 34          | 36            | 116         | 26            | 198         | 24            |
| 35          | 36            | 117         | 26            | 199         | 24            |
| 36          | 36            | 118         | 26            | 200         | 24            |
| 37          | 36            | 119         | 26            | 201         | 24            |
| 38          | 36            | 120         | 26            | 202         | 24            |
| 39          | 36            | 121         | 26            | 203         | 24            |
| 40          | 36            | 122         | 26            | 204         | 24            |
| 41          | 34            | 123         | 26            | 205         | 24            |
| 42          | 34            | 124         | 26            | 206         | 24            |
| 43          | 34            | 125         | 26            | 207         | 24            |
| 44          | 34            | 126         | 26            | 208         | 24            |
| 45          | 34            | 127         | 26            | 209         | 24            |
| 46          | 34            | 128         | 26            | 210         | 24            |
| 47          | 34            | 129         | 26            | 211         | 24            |
| 48          | 34            | 130         | 26            | 212         | 24            |
| 49          | 34            | 131         | 26            | 213         | 24            |
| 50          | 34            | 132         | 26            | 214         | 24            |
| 51          | 32            | 133         | 26            | 215         | 24            |

| Cardinality | IV/Block-size | Cardinality | IV/Block-size | Cardinality | IV/Block-size |
|-------------|---------------|-------------|---------------|-------------|---------------|
| 52          | 32            | 134         | 26            | 216         | 24            |
| 53          | 32            | 135         | 26            | 217         | 24            |
| 54          | 32            | 136         | 26            | 218         | 24            |
| 55          | 32            | 137         | 26            | 219         | 24            |
| 56          | 32            | 138         | 26            | 220         | 24            |
| 57          | 32            | 139         | 26            | 221         | 24            |
| 58          | 32            | 140         | 26            | 222         | 24            |
| 59          | 32            | 141         | 26            | 223         | 24            |
| 60          | 32            | 142         | 26            | 224         | 24            |
| 61          | 32            | 143         | 26            | 225         | 24            |
| 62          | 32            | 144         | 26            | 226         | 24            |
| 63          | 32            | 145         | 26            | 227         | 24            |
| 64          | 32            | 146         | 26            | 228         | 24            |
| 65          | 30            | 147         | 26            | 229         | 24            |
| 66          | 30            | 148         | 26            | 230         | 24            |
| 67          | 30            | 149         | 26            | 231         | 24            |
| 68          | 30            | 150         | 26            | 232         | 24            |
| 69          | 30            | 151         | 26            | 233         | 24            |
| 70          | 30            | 152         | 26            | 234         | 24            |
| 71          | 30            | 153         | 26            | 235         | 24            |
| 72          | 30            | 154         | 26            | 236         | 24            |
| 73          | 30            | 155         | 26            | 237         | 24            |
| 74          | 30            | 156         | 26            | 238         | 24            |
| 75          | 30            | 157         | 26            | 239         | 24            |
| 76          | 30            | 158         | 26            | 240         | 24            |
| 77          | 30            | 159         | 26            | 241         | 24            |
| 78          | 30            | 160         | 26            | 242         | 24            |
| 79          | 30            | 161         | 26            | 243         | 24            |
| 80          | 30            | 162         | 26            | 244         | 24            |
| 81          | 30            | 163         | 26            | 245         | 24            |
| 82          | 30            | 164         | 26            | 246         | 24            |
| 83          | 30            | 165         | 26            | 247         | 24            |
| 84          | 30            | 166         | 26            | 248         | 24            |

| Cardinality | IV/Block-size | Cardinality | IV/Block-size | Cardinality | IV/Block-size |
|-------------|---------------|-------------|---------------|-------------|---------------|
| 85          | 28            | 167         | 26            | 249         | 24            |
| 86          | 28            | 168         | 24            | 250         | 24            |
| 87          | 28            | 169         | 24            | 251         | 24            |
| 88          | 28            | 170         | 24            | 252         | 24            |
| 89          | 28            | 171         | 24            | 253         | 24            |
| 90          | 28            | 172         | 24            | 254         | 24            |
| 91          | 28            | 173         | 24            | 255         | 24            |

## MKEK Rotation

The CipherTrust Manager allows you to rotate the Master Key Encryption Key (MKEK). Key rotation protects the key material from malicious interceptions. For CipherTrust Managers using a Hardware Security Module (HSM) as a root of trust, you can also [rotate the Root of Trust \(RoT\) key](#) which is above the MKEK on the key hierarchy.

### Note

Only Admin in the root domain is allowed to use MKEK APIs.

## Managing MKEK using ksctl

The following operations can be performed:

- Rotate MKEK
- Get details of MKEK
- List all MKEKs

## Rotating MKEK

The rotation of a MKEK generates a new MKEK in a clustered as well as in a non-clustered setup. The secrets also get migrated to the new MKEK. It allows you to provide a custom name to the newly generated MKEK. To rotate an MKEK, run:

### Syntax

```
ksctl mkeys rotate --name <CustomName>
```

### Example Request

```
ksctl mkeys rotate --name test
```

### Example Response

```
{
 "id": "9fb3264f-33ad-4f2b-ba5f-b3e841543bd9",
 "name": "test",
 "is_default": true,
 "created_at": "2021-03-09T07:21:06.541873461Z",
 "sealer_name": "none"
}
```

#### Note

The `sealer_name` parameter value signifies whether the HSM is configured for the given instance of CipherTrust Manager. Values are:

- `none` - HSM is not configured.
- `rapido` - HSM is configured.

## Getting Details of MKEK

To get details of an MKEK, run:

### Syntax

```
ksctl mkeys get -i <mkek_id>
```

### Example Request

```
ksctl mkeys get --id 9fb3264f-33ad-4f2b-ba5f-b3e841543bd9
```

### Example Response

```
{
 "id": "9fb3264f-33ad-4f2b-ba5f-b3e841543bd9",
 "name": "test",
 "is_default": true,
 "created_at": "2021-03-09T07:21:06.541873461Z",
 "sealer_name": "none"
}
```

---

## Getting List of MKEKs

To get a list of MKEKs, run the following command. By default it returns only one MKEK on the system.

### Syntax

```
ksctl mkeks list
```

### Example Request

```
ksctl mkeks list
```

### Example Response

```
{
 "total": 1,
 "info": [
 {
 "id": "658027ff-9780-4e9c-9041-cec99516a162",
 "name": "mkek_sample_name",
 "is_default": true,
 "created_at": "2021-03-09T07:16:55.864674685Z",
 "sealer_name": "none"
 }
]
}
```

If you fetch the list of MKEKs during rotation process, it returns more than one MKEK on the system. To get a list of all MKEKs, run:

### Syntax

```
ksctl mkeks list --all
```

### Example Request

```
ksctl mkeks list --all
```

### Example Response

```
{
 "total": 2,
 "info": [
 {
```

```

 "id": "632ac9aa-f927-4704-b9af-859b5e42fc8b",
 "name": "632ac9aa-f927-4704-b9af-859b5e42fc8b",
 "created_at": "2021-03-08T10:40:34.038989394Z",
 "sealer_name": "none"
 },
 {
 "id": "658027ff-9780-4e9c-9041-cec99516a162",
 "name": "mkek_sample_name",
 "is_default": true,
 "created_at": "2021-03-09T07:16:55.864674685Z",
 "sealer_name": "none"
 }
]
}

```

In the above example response, you can see the two (2) MKEKs being listed. The MKEK with `"is_default": true` parameter is the newly generated default MKEK. The other MKEK is the MKEK before rotation, which stays temporarily until the rotation process is complete.

## Certificate Authority

A Certificate Authority (CA) acts as the initially trusted shared entity between peers and issues signed certificates to make it possible for each party to trust the other. The CA issues and installs digital certificates and issues certificate signing requests (CSR).

A certificate generally acts as the identity of a server or client and certificate authorities can be used to issue server and client certificates for setting up trusted communication channels to the CipherTrust Manager system.

CipherTrust Manager distinguishes between local CAs and external CAs. A local CA can issue signed certificates since the private signing key is stored inside the CipherTrust Manager system. An external CA does not store the private key. Instead an external CA is used as a trusted entity for various interfaces and services inside the system. In this case certificates are issued externally. It is fine to have a mix of both.

The first time a CipherTrust Manager is started, a new local `CipherTrust Manager Root CA` is automatically generated. This CA is used to issue initial server certificates for the interfaces available in the system.

An easy way to inspect the certificate chain is to view the certificates in your browser when you connect to the web interface. All interfaces and services will by default trust this CA, meaning that for interfaces that support client authentication, a client certificate, issued from this initial `CipherTrust Manager Root CA`, will automatically be trusted by the system. If preferred, it is possible to create new local CAs and/or external CAs and use these instead for the internal interfaces and services.

## Creating a Local CA in the Web Console UI

To create a new Local CA, you first generate a CSR, which is then listed a pending CA. The CSR can then be self-signed, signed by another Local CA, or signed by an External CA. Self-signed certificates become Root CAs, and certificates signed by other CAs become intermediate CAs.

1. Login as an Application Administrator in the Admins `group`, such as `admin`.
2. Navigate to **CA > Local**.
3. Click **+ Add Local CA**.

- 
4. Provide a **Common Name**.
  5. Examine the **Algorithm** and **Size** that determine the private key, and use the dropdown to change values if desired. The available values are:
    - ECDSA 384 (default)
    - ECDSA 256
    - ECDSA 521
    - RSA 1024

**Caution**

We strongly discourage using RSA 1024, as this algorithm and key length is no longer considered secure enough for this purpose, as outlined in [NIST Special Publication 800-131A Revision 2](#). This key size is only present for backward compatibility.

- RSA 2048
  - RSA 3072
  - RSA 4096
6. Provide the following optional values:
    - Display Name
    - DNS Names (comma separated)
    - IP Addresses (comma separated)
    - Email Addresses (comma separated)
    - Name (comma separated)
  7. Click **Add Local CA**.

The CSR is visible in the **Pending CAs** table. The CipherTrust Manager internally stores the private key associated with the CSR.

8. Sign the CSR to make the new Local CA active. You can self-sign the certificate, sign with another local CA, or sign with an external CA. Self-signed certificates become Root CAs, and certificates signed by other CAs become intermediate CAs.

---

## Self-sign

- a. Find the desired CSR in the **Pending CAs** table.
- b. Click the corresponding overflow icon (  ) and select **Self-sign**.

A confirmation pop-up dialog displays.

- c. Select a **Duration(days)** before the CA will expire. The default value is 365.

### Caution

Whenever a CA expires, all certificates it has issued also expire. This means that if you create a Local CA and use it to issue multiple client and server certificates for access to CipherTrust Manager, when the Local CA is at or near expiry, your organization will need to replace the certificate chain in multiple interfaces and clients to ensure continued communication. Consider how frequently your organization would like to perform this maintenance when setting a CA duration.

- d. Click **Save**.

The new Local CA is now visible in the **Local Certificate Authorities** table.

## Sign with an Existing Local CA

- a. Find the desired CSR in the **Pending CAs** table.
- b. Click the corresponding overflow icon (  ) and select either **Copy CSR** to copy the CSR contents to the clipboard, or **Download CSR** to download the CSR as a file.
- c. In the **Local Certificate Authorities** table, find the existing local CA you wish to sign the CSR. Click the local CA's name.

The certificates issued by the existing local CA are displayed.

- d. Click **Upload CSR**.

A pop-up dialog displays to upload and sign the certificate.

- e. Enter the **Display name** you would like for the new CA.
- f. In the **CSR** text field, paste in the text of the CSR.
- g. In the **Certificate Purpose** drop-down, select **CA**.
- h. Select a **Duration in days** before the new CA will expire.



### Caution

Whenever a CA expires, all certificates it has issued also expire. This means that if you create a Local CA and use it to issue multiple client and server certificates for access to CipherTrust Manager, when the Local CA is at or near expiry, your organization will need to replace the certificate chain in multiple interfaces and clients to ensure continued communication. Consider how frequently your organization would like to perform this maintenance when setting a CA duration.

- i. Click **Issue Certificate** to issue the CA certificate.
- j. Find the new CA certificate in the **Certificates issued by** table. Click the corresponding overflow icon (  ) and select either **Copy** to copy the CA certificate contents to the clipboard, or **Download** to download the CA certificate as a file.
- k. Navigate back to **CA > Local** and find the desired CSR in the **Pending CAs** table. Click the corresponding overflow icon (  ) and select **Install**.

A pop-up dialog displays to input the signed CA certificate.

- l. For **Parent CA Type**, select **Local CA** from the drop-down.
- m. In the **Select Local Parent CA** drop-down, select the local CA which signed the CA certificate.
- n. In the **Certificate** text field, paste the text of the signed CA certificate.
- o. Click **Save**.

The new local CA is now created and visible in the **Local Certificate Authorities** table.

## Sign with an External CA

- a. Make sure the desired external CA is known to the CipherTrust Manager. Navigate to **CA > External** to see if the external CA is on the page. If it is not, [add the External CA](#).
- b. Navigate back to **CA > Local** and find the desired CSR in the **Pending CAs** table. Click the corresponding overflow icon (  ) and select either **Copy CSR** to copy the CSR contents to the clipboard, or **Download CSR** to download the CSR as a file.
- c. Outside CipherTrust Manager, have the External CA sign the certificate. Retain the signed CA certificate.

### Caution

Whenever a CA expires, all certificates it has issued also expire. This means that if you create a

Local CA and use it to issue multiple client and server certificates for access to CipherTrust Manager, when the Local CA is at or near expiry, your organization will need to replace the certificate chain in multiple interfaces and clients to ensure continued communication. Consider how frequently your organization would like to perform this maintenance when setting a CA duration during CSR signing.

- d. Navigate back to **CA > Local** and find the desired CSR in the **Pending CAs** table. Click the corresponding overflow icon (  ) and select **Install**.

A pop-up dialog displays to input the signed CA certificate.

- e. For **Parent CA Type**, select **External CA** from the drop-down.
- f. In the **Select External Parent CA** drop-down, select the external CA which signed the CA certificate.
- g. In the **Certificate** text field, paste the text of the signed CA certificate.
- h. Click **Save**.

The new local CA is now created and visible in the **Local Certificate Authorities** table.

## Creating a Local CA in ksctl CLI

When creating a new local CA, it remains pending until signed:

```
ksctl ca locals create --cn "Test CA" --csr-outfile csrfile
```

This returns a CSR. The key remains with the CipherTrust Manager. This CSR can be self-signed or signed by another local CA or an external CA.

To just self-sign the CA with a one year duration, use the id returned in the call above:

```
ksctl ca locals self-sign --id 3593c53b-fbeb-4edb-b84d-c85526ae2f83 -x 365
```

## Issuing Certificate Signed by Local CA

CipherTrust Manager's local CA can issue certificates for external clients, servers and CAs.

To create a certificate, you must have a CSR available with you. CSR can be created within CipherTrust Manager or obtained externally.

**Tip**

For the highest level of security, it is recommended to get CSR external to the CipherTrust Manager, so that the private key is never exposed. However, you can also **Download CSR** from the pending CA on the CipherTrust Manager GUI as this will not include the private key. This CSR can only be used to issue certificate for external CAs.

## Issue Client Certificate Using ksctl CLI

To issue certificate for clients you must have a CSR available.

Use the following ksctl command to create a CSR and private key within CipherTrust Manager:

```
ksctl ca csr -cn "My Client" --csr-outfile csrfile --key-outfile keyfile
```

Additionally, you can use the `--dns` option to add Subject Alternative Names (SAN) values. If multiple values are specified, separate them with comma (optional).

```
ksctl ca csr -cn "My Client" --csr-outfile csrfile --key-outfile keyfile --dns "thalesgroup.com,thalesgroup2.com" --ips 1.1.1.1
```

**Note**

You cannot add SAN in a default web server certificate post-deployment.

Instead, perform the following steps:

1. Generate a new CSR with the SAN fields. Refer to the above command.
2. Issue a new certificate using the generated CSR. Refer to the below command.
3. Update the web interface by uploading this issued certificate and restart the system.

Once you have a CSR, use the following ksctl command to issue a certificate:

```
ksctl ca locals certs issue --ca-id 3593c53b-fbeb-4edb-b84d-c85526ae2f83 --csr-in file csrfile -x 700 -o client
```

The example above issues certificate for client purpose. Change the argument for `-o` flag to `server` or `cas` to issue certificates for server or external CAs purpose. For complete list of flags (options) available for this command, refer to CipherTrust Manager CLI guide.

**Note**

- When signing the local CA and certificate, duration is set a day (24 hours) before the current date; therefore the `notBefore` flag also reflects the same date. This is done to handle the multiple time zone

differences.

- The certificate duration shouldn't be more than the CA duration. However, if the certificate duration exceeds the CA duration, the certificate duration is automatically set to CA's duration.
- It is also possible to create the CSR and the private key using any other software, as this API is stateless and doesn't store anything within CipherTrust Manager.

## Issue certificate using GUI

This procedure describes how to issue a client or server certificate.

### Note

The following steps can be used to issue both **client** and **server** certificates using the local CA on CipherTrust Manager. Certificates generated using this method return the CSR and private key. This private key is not stored on the CipherTrust Manager. These certificates can be used with any client or server outside of CipherTrust Manager.

The server certificate issued using this method can also be used on any interface (NAE, KMIP, Web) on the CipherTrust Manager.

Alternatively, for server certificates, you can [generate CSRs directly through the interface](#). When using this method, the private key is stored on the CipherTrust Manager.

1. Log on to the CipherTrust Manager GUI.
2. Navigate to **CA > CSR Generator**.
3. Select the **Generic CSR** radio button and fill in the required details. The **Common Name** field is mandatory.
4. Click **Generate CSR and download Private Key**.  
It generates a CSR and downloads the private key. Make sure to save the private key.
5. Click **Download CSR** to download the generated CSR. Make sure to save the CSR.
6. Navigate to **CA > Local**. The list of available Local CAs is displayed.
7. Click the name of the any local CA displayed on the page. The Certificate issued screen by that CA is displayed.
8. Click **Upload CSR**.
9. Enter the **Display Name**.

10. Provide the **CSR** that you downloaded in the previous step.
11. Select the **Certificate Purpose** according to your requirement: **client**, **server**, or **ca** for issuing a certificate.
12. Click **Issue Certificate**.
13. Click the ellipsis ( **⋮** ) icon corresponding to the newly generated certificate and select **download**.

Save the downloaded certificate.

## Certificate expiration check

The CipherTrust Manager inspects the expiration date of the following types of certificates every day, at a preset system time to log the record:

- Local CA certificates available on CipherTrust Manager
- Certificate issued by Local CA and available on CipherTrust Manager
- External CA certificates uploaded to CipherTrust Manager

The CipherTrust Manager then creates list a of certificates based on their expiration date:

- Certificates whose expiration dates are within 91 days

This list is logged in the **Records** section once every week

- Certificates whose expiration dates are within 7 days

This list is logged in the **Records** section once every day

- Certificates that are already expired

This list is logged in the **Records** section once every day

### Note

Interface setting such as NAE allows you to upload certificates directly. However, the CipherTrust Manager doesn't allow to upload expired or revoked certificates.

You can also create alarm triggers for these records. For more details, go to [Creating Alarm Trigger for Certificate Expiration](#).

## Revoke/resume Certificate Signed by the Local CA

The CipherTrust Manager allows you to revoke/resume the client/server certificates signed by the local CA. You can also revoke/resume the certificates of intermediate CAs.

In addition, you can:

- Publish and maintain the Certificate Revocation List (CRL) for the certificates revoked by the local CA.
- Migrate the revocation status of the certificates from KeySecure Classic to CipherTrust Manager.
- Resume the revoked certificates.

#### Note

- Revocation of local CA is not supported, only local CA signed certificates can be revoked.
- You can only resume the certificates revoked with the reason "certificateHold" or "Certificate Hold"

## Valid Revocation Reasons

You must provide a reason to revoke a certificate. Allowed revocation reasons are defined by RFC5280.

| Reason Value in Web Console GUI | Reason Value in ksctl CLI, REST API |
|---------------------------------|-------------------------------------|
| AA Compromise                   | aAcompromise                        |
| Affiliation Changed             | affiliationChanged                  |
| CA Compromise                   | cAcompromise                        |
| Certificate Hold                | certificateHold                     |
| Cessation of Operation          | cessationOfOperation                |
| Key Compromise                  | keyCompromise                       |
| Privilege Withdrawn             | privilegeWithdrawn                  |
| Remove from CRL                 | removeFromCRL                       |
| Superseded                      | superseded                          |
| Unspecified                     | unspecified                         |

## Revoke a Certificate Signed by a Local CA in Web Console UI

1. Login to the root [domain](#) as an Application Administrator in the Admins [group](#), such as `admin`.
2. Navigate to **CA > Local**.
3. In the **Local Certificate Authorities** table, find the Local CA that issued the certificate that you wish to revoke and click on its name.

The certificates issued by the existing local CA are displayed.

4. Find the certificate to revoke. Click the corresponding overflow icon (  ) and select **Revoke**.

A pop-up dialog to complete revocation displays.

---

5. Select a reason for the revocation from the drop-down. Valid reasons are:

- AA Compromise
- Affiliation Changed
- CA Compromise
- Certificate Hold
- Cessation of Operation
- Key Compromise
- Privilege Withdrawn
- Remove from CRL
- Superseded
- Unspecified

6. Click **Revoke** to confirm.

The **State** of the certificate in the table is a red x, to indicate it is revoked.

## Resume a Certificate Signed by a Local CA in Web Console UI

Certificates revoked with the reason "certificateHold" or "Certificate Hold" can be resumed.

1. Login to the root domain as an Application Administrator in the Admins group, such as `admin`.
2. Navigate to **CA > Local**.
3. In the **Local Certificate Authorities** table, find the Local CA that issued the certificate that you wish to revoke and click on its name.

The certificates issued by the existing local CA are displayed.

4. Find the certificate to resume. Click the corresponding overflow icon (  ) and select **Resume**.

### Tip

Revoked certificates have a **State** of a red x icon in the table.

The **State** of the certificate in the table changes to a green checkmark, to indicate it is resumed.





## Syntax

### Example Request

### Example Response

```
{
 "id": "2526ebf8-8ac2-4e3b-8c2c-9752c3da536d",
 "uri": "kylo:kylo:naboo:certs:2526ebf8-8ac2-4e3b-8c2c-9752c3da536d",
 "account": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2021-05-10T15:13:08.429514Z",
 "updatedAt": "2021-05-12T06:44:54.401005002Z",
 "cert": "-----BEGIN CERTIFICATE-----\nMIIIEuDCCAqCgAwIBAgIQOivQNtvylbsD+ZtTiktSbjANBqkqhkiG9w0BAQsFADBAlmMQswCQQYDVQQGEWJVUzELMAkGA1UECBMCTUxEdDAOBgNVBACTB0JlbnGNhbXAxEDAO\nnBgNVBAoTB0dlbWFSdG8xGjAYBgNVBAMTEUtleVnlY3VyZSBSb290IENBMXB4XDITix\nnMDUwOTE1MTMwOFoXDTIzMduwOTE1MTMwOFowJTEOMAwGA1UEAxFMYWRtaW4xeZAR\nnBgoJkiaJk/IsZAEBEwMxmMjMwggeEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKaoIB\nnAQDf0/15sDKlmZ940mc3YAmpdeHmAPlf6kDZgtqpun9ftXji65WIHywZ5VN/5YYVD\nnREdbS96kAdNMNyc8AsOE0lbgirxaW2HFOzVcdfUYyh8FnQWq4kAcGBdl19gvdEm6\nnoZOax6XLKZq3REfvFXjPg3YkhOvmaIF/9WFoVafCplpgpiB3kiijd3mlZUHP+uxW\nnkfJ6ddxMs3Qeq3gltfmpnjohY433rzHzCFr/W5wufRKZXWmlu2ObwTKJsixJbcRJR1\n\n93+XVELt6r7Umryczjmi3RIMkJ0WC+KpkL0ZetYtXL/7IyKRkzlqAwKI4mpyJjAS\n\n/n/3yQgJKCdSBz80BzmnbDevQ9AgMBAAEjga4wgaswDgYDVDR0PAQH/BAQDAgOI
MBMG\nnA1UdJQQMMMAoGCCsGAQUFBwMCMAwGA1UdEwEB/wQCMAAwHwYDVDR0jBBgwFoAUl4YP\n\nnF8V39/lnMb8i5iDOptXjQ4owVQYDVDR0fBE4wTDBKoEigRoZEaHR0cDovL2tleXNl\n\nnY3VyZS5sb2Nhbc9jcmxzL2Y4YWFiZDRmLTlY0NTktNGNiMClhMjZhLWRmZDg4MTI5\n\nnYmQlZS5jcmwwDQYJKoZIhvcNAQELBQADggIBAAYsYXivy9vD+WMqs4ceC+W3O8Tx\n\nnIW/jaChFWZKXR4fk01n1Mh02OT67wIKqQXUoTkGp9U7vmNMd/RKrj1NS191Eh8sm\n\nnHxy7/bvcSDXajw2LpsmIRaWeqYgO0qOtluMQMMnSBiLbdgSAXKEAjRMQqvQfzqUV\n\nneTSPWaWzyfBnfhSEfu0s46Xs6lgWTfwclvb40Xk7HKFTNUP/xPIflLht4H9J3Bx\n\nnyrWz5bJYLz6Cx95/gXsQptccmYik+WGy7IJofvNJJDugclt6SeVG2aEl8fNiuS5a\n\nnp9O6ThUcM3MgHcL0cOlqm9+jzs5j8PUWBJ+7lsDS17Y+uFvHEJN8XGXQLhFf3p/4\n\nnvNgyMAMB9uvC5rbqEsCKUGpxkNa0sm0WflVoIQlh2ku01yqtG8krma9qr4zy+bML\n\nnO6Zk37Vn1/8puJGYWHIPhjX6e+/wlRIMufyqKg7M/OhlqOS6eOpaX13tXxYNnaVm\n\nngN2mkFvmN3W6sMdtCKifRNeTcuF5R7ZRWXKqHp00Y6N2Tk2FyZjgWAxUtg7VnLPW\n\nnRfuQBQ/Jud7zVDWxtftv6nmrV1nlqErPPDnRt3D49AD5lj4+JhdzKz47F094T++8\n\nnrauAODq6i+FZe/05RwSCB1fqWJ8ja9gwAWaBVxfQpIDIY3KFtC2tzhjUUOi++d\n\nnp6WaJc1NqTcWns8H\n\nn-----END CERTIFICATE-----\n",
 "ca": "kylo:kylo:naboo:localca:f8aabdf-f6459-4cb0-a26a-dfd88129bd5e",
 "revoked_at": "0001-01-01T00:00:00Z",
 "state": "active",
 "sha1Fingerprint": "C5BF83559D11C81ED84D8F7CC15094DA365D775D",
```

```

"sha256Fingerprint": "11C91396CB62BA28EDAE07E79681C25276C7B93DBA033DF08DEB13D9FDB
E353F",
"sha512Fingerprint": "28C66DAA4775F37B4294964D45BCC9DB8BAD26C24EB4D9370DE6AB3BEF3
2C4D156D5FC45D98F6201F97F0D5DEBB43BBC6E24A1FE3743F64E5D1F0B9B2B93FCEE",
"serialNumber": "77322715608031240047279766081599394414",
"subject": "/CN=admin",
"issuer": "/C=US/ST=MD/L=Belcamp/O=Gemalto/CN=KeySecure Root CA",
"notBefore": "2021-05-09T15:13:08Z",
"notAfter": "2023-05-09T15:13:08Z"
}

```

## Certificate revocation list (CRL)

CRL is a list of certificates that have been revoked by the CA before their scheduled expiration date and should no longer be trusted.

- On CipherTrust Manager, when you create a local root CA, one CRL file is created for the CA with the name `<local_ca_id>.crl`.
- It is mandatory to pass the valid DNS names while creating local root CA to ensure that the CRL URL is accessible. If DNS names are not provided, the default name `keysecure.local` is used.
- When a new certificate is issued by the local root ca, the certificate contains the URL of the CRL. You can check the URL of the CRL in the certificate under CRL Distribution Points by decoding it and can download the CRL file from the URL.
- The CRL can only be decoded and accessed through the command line.

**Example:** OpenSSL command to decode and check the URL of the CRL:

```

openssl x509 -in intermediate/certs/bob2@example.com.cert.pem -noout -text
X509v3 CRL Distribution Points:
Full Name:
URI:http://keysecure.local/1d073d93-b156-49ee-8533-e338953fc6d9.crl

```

### Steps to download and decode the CRL

- Download the CRL from the URL and decode it to check the revocation list.

```
openssl crl -in 1d073d93-b156-49ee-8533-e338953fc6d9.crl.crl -noout -text
```

- After decode, you can see the list of revoked certificates. It contains serial numbers of the revoked certificates. The serial numbers are in hex format; whereas CipherTrust Manager stores the serial numbers in decimal format.

Therefore, to convert the serial number to decimal format, run:

```
echo "ibase=16; <hex_serial_number>" | bc
```

## Renewing Local CA certificate

The CipherTrust Manager allows you to renew the local CA certificates in the certificate chain when they are about to expire.

When a local CA in the certificate chain expires, the entire certificate chain validation fails.

In such scenarios, the user has to create a new CA and renew the client/server certificates or both using the new CA.

### When to renew

To avoid any service downtime, it is recommended to prepare the new Local CA certificate chain in advance before the CA expiration. During the overlapping period, the new server certificate chain as described in the below [section](#) should be downloaded and added to the trust store in the client's setup.

### How to renew in the web console UI

The CA certificate chain renewal process contains the following steps:

1. Create a new local CA using the same attributes as an existing CA.
2. Renew server certificates issued by the original local CA.
3. Renew KMIP client certificates issued by the original local CA.

#### Note

Only KMIP client certificates can be renewed through the web console UI. Other client types [must renew their certificate certificates through the ksctl CLI](#) or REST API.

### web console UI: create a new local CA using the same attributes as existing CA

1. Login as an Application Administrator in the Admins [group](#), such as `admin`.
2. Navigate to **CA > Local**.
3. In the **Local Certificate Authorities** table, find the local CA to copy attributes from. Click the corresponding overflow icon (  ) and select **Renew**.

A CSR with the same attributes as the local CA is present in the **Pending CAs** table.

4. Sign the CSR to make the new Local CA active. You can self-sign the certificate, sign with another local CA, or sign with an external CA. Self-signed certificates become Root CAs, and certificates signed by other CAs become intermediate CAs.

---

## Self-sign

- a. Find the desired CSR in the **Pending CAs** table.
- b. Click the corresponding overflow icon (  ) and select **Self-sign**.

A confirmation pop-up dialog displays.

- c. Select a **Duration(days)** before the CA will expire. The default value is 365.

### Caution

Whenever a CA expires, all certificates it has issued also expire. This means that if you create a Local CA and use it to issue multiple client and server certificates for access to CipherTrust Manager, when the Local CA is at or near expiry, your organization will need to replace the certificate chain in multiple interfaces and clients to ensure continued communication. Consider how frequently your organization would like to perform this maintenance when setting a CA duration.

- d. Click **Save**.

The new Local CA is now visible in the **Local Certificate Authorities** table.

## Sign with an Existing Local CA

- a. Find the desired CSR in the **Pending CAs** table.
- b. Click the corresponding overflow icon (  ) and select either **Copy CSR** to copy the CSR contents to the clipboard, or **Download CSR** to download the CSR as a file.
- c. In the **Local Certificate Authorities** table, find the existing local CA you wish to sign the CSR. Click the local CA's name.

The certificates issued by the existing local CA are displayed.

- d. Click **Upload CSR**.

A pop-up dialog displays to upload and sign the certificate.

- e. Enter the **Display name** you would like for the new CA.
- f. In the **CSR** text field, paste in the text of the CSR.
- g. In the **Certificate Purpose** drop-down, select **CA**.
- h. Select a **Duration in days** before the new CA will expire.

### Caution

Whenever a CA expires, all certificates it has issued also expire. This means that if you create a Local CA and use it to issue multiple client and server certificates for access to CipherTrust Manager, when the Local CA is at or near expiry, your organization will need to replace the certificate chain in multiple interfaces and clients to ensure continued communication. Consider how frequently your organization would like to perform this maintenance when setting a CA duration.

- i. Click **Issue Certificate** to issue the CA certificate.
- j. Find the new CA certificate in the **Certificates issued by** table. Click the corresponding overflow icon (  ) and select either **Copy** to copy the CA certificate contents to the clipboard, or **Download** to download the CA certificate as a file.
- k. Navigate back to **CA > Local** and find the desired CSR in the **Pending CAs** table. Click the corresponding overflow icon (  ) and select **Install**.

A pop-up dialog displays to input the signed CA certificate.

- l. For **Parent CA Type**, select **Local CA** from the drop-down.
- m. In the **Select Local Parent CA** drop-down, select the local CA which signed the CA certificate.
- n. In the **Certificate** text field, paste the text of the signed CA certificate.
- o. Click **Save**.

The new local CA is now created and visible in the **Local Certificate Authorities** table.

## Sign with an External CA

- a. Make sure the desired external CA is known to the CipherTrust Manager. Navigate to **CA > External** to see if the external CA is on the page. If it is not, [add the External CA](#).
- b. Navigate back to **CA > Local** and find the desired CSR in the **Pending CAs** table. Click the corresponding overflow icon (  ) and select either **Copy CSR** to copy the CSR contents to the clipboard, or **Download CSR** to download the CSR as a file.
- c. Outside CipherTrust Manager, have the External CA sign the certificate. Retain the signed CA certificate.

### Caution

Whenever a CA expires, all certificates it has issued also expire. This means that if you create a

Local CA and use it to issue multiple client and server certificates for access to CipherTrust Manager, when the Local CA is at or near expiry, your organization will need to replace the certificate chain in multiple interfaces and clients to ensure continued communication. Consider how frequently your organization would like to perform this maintenance when setting a CA duration during CSR signing.

- d. Navigate back to **CA > Local** and find the desired CSR in the **Pending CAs** table. Click the corresponding overflow icon (  ) and select **Install**.

A pop-up dialog displays to input the signed CA certificate.

- e. For **Parent CA Type**, select **External CA** from the drop-down.
- f. In the **Select External Parent CA** drop-down, select the external CA which signed the CA certificate.
- g. In the **Certificate** text field, paste the text of the signed CA certificate.
- h. Click **Save**.

The new local CA is now created and visible in the **Local Certificate Authorities** table.

5. Take a note of the local CA's name for renewing certificates.

## web console UI: renew server certificate using new local CA

To renew the server certificate using the new local CA, perform the following steps:

1. [Issue a new server certificate from the new CA.](#)
2. Create a certificate chain and combine them into a single PEM or PKCS12 file. The file should include the following in the indicated order:
  - a. Server certificate
  - b. Any intermediate certificate authorities that sign the server certificate. Start with the intermediate CA that issued the server certificate. Next, add the issuer of the intermediate CA, if any. Continue adding any intermediate CAs higher up the hierarchy, each time adding the issuer of the last certificate you uploaded.
  - c. The root CA that signs any intermediate CAs. If there are no intermediate CAs, add the root CA that issued the server certificate.
  - d. The server certificate's private key.

**Note**

- In the PEM format, the server certificate's private key should be included first, and in the PKCS12 format, it should be included last.
- It is optional to include the private key in the PEM file if the user has previously generated a CSR. Creating a CSR allows the user to generate the key pair on the CipherTrust Manager and prevents exposing the private key outside of the CipherTrust Manager.

3. Upload the certificate chain to the CipherTrust Manager [interface](#) associated with the clients.
  - a. Login to the root [domain](#) as an Application Administrator in the Admins [group](#), such as [admin](#).
  - b. Navigate to **Admin Settings > Interfaces**.
  - c. In the table, find the interface associated with relevant clients.

Most clients authenticate to the [web](#) interface. NAE clients authenticate to the original [nae](#) interface, or additional NAE interfaces created later. KMIP clients authenticate to the original [kmip](#) interface, or additional KMIP interfaces created later.

- d. [Upload the renewal certificate](#).
4. For clients, the chain uploaded in the previous step is available to [download](#).
  5. Add the downloaded certificate to the client trust store.

Now, the clients can use this new chain to setup their environment.

6. (Optional) After the clients have been updated with the new certificate chain, [apply the new certificate chain](#) on the CipherTrust Manager. This immediately replaces the existing server certificates with the newly generated server certificates.

If you don't immediately apply the new certificate chain manually on the CipherTrust Manager, the auto-apply option is triggered when the CA expires.

## web console UI: renew KMIP client certificate using new local CA

**Note**

Only KMIP client certificates can be renewed through the web console UI. Other client types [must renew their certificate certificates through the ksctl CLI or REST API](#).

1. Generate a Certificate Signing Request (CSR) for a new client certificate. For the highest level of security, it is usually best for the CSR to be generated externally to CipherTrust Manager so the private key is never exposed. However, an internal CSR utility is available in **CA > CSR Generator**. The new CSR must have the same subject name as the existing client certificate.
2. Login to the root **domain** as an Application Administrator in the Client Admins **group**, such as `admin`.
3. Update the client profile associated with the KMIP client to use the new local CA.
  - a. Navigate to **Access Management > Client Profiles**.
  - b. Find the name of the client profile. Click the corresponding overflow icon (  ) and select **View/Edit**.
  - c. In the **Select Local CA** dropdown, select the ID of the new local CA.
  - d. Click **Update**.
4. Navigate to **Access Management > Client Hub**.
5. Find the name of the applicable KMIP client in the table. Click the corresponding overflow icon (  ) and select **Renew Certificate**.

A dialog **Renew KMIP Client Certificate** displays.

6. Provide details to re-issue the client certificate with the new local CA.
  - a. Select **Client CSR** for the **Renewal Type**.
  - b. Expand the **Certificate details** section.
  - c. Provide the **CSR** from step 1, either as a file upload or by pasting the CSR contents as text.
  - d. Click **Renew Certificate**.

## How to renew in the ksctl CLI

The CA certificate chain renewal process contains the following steps:

1. Create a new local CA using the same attributes as an existing CA.
2. Renew client certificates issued by the original local CA.
3. Renew server certificates issued by the original local CA.

### ksctl CLI: create a new local CA using similar attributes as existing CA

The CipherTrust Manager provides a utility that copies the CSR attributes of the old CA while creating a new local CA. Only key-pair and validity/expiration date are altered. To copy the CSR attributes of the old CA, use the `copy_from_ca: <old_ca_id>` field.

#### Example request



```
{
 "id": "305c1af0-e087-4004-aeba-d59f275ac477",
 "uri": "kylo:kylo:naboo:localca:305c1af0-e087-4004-aeba-d59f275ac477",
 "account": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2023-03-22T07:04:18.33905Z",
 "updatedAt": "2023-03-22T07:04:18.33905Z",
 "name": "example-CA",
 "state": "pending",
 "csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIIEIjCCAr4CAQAwUjELMAkGA1UEBhMCVVMx CzA JBgNVBAgtAlRyMQ8wDQYDVQQH\nnEwZBdXN0aW4xDzANBgNVBAoTB1RoYWxlczEUMBIGA1UEAxMLZXhhbXBsZS5jb20w\nnggiImA0GCSqGSIlb3DQEBAQUAAICDWAwwgIKAoICAQRDR4xfGAJcGT3eg432YHXF7\nnm+5Vtm5lDATk6sdDL8oEJFrHNvftZG8hNdW8hk2pNmBFoKk2DRF8sAPC8t/3KumP\nnkZLdzKdhi5V088dOr4crz/pj/zd0r7X3E77WSNr5R+O5kIa/UC9txQB i5TtxsJeo\nn45clpIvcNfADVrYdZUtnW/60zuNEreTvKTztlcYaV30aYsDe6qAKR/5jQKlpXWwZ\nn4jSrIwE0PRZIoGDhgVPn62dx3Ti3cSaZresr56sfFtVgyn8IQKNqDCGQi+a2c5TC\nntDi5MV/JCi2aPM7I5Tx3bunbldkYye/tLtP6awT66FUgGTFcGZrg5USkpCLM/GC8\n27GekIL4AYlSyXOWrO98uu8Xkm16iMwGSmAbModr2rNqSKmsSkuzER6JCQ4GstKZS\nn13NGUlz9qFivE24QM7eh+acz23eOFVGowAq+gA9atQyrkhVOBu/st3MpHsOCUB/\nnSBofYFpSzROUQ9rSr/H/dSR/wGKFEnr6NnuR+otESQNe6GwzNWqGdkLyTxq6H47v\nn/j1UwYaEWQolj6YtX0MUVR6HNWv5GNXkAgvPbBAP7e6k4+tDJQ3J0Zv8HjN1YUhe\nnKo9KhKcVTOMsfD0PCTnd/JIGtlR2KBwmue5QvBRulfozlOPKX07py6NPHGmcdnjQ\nn5Fdwia0orlioLD8nj8OGCQIDAQABOD8WPQYJKoZIhvcNAQkOMTAwLjAsBgNVHREE\n\nnJTAjgSF0ZWNo bmljYWwuc3VwcG9ydEB0aGFsZXNNcm91cC5jb20wDQYJKoZIhvcN\n\nnAQENBQADggIBALy6p5tjadzg7WJzf5K2eOZ4LNS7Nm7lqRNc68wHCULYViobF8WF\n\nnpUAXK/Iz7apKLGIwPT8Ssz39r3JDtN00UIAWVxsrnRPvmiPEjuDICjlwWswSYCoA\nnrW8hlUR8Ph5dzcto88vSxJpc2TvfaW6h46/+Day/iKbAjlgRMiqFA6Prha7aJo6\nnGphMfQx7EQWXsqeo3EIDfbzVKwn5zd676yKiYKSowtR5B6pgftfwblY6iVldjpX\n\nnxbyYjvbfZk6G1TlLT0ZC5Yxs4foDj0WyKn7XK6QkJ+VDaHWX76Z9zfnPosHZmWqp\n\nnFFVvn1P0afRi0vKyulBV6pbQEVrEtgG/uaaxUmlItYksIVSgjuPLgrLrj46lYUxc\n\nnAbwxJRvtwZNWpfxxgrqQeJrN/+5gGU/div4o08py5Yq75G8o2fBzXbjK9/pyNe0Mo\n\nn5Y0lvP3Id5hWHZDNANDlTjxqqgzogGU2vgxn6GTesgsOxjwzunOO6YQTzbHMq6ZHe\n\nn10xncYrMsiiilVE6ZvG8SZipODtPvUF2C/EcZmG7VSTqNYw9szc6vu20RlkeBds2\n\nnUuvvw9t069IRfLv6OuSQJNQQuGHjzvFWv068q0spDuAdxPTWYNXnBhKaPC6YNDGE\n\nn6/tjVlldCwtbfhKesGs16A6lu1hkOXbJVZvKFczrzmgOTmcYcG+B2ARS\n\nn-----END CERTIFICATE REQUEST-----\n",
 "subject": "/C=US/ST=TX/L=Austin/O=Thales/CN=example.com",
 "notBefore": "0001-01-01T00:00:00Z",
 "notAfter": "0001-01-01T00:00:00Z",
 "sha1Fingerprint": "7666371994B5A8EB88F25925108EEE381629F20A",
 "sha256Fingerprint": "48D3918E611D51FB37643146461FBB78B582AF528766C4E76FF2350F6011152F",
 "sha512Fingerprint": "491913CB31D2567BFE7EE85B9F01B4D0E63A1B64C06E145EB111002FA6C649F28D4A9C52C307ECF554BC35CF906A1E1EFBB1FD29A9CF193E99144B37CD86E1FE"
}
```

## ksctl CLI: renew client certificate using new local CA

1. For clients that have a **profile** associated, use the following steps to renew the client certificate:

- Thales Cybersecurity Docs Hub © 2019–2026, Thales Group. All rights reserved.  
CipherTrust Manager | Last Updated: January 29, 2026

### Example request

```
ksctl clientmgmt profiles update --profile-id 2fa716db-2264-473f-86a6-1c0d31c69fdc --ca_id localca-b1963e85-02e1-462d-a436-a5e11cecca01
```

### Example response

```
{
 "id": "2fa716db-2264-473f-86a6-1c0d31c69fdc",
 "uri": "kylo:kylo:client-management:generic-client-profiles:ceb226b8-8151-4e26-ab6c-038b4ae797e8",
 "account": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2023-01-12T17:33:11.690693Z",
 "updatedAt": "2023-01-12T23:27:06.275532Z",
 "ca_id": "localca-b1963e85-02e1-462d-a436-a5e11cecca01",
 "csr_params": {
 "csr_city": "example-city",
 "csr_cn": "example.com",
 "csr_country": "example",
 "csr_email": "john.doe@example.com",
 "csr_org_name": "example",
 "csr_org_unit": "example tech",
 "csr_state": "example-state",
 "csr_uid": ""
 }
}
```

- b. Renew the client certificate.

### Example request

```
ksctl clientmgmt clients renew --client-id 1e33456b-8782-43a2-9efe-b415dc76ce52 --cert-duration 730
```

### Example response

```
{
 "client_id": "1e33456b-8782-43a2-9efe-b415dc76ce52",
 "cert": "-----BEGIN CERTIFICATE----- MIIDfTCCAwwGawIBAgIRAPvsR8igXyZzpMAnmkgldAwwDQYJKoZIhvcNAQELBQAw WjELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAK1EMRAwDgYDVQQHEwdCZWxjYW1wMRAw DgYDVQQKEwdHZW1hbHRvMR0wGAYDVQQDExFkZlZX1tZW11cmUgUm9vdCBkQTAeFw0x ODEyMTgwMDE2MjZaFw0yMDEyMTcwMDE2MjZaMBExDzANBgNVBAMTBnRlc3RlcjB2
```

```

MBAGByqGSM49AgEGBSuBBAaIA2IABEvBmz1WRQmfiG2IG0jE7fpPyDTCNwvqSXsW HAhrVCRDOM
PLuaiVn08/k7zRFum5UxcIWjwxJ5tnO7Z38Y3gKIyE42mHInqQHPOT cz9JLKqaGALwZtQCzB61
M0ul7dGA5aM1MDMwDgYDVR0PAQH/BAQDAgOIMBMGA1Ud JQMMMAoGCCsGAQUFBwMCMaAwGA1UdEw
EB/wQCMAAAwDQYJKoZIhvcNAQELBQADggIB AK7rTYW1+woOfHOLeYjs6jobO7kRoM71ffdVwcHI
MS3IE0B1eLdteKdG3yy2znAy VU7Jkwo2396Z9cPofrKt95wURakSYvtz3IpTL9ibrpqJ47XxEX
LHl+OycWdYoqAm YJe4A/mW3OxdR4kPbxnDXPNMZiId2xSyzkrEqafTBBtlkdjuljEfQraKW7Ti
Qovd dKb8xzAgozuZ4C200G1KbgjPkRF4iEXk6sihzYikmyE0s5VBeyAGvdv+s6rv6+4n mbaLk
TF/ReXJryIRLDJ1uWN/PDKIqGyU1IrB26wYUWEG+4xcTlLqBxS2HL0ko1Cr 5yeWMEo952YyGeM
wW0oWzhIDMxPVRXEFru0nG35K2Gpz4KywhFVqK11rd7/FLwUH mrMtMwr5LG14I1NG3kEz+UVcd
wfCeYxnIGW/u9CbUSmedlklZtuXjEN6bQdP+oZi f32u0mI4MSHYK55bdMWw7Rr4IlgDKRdUD0l
7luZt8nztQuWVHTrii34gN5Hvz4EY g7jpDq9ZXpb1ZtLmEq2TM8XzyBzJkdIAT304L666826c1
elkOgsZQw08W72ju02B 1qj/HtqGoRXPwlvk+y2XIYIwcPP3T6YctJA6TMafZ11IKoWWflT0uqF
o19CadC8z PylaiQwwuJGV7MmJ71C8LmYUP2Pj2v+S+5s8j0QgY0C5 -----END CERTIFICAT
E----- ",

```

```

"csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIICVzCCAT8CAQAwEjEQMA4GA1
UEAxMHA21pcDEyMjCCASIwDQYJKoZIhvcNAQEB\nBQADggEPADCCAQoCggEBAO93JWGgUtIJKoZ
NgpzYBrLQgPKxaKgn42Js9pxeIAEo\nTvDfPGk2bKgOO+7GsoKCIthRn6/4fkd51TwR3tBK3Y2X
s9TIkBQ+gpzyAM1bIlTf\nXd8xVaHgsVNS58laY1FTqM+jlVIfAlpKJnboYkGc8n6aCt9kgWdt5
61Nc0AfVbj0\nTD8n5wTm7uJy1GuFiwCuYbaVuEsHZbpNh3GJ1tvXpRxyp7IzdCc+244cvat2L5
xZ\niDiV4BeoOG3gfddQ9WuqWY+6TVAdZNL7JVMW+3qofib1uHyCHNw0Bec/IMA48gg\n7JNnS
wDB6FXSeYr7nqfHORWOaQi7DT7F6JdY3cXXuNMCaWEAAaAMA0GCSqGSIB3\nDQEBcWUAA4IBAQ
BpUtybSG6DG5J3LR0kGj3/qcvu2FdZ6oCDq+B3Pnz06iJX2w4E\nnFZGIGMYotqlm0DXv4xODFOM
iLa8D8waef/+cN7dihPq1wKqW6M12I0/5nNY/51c4\ntuCRVDZ5zuBLVfw77yp93+VqwUHKP343
98PcsYwtafm9jQM41T7mLlaTjynVmyoF\nnitocPLQLdXMBakAWPpu/+XJt4rGPCh35dv8ojPyCh
R0H43NMcXNX8sw2MzVwAHSE\nnNJBcgC/6IIME8yNcljV3YTywe0VkiVJHgA5rJN9OwV3M3Hfji/
9S/u3pD1Ixt048\nnDJXbUwe5ubTKH9Eqo6TIuIsxdreKz1ONvlyV\nn-----END CERTIFICATE
REQUEST-----\n",

```

```

"ca_cert": "-----BEGIN CERTIFICATE-----\nMIIFoDjlbgNhbXAxEDAO\nMTUxMFow
WjELMAkGA1UEBhMCVVMxMzA5BjBjNV\nBAGTAk1EMRAwDgYDVQQHEwdCZWxjYW1wMRwDgYDVQQKE
wdHZW1hbHRvMR0wGAYD\nVQQDExFLZX1TZWN1cmUgUm9vdCBDQTCCAiIwDQYJKoZIhvcNAQEBBQ
ADggIPADCC\nAgoCggIBAPgJPiSciHZcAPMnEv7dpP1/jc82V9a9pmOIU2jkE7xIvhc7wQ/xVYZ
X\npl0c/+9v4YEcle/Gjks17v04hOg+klf10lpTTp2ctdUd83gECDVrwpUUMpFtdhiL\nnAC/hXN
GobnJxjEMZPV3/gZIkxR4jDoa8A3FiLL5xLoWc9YLn85JD1YRVE1rdcpGw\nn0ElTNRoko1mUJ1g
90mXBie7TGHdHR6gtbloSNZOUBFlf0P17pQPLyzZxR3tlq3qo\nn/l/+hdcYfLw/Jf323c30CbuV
FFbYQzAdmB6k0rZaajQMZJIhYO+EUt7HkrF/gU6E\nnj0uq18yxQxsXnxs2n94fpeSWF/UfuIIkj
J8mA6yGgkgT3Nw/MoD+8eTnMeoAH04S\nnbm3alpi7nlVKYdRednFphxx9YmkIMy+2VQoWfVmKvJ
TxCtE7rzElZsqKQ6ZFvtPi\nn71YPlt0gWwHMKwY41FuUYPMCH7x7Zzb/adggES17DhmrqUivIEQ
gl4VYQSBKk/b3\nnPQ64+iXhtnLDiiSneKErEvMqA81RIqWd3c6XG07+6YTFoL3peOEm5XWw0Kvz
DhUT\nnomJkNTsh+Og4OXBtLXSCJzUveY6yuxALb6GaSS0a90k34/iRP71BES00EtngH3lr\nnQhO
VYibGMKfJDSMEEFcATbY4fBn1uj1RrAUhQ3GlaU/1LzZ8gjDagMBAAGjYjBg\nnMA4GA1UdDwE
B/wQEAWIBBjAPBgNVHRMBAf8EBTADAQH/B0GA1UdDgQWBbTzcq97\nnqHASsOatm3N+6Iq1TD0gI
zAeBgNVHREEFzAVGRNzdXBwb3J0QGdlbWFsdG8uY29t\nnMA0GCSqGSIB3DQEBcWUAA4ICAQBmwT
dayCb9gBlAKJVhW5mBh+muajk53cXxaXJx/VwLe\ntyNQQZhV5r6AIgdSLuy8UPj9rWeVMeI4xW
utdy/ANj6737pZr4WjNNBirVtkDhRh\nnMZtV9Q==\n-----END CERTIFICATE-----\n"

```

}

- For clients that don't have any profile associated, run the following command and pass the `ca_id` in the request:

#### Example request

```
ksctl clientmgmt clients renew --client-id 72e24314-dc1e-4bfe-80c1-8127900cee9
b --ca_id 911671ca-6c15-45cf-a0a8-bcde9e82f721 --cert-duration 730
```

## ksctl CLI: renew server certificate using new local CA

To renew the server certificate using the new local CA, perform the following steps:

- Issue a new server certificate from the new CA.
- Create a certificate chain and combine them into a single PEM or PKCS12 file. The file should include the following in the given order:
  - Server certificate
  - Any intermediate certificate authorities that sign the server certificate. Start with the intermediate CA that issued the server certificate. Next, add the issuer of the intermediate CA, if any. Continue adding any intermediate CAs higher up the hierarchy, each time adding the issuer of the last certificate you uploaded.
  - The root CA that signs any intermediate CAs. If there are no intermediate CAs, add the root CA that issued the server certificate.
  - The server certificate's private key.

#### Note

- In the PEM format, the server certificate's private key should be included first, and in the PKCS12 format, it should be included last.
- It is optional to include the private key in the PEM file if the user has previously generated a CSR. Creating a CSR allows the user to generate the key pair on the CipherTrust Manager and prevents exposing the private key outside of the CipherTrust Manager.

- Upload the certificate chain to the CipherTrust Manager.

#### Example request

```
ksctl interfaces renewal-cert upload --name nae --file ./serverCertKey.pem --format pem
```

### serverCertKey.pem

```
-----BEGIN CERTIFICATE-----
MIIEqjCCApKgAwIBAgIQZkQE+oYbUzqL+bmYF2f3VjANBgkqhkiG9w0BAQsFADBa
MQswCQYDVQQGEwJVUzELMAkGA1UECBMCMVFGxZANBgNVBACTBkF1c3RpbjEPMA0G
A1UEChMGMGVGhnbGVzMRwwGgYDVQQDEXNDaXBoZXJUCnVzdCBSb290IENBMB4XDTIz
MDMyMjA2MTEwOVoXDTMzMjMDMxODA5MzM1OFowDjEMMAoGA1UEAxMDbmF1MIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuB2UJMu2lMYSx181SaLKXxuOrz39
DFd6yae+fzFYXzyZxudRDeHMSt0knPoV2L3WsfP8PibtAKHyUK6b8Ego9Ccah16X
YeOmeehBcYk/IwbUcp0LAW4/4lB/a++Gkd9HPmUmOezbVMOYjk4LoerSLiqfbXIUh
pbrFcXuTNKjvelymMMT2LubguXWu6xPB6faZtspn3ZB22FgqT4KPULUrDuiTTJ3
wXJyKCIEb5kpeIk56u6aE2N+IGQ2gFWN9tHWr6SwjQNm7QhdV63y5+JLMXyHuINf
Lr24fHEBUoSptbu4yhTu5t7h1O5TTKIV/PBjjvbYtt0uYIcQhw3U5qhQWQIDAQAB
o4G3MIGOMA4GA1UdDwEB/wQEAwIDiDATBgNVHSUEDDAKBggrBgEFBQcDATAMBgNV
HRMBAf8EAjAAMB8GA1UdIwQYMBaAFDEexvfPs6aqKYwDCCc6r7dQUEhKMF4GA1Ud
HwRXMFUwU6BRoE+GTWh0dHA6Ly9jaXB0ZXJ0cnVzdG1hbmFnZXIubG9jYWwvY3Js
cy9iMTk2M2U4NS0wMmUxLTQ2MmQtYTQzNi1hNWUxMWNlY2NhMDEuY3JsMA0GCSqG
SIb3DQEBCwUAA4ICAQA+4SrI4C4214C8Aw5CA1bPdLFpOMoABwaCg2p0DqR0Su7E
XZSZeXod8w/5aNAJ2rrHCJqwaDoKehNfaa/fQ/wDhVPkw6Nn2gjn6IEVNpxgU
BQbE+i0vmU0NnsHF6VEMh3299sf94K9IxHqZbDgogBeBLxtqjvLSsECeoHmMfL2V
/GfzMXKsiHp97gNwK8Mt6vWA4uJ/ojhK5mUlfRoEp4MyH6IuauLoWMNDGQ9xEmj9
RYKzZgZsEraS3sOLj3ZWDkpF/tGMAiadqRav9Zyb7/mMJ014RZdpRKMTLFQ/xqnM
ZPsKYj0g+Q9BfK2vRKmTX2phTnjqeoVTy3qB6XxIHItmViURCT0t40tva7Ici89U
xYanshAtFyCzq+1aMKbt+SeDD6jo2Q4LBNnMTK8liZqMhNUS3A0rz1mTlpUbiLbi
d4Nz8d55iJxwOMSkbdQvWpoXHJnDEQW9d/9KFgnJu4nR8192Yta5BuxvuZHY57HL
riA3FJ+VFzZeZpEJOdUIpaAfGcsMBWd+irH+uPA70bcrkIlF+9YCr81LOcv7gY7l
8cIMrAikMjY3YMOgXmqxzHq1WeoOHKFrtrjYJDL9gVeKn2EmMWCSYY11KXzZ1pgG
mdEOyrSYjCYgMyj5vp/Irfl57KwVVUo5DHZVx9fBLWAosKdeXNgyrXwhEUzx9A==
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAuB2UJMu2lMYSx181SaLKXxuOrz39DFd6yae+fzFYXzyZxudR
DeHMSt0knPoV2L3WsfP8PibtAKHyUK6b8Ego9Ccah16XYeOmeehBcYk/IwbUcp0L
AW4/4lB/a++Gkd9HPmUmOezbVMOYjk4LoerSLiqfbXIUhpbRfcXuTNKjvelymMMT
2LubguXWu6xPB6faZtspn3ZB22FgqT4KPULUrDuiTTJ3wXJyKCIEb5kpeIk56u6a
E2N+IGQ2gFWN9tHWr6SwjQNm7QhdV63y5+JLMXyHuINfLr24fHEBUoSptbu4yhTu
5t7h1O5TTKIV/PBjjvbYtt0uYIcQhw3U5qhQWQIDAQABAoIBAQCmXywgN+kr9PTB
LlqDyNPg5G272f7wA0Xrjnoy9lUbflzkFFhHvoxadyO17ooGhuoxnWA4EhOAw9q
0u3cqgg/4BoL20EmVsV3gZWpl+DysNhhXv1df/KbP1HtrE+nanOSy7IxyzHivP1
vbarQPtedmuMRbrCjZOvJgVB7FHEbAP8BG+qsGHU/6EOLQBvskoR7Edwuz/q12n4
1as0Sy0qAtLhPcdOA+iUXWvxTsm5GVO/+9I/xs/SDT4ki8CgBgyQpuH0B8j5Gw9W
Y3NSn8ahXhy9mh1tGY41tJSby6F3YUy57I117loJ8w4fRqfQEdxr6rQgxvRNM36h
```

```
DuSefqxFAoGBAOzAcvrbSW7uiHjnz37xbr7AdegF47dtOJnoIXYKNF3MQ3lw5PxM
wc+SMJqpL84sG6OZXxs9qOxj2ZCrkthzPJxEhnOp1qGd1jvL4pW5N4e8cXF3i5k/
qY+12JZOHy55tIO8vUXwma0bDnU+YrC7APb6AtaOVi96EjU8QITzqmXAoGBAMcV
mpovSrjbVrAEe9/XPmnlk0Sk2GDWSHGicdwCVXmWqwTA20Yra8z6Ax26ARcotKw
eTyEMA5LQh112MytLaAqTfjIPrpD3ZsIJ0JWYmlvOLwxqVcgesx+YaEvN8tiBOR4
HYNWJteDWgXHrbNX45Q6BOYDPorj0a5vb4P91bOPAoGAeqBBa4L9EAEsM3bpWC9e
axtxK0PrWm75WQJuP38mB4sec9tx6HeRd+ckc8aDwFCwC/rxBI+hTpe8cilNcEIA
rrqlaUwFwq+0PxxgXR3g+6irlMewZy4C6s1Z1571VqYImqrgKXUX1QWdIbHRY/ZXA
Q711UA9VG6o1MCWwoF02kRkCgYEAqb4r0WadKTWqj20I9dD8LwjKx/AcpTnel0Nd
rbRC4XDEwo57j1tJS2bQZUBE1uM9GLLKy+RRLP7R4kriLLFDg5pOXN4vTZVrrzGm
d5M72XWxRBR7tAPHb/AZwdqGkeyDC2G2mDKub3ZZHTCBU7aOHeXSI2OEwdsQEBCF
v16BBNUCgYEApKwKLv+u9KrK/bdQCYsNTLXTeuiZh8FbnYZtT3dXvgsFUiCuDOGI
j1XLL6+p7Cs+O2k145WCb6yFzk5X4Z9y0gV7jIkumRX07QA5K5Zdfjrx4cBjn2y
8ZnEXTu3PkbGPlkIaIaASW19C5JgLoRCPTKT/taDJaSKa/L4RN1iv2o=
-----END RSA PRIVATE KEY-----
```

### Example response

```
{
 "certificates": "-----BEGIN CERTIFICATE-----\nMIIEqjCCApKgAwIBAgIQZkQE+oYb
UzqL+bmYF2f3VjANBgkqhkiG9w0BAQsFADBA\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCFgxDZAN
BgNVBACTBkflc3RpbjEPMA0G\nA1UEChMGVGVhbGVzMRwwGgYDVQQDEwNDaXB0ZXJUCnVzdCBSb290
IENBMB4XDTEz\nnMDMyMjA2MTEeXOV0XDTMzMMDMxODA5MzM1OFowDjEMMAoGA1UEAxMDbmFlMIIBIjA
N\nnBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuB2UJMu2lMYSx181SaLKXxuOrz39\nnDfD6yae+f
zFYXzyZxudRDeHMSt0knPoV2L3WsfP8PibtAKHyUK6b8Ego9CcAh16X\nnYeOmeehBcYk/IwbUcp0LA
W4/41B/a++Gkd9HPmUmOezbVMOyjk4LoerSLiqfbXIU\nnHpbrFcXuTNKjve1ymMMT2LubguXWu6xPB
6faZtspn3ZB22FgqT4KPULUrDuiTTJ3\nnwXJyKCIEb5kpeIk56u6aE2N+IGQ2gFWN9tHWR6SwjQNm7
QhdV63y5+JLMXyHuInf\nnLr24fHEBUoSptbu4yhTu5t7h105TTKIV/PBjjvbYtt0uYIcQhw3U5qhQW
QIDAQAB\no4G3MIG0MA4GA1UdDWEB/wQEAwIDiDATBgNVHSUEDDAKBggrBgEFBQCcDATAMBgNV\nnHRM
BAf8EAjAAMB8GA1UdIwQYMBaAFDEexvfPs6aqKYwDCCc6r7dQUEhKMF4GA1Ud\nnHwRXMFUwU6BRo
E+GTWh0dHA6Ly9jaXB0ZXJ0cnVzdG1hbmFnZXIubG9jYWwvY3Js\ncy9iMTk2M2U4NS0wMmUxLTQ2M
mQtYtQzNi1hNWUxMWNlY2NhMDEuY3JsMA0GCSqG\nsSIb3DQEBCwUAA4ICAQA+4SrI4C4214C8Aw5CA
1bPdLFPOMoABwaCg2p0DqR0Su7E\nnXZSZexDxod8w/5aNZAJ2rrHCJqwaDoKehNfaa/fQ/wDhVPkw6
Nn2gjn6IEVNpxgU\nnBQbE+i0vmU0NnsHF6VEMh3299sf94K9IxHqZbDgogBeBLxtqjvLSSECEoHmMf
L2V\nn/GfzMXKsiHp97gNwK8Mt6vWA4uJ/ojhK5mUlfRoEp4MyH6IuauLoWMNDGQ9xEmj9\nnRYKzZgZ
sEraS3sOLj3ZWDkpF/tGMAiadqRav9Zyb7/mMJ014RZdpRKMTLFQ/xqnM\nnZPSKYj0g+Q9BfK2vRKm
TX2phTnjqeoVTy3qB6XxIHItmViURCT0t40tva7Ici89U\nnxYanshAtFyCzq+1aMKbt+SeDD6jo2Q4
LBNnMTK81izqMhNUS3A0rz1mTlpUbiLbi\nnd4Nz8d55iJxwOMSkbdQvWpoXHJnDEQW9d/9KFgnJu4n
R8192Yta5BuxvuZHY57HL\nnriA3FJ+VFzZeZpEJOdUIpaAfGcsMBWd+irH+uPA70bcrkIlF+9YCr81
LOcv7gY71\nn8cIMRAIkMjY3YMOgXmqxzHq1WeooHKFrtrjYJDL9gVeKn2EmMWCSYY1lKXZz1pgG\nnm
dEOyrSYjCYgMyj5vp/Irf157KwVVUo5DHZVx9fBLWAosKdeXNgyrXwhEUzx9A==\nn-----END CERT
IFICATE-----\n"
}
```

- For clients, the chain uploaded in the previous step is available to download.

### Example request

```
ksctl interfaces renewal-cert get --name nae
```

### Example response

```
{
 "certificates": "-----BEGIN CERTIFICATE-----\nMIIEqjCCApKgAwIBAgIQZkQE+oYb
UzqL+bmYF2f3VjANBgkqhkiG9w0BAQsFADBa\nMQswCQYDVQQGEwJVUzELMAkGA1UECBMCFgxDZAN
BgNVBACTBkF1c3RpbjEPMA0G\nA1UEChMGVGVhbnBzMRwwGgYDVQQDExNDaXBoZXJUCnVzdCBSb290
IENBMB4XDTEz\nnMDMyMjA2MTEeXOVoXDTMzMMDMxODA5MzM1OFowDjEMMAoGA1UEAxMDbmF1MIIBIjA
N\nBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuB2UJMu2lMYSx181SaLKXxuOrz39\nnDFd6yae+f
zFYXzyZxudRDeHMSt0knPoV2L3Wsfp8PibtAKHyUK6b8Ego9CcAh16X\nnYeOmeehBcYk/IwbUcp0LA
W4/41B/a++Gkd9HPmUmOezbVMOyjk4LoerSLiqfbXIU\nnHpbrFcXuTNKjvelymMMT2LubguXWu6xPB
6faZtspn3ZB22FgqT4KPULUrDuiTTJ3\nnwXJyKCIEb5kpEik56u6aE2N+IGQ2gFWN9tHWR6SwjQNm7
QhdV63y5+JLMXyHuInf\nnLr24fHEBUoSptbu4yhTu5t7h105TTKIV/PBjjvbYtt0uYIcQhw3U5qhQW
QIDAQAB\no4G3MIG0MA4GA1UdDWEB/wQEAwIDiDATBgNVHSUEDDAKBggrBgEFBQCcDATAMBgNV\nnHRM
BAf8EAjAAMB8GA1UdIwQYMBaAFDEexvfPs6aqKYwDCCc6r7dQUEhKMF4GA1Ud\nnHwRXMFUwU6BRo
E+GTWh0dHA6Ly9jaXBoZXJ0cnVzdG1hbmFnZXIubG9jYWwvY3Js\ncy9iMTk2M2U4NS0wMmUxLTQ2M
mQtYTQzNi1hNWUxMWNlY2NhMDEuY3JsMA0GCSqG\nnSIb3DQEBwUAA4ICAQA+4SrI4C4214C8Aw5CA
lbPdLFPoMoABwaCg2p0DqR0Su7E\nnXZSZexDxod8w/5aNZAJ2rrHCJqwaDoKehNfaa/fQ/wDhVPkw6
Nn2gjn6IEVNpxgU\nnBQbE+i0vmU0NnsHF6VEMh3299sf94K9IxHqZbDgogBeBLxtqjvLSsECeOHzMf
L2V\nn/GfzMXKsiHp97gNwK8Mt6vWA4uJ/ojhK5mUlfRoEp4MyH6IuauLoWMNDGQ9xEmj9\nnRYKzZgZ
sEraS3sOLj3ZWDkpF/tGMAiadqRav9Zyb7/mMJ014RZdpRKMTLFQ/xqnM\nnZPsKYj0g+Q9BfK2vRKm
TX2phTnjqeoVTy3qB6XxIHItmViURCT0t40tva7Ici89U\nnxYanshAtFyCzq+1aMKbt+SeDD6jo2Q4
LBNnMTK8liZqMhNUs3A0rz1mTlpUbiLbi\nnd4Nz8d55iJxwOMSkbdQvWpoXHJnDEQW9d/9KFgnJu4n
R8192Yta5BuxvuZHY57HL\nnriA3FJ+VFzZeZpEJOdUIpaAfGcsMBWd+irH+uPA70bcrkILf+9YCr81
LOcv7gY71\nn8cIMrAikMjY3YMOgXmqxzHq1WeoohKFrtrjYJDL9gVeKn2EmMWCSYY11KXzZ1pgG\nnm
dEOyrSYjCYgMyj5vp/Irf157KwVVUo5DHZVx9fBLWAosKdeXNgyrXwhEUzx9A==\n-----END CERT
IFICATE-----\n"
}
```

Now, the clients can use this new chain to setup their environment.

- After the clients have been updated with the new certificate chain, the *admin* can apply the new certificate chain on the CipherTrust Manager. This replaces the existing server certificates with the newly generated server certificates.

### Example request

```
ksctl interfaces renewal-cert apply --name nae
```

---

There will be no response if the new certificate chain is applied successfully.

If *admin* doesn't apply the new certificate chain manually on the CipherTrust Manager, the auto-apply option is triggered when the CA expires.

## Adding an External CA

You must add an external CA on CipherTrust Manager to use the CA as a trusted entity for various interfaces and services inside the system.

You need to obtain and upload the external CA's PEM-formatted certificate file.

### Add an External CA in web console UI

1. Login to the root **domain** as an Application Administrator in the Admins **group**, such as `admin`.
2. Navigate to **CA > External**.
3. Click **+ Add External CA**.
4. Provide a **Display Name**
5. Provide the CA's **Certificate**.
  - Select **File Upload** and **Upload Certificate** to provide the certificate file.or
  - Select **Text** and paste the certificate contents into the text field.
6. Click **Add External CA**.

### Add an External CA in ksctl CLI

Use the following command:

```
ksctl ca externals upload --cert-infile mycert.pem
```

## Trusted CAs in CipherTrust Manager

You can add local and external CAs as trusted CA in CipherTrust Manager. Trusted CAs can be added for the following services:

- Credential Storage
- Cloud Key Management
- Secrets Manager



- HSM secrets

By default, the local root CA is added to the trusted list for all four services mentioned above.

#### Note

- Only the root domain can add Trusted CAs.
- Only users with System Admin and Admin permissions can add and view Trusted CAs.

## GUI

### Adding Trusted CA

1. Log on to the CipherTrust Manager as an administrator.
2. Navigate to **CA > Trusted CA**.
3. Click **Add Trusted CA**.
4. In **Select CA**, select the list of CAs that you want to add as trusted CA.
5. Click **Next**.
6. In **Select Services**, select the service to which Trusted CA is to be added.
7. Click **Next**.
8. Review the Trusted CAs selections and click **Add Trusted CA**.

A cluster-wide restart of the microservice is needed to apply the changes (to reflect the added CA in the trusted store). You can restart the service immediately or postpone and restart it manually later.

9. For immediate restart, type "RESTART" and click **Restart Now**. This will restart the services cluster wide and can affect selected services temporarily. Otherwise, click **Restart Later** to manually restart the services some time later.

#### Note

If you do not restart the microservices immediately, a yellow banner is displayed on the **Trusted CA** page prompting you perform a manual restart for the changes to be applied.

## Removing Trusted CA

To remove a Trusted CA, click the **Remove** button corresponding to the Trusted CA. You will be prompted to restart the service now or later. The restart option will restart the service on the same node in clustered environment, to remove trusted CA from other nodes, restart the nodes individually.

### ksctl

## Adding Trusted CA

You can add a single or multiple CAs as trusted CA in the CipherTrust Manager.

### Single Trusted CA

```
ksctl trusted-ca-cert add --trusted-ca-id <id> --trusted-ca-type local --service
secrets-manager --auto-restart
```

### Multiple Trusted CA using arrays

```
ksctl trusted-ca-cert create-many --trusted-cas <array-of-ca-data> --auto-restart
```

Example of <array-of-ca-data>:

```
[
{
"ca_id": "sample-ca-id",
"ca_type": "local",
"service": "sample service"
}
]
```

### Multiple trusted CAs using JSON

```
ksctl trusted-ca-cert create-many --trusted-cas-jsonfile <filename.json> --auto-r
estart
```

## Listing Trusted CAs

```
ksctl trusted-ca-cert list
```

Flags options:

`--id string` - ID of the Trusted CA resource

`--service string` - Service name where the Trusted CA is applied

`--sort string` - Sort by created\_at, service, ca\_uri, ca\_type, id (comma-delimited)

```
`--trusted-ca-type string` - CA type: local or external
`--trusted-ca-uri string` - URI of the Trusted CA resource
```

## Deleting Trusted CA

```
ksctl trusted-ca-cert delete --id <id> --auto-restart
```

### Note

#### Proxy integration:

- Path to the proxy certificate file: `/opt/keysecure/trusted-proxy-certs/ca_proxy.crt`
- Certificates added to `ca_proxy.crt` are trusted by all four services after a system restart.
- The existing use case, where customers inject certificates into `ca_proxy.crt` for trust, remains unaffected.
- Upgrade behavior: After upgrading to the latest version, certificates in `ca_proxy.crt` will continue to be trusted across all services.

## Using domain CA as trusted entity

To use a domain CA as a trusted entity, you must:

1. Export the domain CA
2. Import the domain CA to the root domain as an external CA
3. Add the imported external CA to the web interface

### Export the domain CA

1. Log on to the CipherTrust Manager GUI as an administrator.
2. Click **Switch Domains** and select the desired domain.
3. Navigate to **CA > Local**.
4. In the **Local Certificate Authorities** table, click the overflow icon (  ) for the local CA that you want to export.
5. Click **Download**.

---

## Import the domain CA to the root domain as an external CA

1. Switch to the root domain.
2. Navigate to **CA > External**.
3. Click **Add External CA**. On the **Add External Certificate** screen:
  - Enter a **Display name** for the external certificate.
  - Under **Certificate**, choose either of the following options:
    - **File Upload**: Click **Upload Certificate**. Select the certificate downloaded in the [Export the Domain CA](#) step and click **Add External CA**. The certificate is added to the list of External Certificate Authorities.
    - **Text**: Paste the content of certificate that you downloaded in the [Export the Domain CA](#) step and click **Add External CA**. The certificate is added to the list of External Certificate Authorities.

## Add the imported external CA to the web interface

1. Navigate to **Admin Settings > Interfaces**.
2. Click the **web** interface.
3. Under **External Trusted CAs**, select the external CA that you imported.
4. Click **Update**.
5. Restart the web interface for the changes to take effect.

The domain CA is now added to the trusted CAs list.

## Verifying revocation status of client certificates

The CipherTrust Manager can be configured to verify the revocation status of client certificate presented to NAE or KMIP interface before establishing a connection with the client.

To configure the CipherTrust Manager for inspecting the client certificate revocation status:

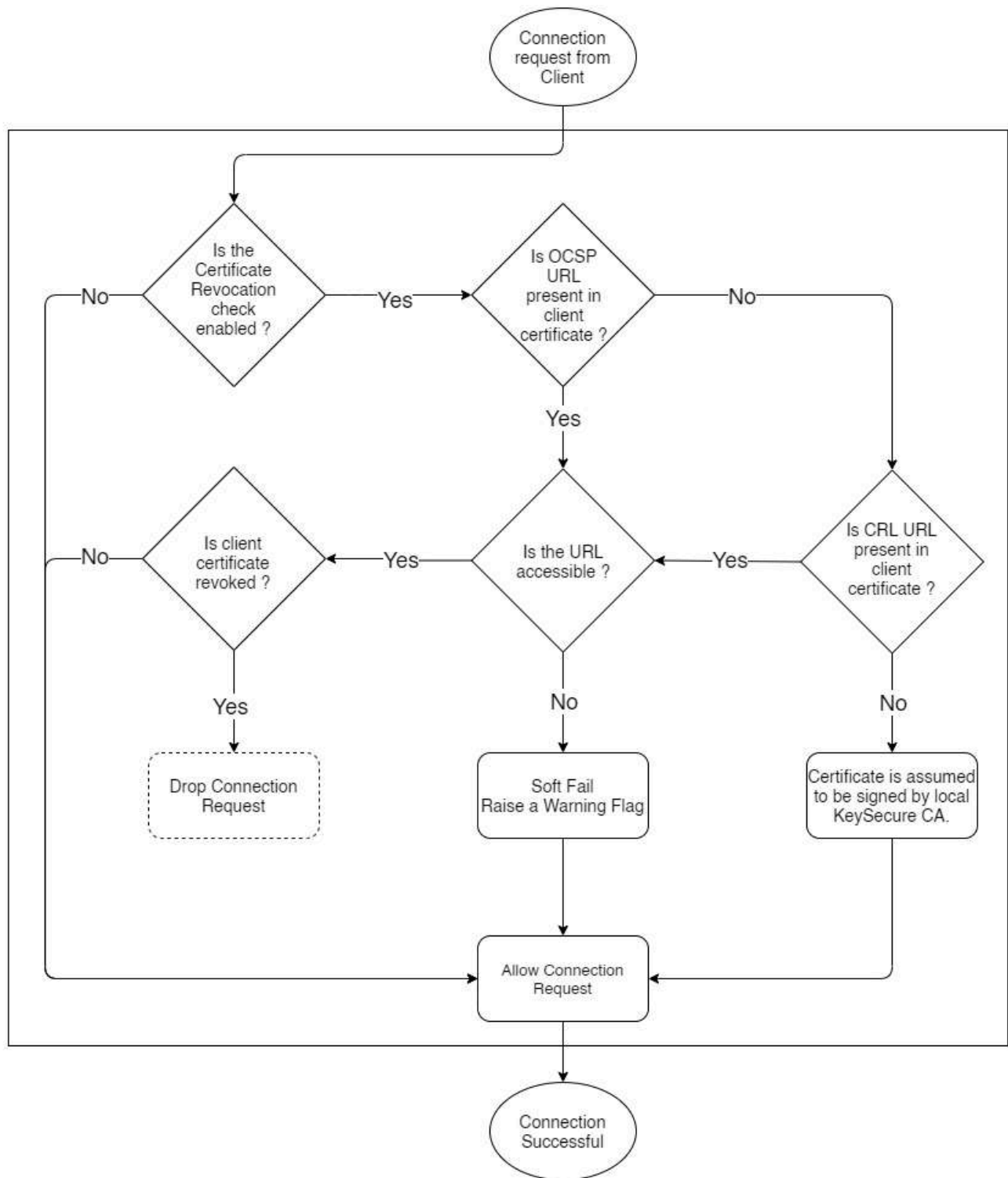
- Select the `TLS verify client cert` [Interface Mode](#).
- [Enable](#) the Certificate Revocation Check.

A certificate contains an OCSP responder URL and a Certificate Revocation List (CRL) URL, which are used for verifying the revocation status of the certificate.

**Note**

An LDAP URL is not supported, that is, if the CRL or OCSP URL begins with **ldap://**, the CipherTrust Manager skips that URL.

Let's understand how CipherTrust Manager verifies the revocation status of a certificate and permits/drops connection requests in such cases:



CipherTrust Manager looks the client certificate for OCSP responder URL and CRL URL.

#### 1. If OCSP URL is present

CipherTrust Manager accesses this URL to verify the revocation status of the client certificate.

**Note**

- The certificate to be verified must contain the URL of the certificate issuer(CA). The issuer certificate should be specified using the AIA extension of X.509. The issuer URL should be a `http` URL and must be accessible.
- If the issuer URL is not specified in the certificate, the certificate revocation check "Soft Fails" and the connection is created.

- a. If OCSP URL is accessible, the status of certificate gets verified successfully, and CipherTrust Manager allows/drops connection request accordingly.
- b. If OCSP URL is not accessible due to any reason, CipherTrust Manager considers the situation as a "Soft Fail". It allows the connection to establish, but reports a warning. This warning audit log and its details can be viewed in the Records.

**2. If OCSP URL is not present, but CRL URL is present**

CipherTrust Manager verifies the status of certificate using the CRL, and allows/drops connection request accordingly.

- a. If CRL URL is accessible, the status of certificate gets verified successfully, and CipherTrust Manager allows/drops connection request accordingly.
- b. If CRL URL is not accessible due to any reason, CipherTrust Manager considers the situation as a "Soft Fail". It allows the connection to establish, but reports a warning. This warning audit log and its details can be viewed in the Records.

**3. If both OCSP URL and CRL URL are not present**

CipherTrust Manager considers the client certificate to be signed by its local CA, and allows the connection to establish.

**Note**

In any case, if the certificate is found to be revoked, CipherTrust Manager drops the connection request and logs it in Records.

## OCSP and CRL caching

As stated above, verifying the revocation status of a certificate involves establishing a connection with the URL (OCSP or CRL) present in the certificate, and verifying its revocation status. Once the revocation status of a certificate is verified, CipherTrust Manager stores this information for some preset time.

---

Let's understand how long this information is stored in and how it is used in this time frame:

- **For OCSP method**

After successfully connecting to the OCSP URL, CipherTrust Manager stores the revocation status of the client certificate for a duration of 5 minutes.

- If CipherTrust Manager receives another connection request from the same client within 5 minutes of a previously successful connection, then CipherTrust Manager refers to the cached revocation status value to verify its revocation status.
- If CipherTrust Manager receives another connection request from the same client after 5 minutes of a previously successful connection, then CipherTrust Manager verifies its revocation status through the OCSP URL again.

- **For CRL method**

Each CA promises to update its CRL at the day and time specified in the *Next Update* field for that CA. While performing a certificate revocation check, the CipherTrust Manager inspects the *Next Update* value for the CRL associated with each CA on the CipherTrust Manager.

- If the *Next Update* value for that CRL is in the past, the CipherTrust Manager attempts to connect to the CRL distribution point (CDP) for the CA to download the updated CRL.
- If the *Next Update* value for that CRL is in the future, the CipherTrust Manager waits until that specified time to attempt to connect to the CDP and download the updated CRL.

The CipherTrust Manager also enables you to fetch the most recent list of revoked certificates (CRL) from issuing authorities before the *Next Update* time. To accomplish this, [configure the Scheduler to synchronize CRLs](#).

Please note that this functionality is currently available only for *Web* and *REST* interfaces and *does not* apply to NAE and KMIP.

## Enabling/disabling certificate revocation check

The Certificate Revocation Check is enabled by default. You can enable or disable the certification revocation Check using:

- API playground

Refer to "Properties" section (*/v1/configs/properties*).

- CLI tool (ksctl)

Refer the following examples:

- Command to enable Certificate Revocation Check

```
ksctl properties modify -n ENABLE_CERT_REV_CHECK -p true
```



- Command to disable Certificate Revocation Check

```
ksctl properties modify -n ENABLE_CERT_REV_CHECK -p false
```

- Available Flags:

| Flags       | Input Type            | Description                                  |
|-------------|-----------------------|----------------------------------------------|
| -h, --help  | <i>not applicable</i> | Command help                                 |
| -n, --name  | string                | Name of the system configuration             |
| -p, --value | string                | Value to be set for the system configuration |

## Getting the fingerprint of the CA certificate

The fingerprint of the old CA certificate that was used to register the client with the CipherTrust Manager is needed when re registering a client. A CipherTrust Manager administrator can provide you the fingerprint.

Fingerprint of the CA certificate can be viewed on the GUI or the API playground.

### On the API Playground

To get the fingerprint of a CA certificate:

1. Acquire an authorization token.
2. In the left pane of the API playground, click **Certificate Authority**.
3. Under **Certificate Authority**, click **Get local CA**. The Get section of the API playground is displayed in the right pane.
4. Click `id` in `/v1/ca/local-cas/{id}`.
5. Enter **id** of the local CA in the text box.
6. Click **GET**. Details of the CA including fingerprints are displayed in the output. Only `sha256Fingerprint` and `sha512Fingerprint` are supported for reregistration.
7. Copy the desired fingerprint. This fingerprint will be used when re-registering clients.

Similarly, you can get fingerprint of external CA certificates.

### On the GUI

To get the fingerprint of a CA certificate:

1. Log on to the CipherTrust Manager as administrator.
2. In the left pane, click **CA > Local**. The list of available CAs is displayed.

3. Click the ellipsis icon corresponding to the CA.
4. Click **Details**. Details of the CA including fingerprints are displayed. Only `sha256Fingerprint` and `sha512Fingerprint` are supported for reregistration.
5. Copy the desired fingerprint. This fingerprint will be used when re-registering clients.

Similarly, you can get fingerprint of external CA certificates.

## Managing usage of certificate authority (CA)

You can manage the usage of CA on individual CipherTrust Manager domains. The CipherTrust Manager supports **user** and **client authentication**.

- **Client authentication:** If enabled, the certificates signed by the CA can be used for client authentication in that domain.
- **User authentication:** If enabled, the certificates signed by the CA can be used for user authentication in that domain.

If authentication is disabled, the CipherTrust Manager domain does not trust the CA for respective authentication mechanisms, and the certificates are rejected even if valid.

## Updating Usage of a Local CA in Web Console UI

1. Login to the root `domain` as an Application Administrator in the Admins `group`, such as `admin`.
2. Navigate to **CA > Local**.
3. In the **Local Certificate Authorities** table, find the Local CA.
4. Click the corresponding overflow icon ( `⋮` ) to view usage change options. Available usage change operations could be:
  - **Disable Client Auth** - Click this to disable client authentication
  - **Disable User Auth** - Click this to disable user authentication
  - **Enable Client Auth** - Click this to enable client authentication
  - **Enable User Auth** - Click this to enable user authentication

The usage change is applied immediately.

## Updating Usage of a Local CA in ksctl CLI

To update usage of a local CA, run:

### Syntax

```
ksctl ca locals update --id <ca-identifier> --allow-user-authentication <true|false> --allow-client-authentication <true|false>
```

### Example request

```
ksctl ca locals update --id e66d047a-2f67-48bf-bcac-862ac773e12a --allow-user-authentication true --allow-client-authentication false
```

### Example response

```
{
 "id": "09e3feb5-7c37-4345-9f05-14a95eb4acd9",
 "uri": "kylo:kylo:naboo:localca:09e3feb5-7c37-4345-9f05-14a95eb4acd9",
 "account": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2022-01-04T16:55:08.995526Z",
 "updatedAt": "2022-01-05T07:31:10.468173266Z",
 "name": "admin",
 "state": "active",
 "csr": "",
 "cert": "-----BEGIN CERTIFICATE-----\nMIIBoTCCASegAwIBAgIQLyw13lVuY5/+BbOYhPfR\npJAKBggqhkjOPQDDAzASMRAw\nDgYDVQQDEwdhZG1pbkNBMB4XD\nTIyMDEwMzE2NTUxMloXD\nTIzMDExMzE2NTUxMlo\nEjEQA4GA1UEAxMHYWRtaW5DQTB2MBAGByqGSM49AgEGBSuBBAAiA2IABMhbWYui\n\nU0/VVZU0bsD6FWt3xGaBWaGrC2BS6EH+YcacosTm0SMWJSYHhN8Yqx\nF8eMmTF1y\n\nnp7tTSRXo89xYqDZK/\nwMmOjv5511yhwV+82o8d1y2Q9obkhgXb39JaxLAlaNCMEAw\n\nDgYDVROPAQH/BAQDAgEGMA8GA1UdEwE\nB/wQFMAMBaf8wHQYDVR0OB\nBYEFIIIBQfXX\n\nnu5bdG1Ojynb7aL4Leq81MAoGCCqGSM49BAMDA2gAMGUCMQ\nD+aEg5pq3VM78Pp5F0\n\nnJEurbY23XoKwixe23IKUhYBTfdpvaDUZGMtV6c7zIfUa2mMCMBN4Nkc6qvalc\nHxo\n\nnFbsmjX42K8fukNqzm39w8vM4o+lhnb5J4bRyghbD/Ej67F6Myw==\n\n-----END CERTIFICATE\n\nE-----\n",
 "serialNumber": "62703269446467328287939974236420428198",
 "subject": "/CN=adminCA",
 "issuer": "/CN=adminCA",
 "notBefore": "2022-01-03T16:55:12Z",
 "notAfter": "2023-01-03T16:55:12Z",
 "sha1Fingerprint": "B3673DFB030A894EF71A71C0C382E78455C1DA84",
 "sha256Fingerprint": "0242266C65ECA9BB893A75EC425A107CCC92C0BEED8224AD87A3A64E9E077739",
 "sha512Fingerprint": "4B20E210A6FE04D5A086346E809076A17C47873D29F1DC071308121BAF2D2AD9C516726E48364281A15224CFA0365B8AFED34B61D8C109DEDBDF4AB365BBF744",
 "purpose": {
 "client_authentication": "Disabled",
 "user_authentication": "Enabled"
 }
}
```

## Updating Usage of External CA in Web Console UI

1. Login to the root [domain](#) as an Application Administrator in the Admins [group](#), such as `admin`.
2. Navigate to **CA > External**.
3. In the **External Certificates Authorities** table, find the external CA.
4. Click the corresponding overflow icon (  ) to view usage change options. Available usage change

operations could be:

- **Disable Client Auth** - Click this to disable client authentication
- **Disable User Auth** - Click this to disable user authentication
- **Enable Client Auth** - Click this to enable client authentication
- **Enable User Auth** - Click this to enable user authentication

The usage change is applied immediately.

## Updating Usage of an External CA in ksctl CLI

To update usage of an external CA, run:

### Syntax

```
ksctl ca externals update --id <ca-identifier> --allow-user-authentication <true|false> --allow-client-authentication <true|false>
```

### Example request

```
ksctl ca externals update --id 5cb55f29-2749-4960-a912-98aeff6accda --allow-user-authentication true --allow-client-authentication false
```

### Example response

```
{
 "id": "f029ff14-9f27-4015-8609-23f41b6de898",
 "uri": "kylo:kylo:naboo:external_ca:f029ff14-9f27-4015-8609-23f41b6de898",
 "account": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2022-01-04T16:57:34.723799Z",
 "updatedAt": "2022-01-05T07:32:10.243282197Z",
 "name": "1234",
 "cert": "-----BEGIN CERTIFICATE-----\nMIIBoTCCASegAwIBAgIQLyw13lVuY5/+BbOYhPfR\npJAKBggqhkJOPQQDAzASMRAw\nDgYDVQQDEwdhZG1pbkNBMB4XD\nTIyMDEwMzE2MDUxMloXD\nTIyMDEwMzE2MDUxMlo\nEjEQMA4GA1UEAxMHYWRtaW5D\nQTB2MBAGByqGSM49AgEGBSu\nBBAAiA2IABMhbWYui\nneU\n0/VVZU0bsD6FWt3xGaBWaGrC2BS6EH+YcacosTm0SMWJSYHhN8Yqx\nF8eMmTF1y\nnp7tTSRXo89xYqDZK/\nwMmOjv5511yhWV+82o8dly2Q9obkhgXb39JaxLAlaNCMEAw\nDgYDVDR0PAQH/BAQDAgEGMA8GA1UdEwE\nB/wQFMAMBAf8wHQYDVR0OB\nBYEFIIIBQfXX\nnu5bdG1Ojynb7aL4Leq81MAoGCCqGSM49BAMDA2gAMGUCMQ\nD+aEg5pq3VM78Pp5F0\nnJEurbY23XoKwixe23IKUhYBTfdpvaDUZGMtV6c7zIfUa2mMCMBN4Nkc6qvalc\nHxo\nnFbsmjX42K8fukNqzm39w8vM4o+lhnb5J4bRyghbD/Ej67F6Myw==\n\n-----END CERTIFICATE\nE-----",
 "purpose": {
 "client_authentication": "Disabled",
 "user_authentication": "Enabled"
 },
 "serialNumber": "62703269446467328287939974236420428198",
 "subject": "/CN=adminCA",
 "issuer": "/CN=adminCA",
 "notBefore": "2022-01-03T16:55:12Z",
 "notAfter": "2023-01-03T16:55:12Z",
}
```

```

 "sha1Fingerprint": "B3673DFB030A894EF71A71C0C382E78455C1DA84",
 "sha256Fingerprint": "0242266C65ECA9BB893A75EC425A107CCC92C0BEED8224AD87A3A64
E9E077739",
 "sha512Fingerprint": "4B20E210A6FE04D5A086346E809076A17C47873D29F1DC071308121
BAF2D2AD9C516726E48364281A15224CFA0365B8AFED34B61D8C109DEDBDF4AB365BBF744"
}

```

## Certificate format in REST-API and UI

If you are uploading certificates using the REST API, certificates are encoded in a JSON string and have `\n` characters to indicate line endings. CipherTrust Manager's UI web console does not have these characters, so it's easiest to remain in the same interface for certificate operations, and/or pass in and export certificates using a file instead of a pasted string wherever possible. If you are copy-pasting certificate strings between the UI and other interfaces, you must re-encode the certificate strings.

- To change PEM-encoded strings to JSON, use `echo $v | jq -R --slurp` where `$v` is a variable for the string. This is needed to format certificates in the UI to REST API format.
- To change JSON encoded strings to PEM, use `echo $v | /usr/local/bin/jq -r`, where `$v` is a variable for the string. This is needed to format certificates from REST API to the UI format.

## Adding Web Certificate for FQDN Connectivity

To add a web certificate for FQDN connectivity:

1. View the parsed server certificate in use.
  - a. Log on to the CipherTrust Manager as an administrator.
  - b. Go to **Admin Settings > Interfaces**.
  - c. On the Interfaces page, click the overflow icon next to the **web** interface.
  - d. Click **Certificate Options....**
  - e. On the Interface Certificate Options on 'web' dialog box, select **View Certificate** and click **Ok**.
  - f. On the View WEB Certificate dialog box, copy the value of the **subject** Property. For example, **C=US, ST=TX, L=Austin, O=Thales, CN=web.ciphertrustmanager.local**.
  - g. Close the dialog box.
2. Issue a new certificate.
  - a. Go to **CA > Local**, and click the Name link of the local CA. The Certificate issued by <local-ca-name> screen is displayed.
  - b. Click **+ Issue Certificate**.
  - c. Paste the subject value you copied [above](#) in the **Common Name** field.

- 
- d. Select **RSA** as **Algorithm** and **4096** as **Size**.
  - e. Under Subject Alternative Names, in the **DNS Names (comma separated)** field, specify the FQDN of your CipherTrust Manager instance.
  - f. In the **Name (comma separated)** field, enter **web**.
  - g. Click **Issue Certificate**, and click **save private key**. The private key, `key.pem` is downloaded.
  - h. Change the **Certificate Purpose** to **server**, **Duration in days** to **365** days, and click **Issue Certificate**.

The certificate is successfully issued. You are returned to the Certificate issued by <local-ca-name> screen.

3. Download the certificate. Click the overflow icon corresponding to the newly issued certificate and click **Download**. The Certificate, `certificate.pem`, is downloaded.
4. Concatenate the **private key** and the certificate downloaded above. For example, run `cat key.pem certificate.pem > web_certificate.pem`.
5. Upload the above created `web_certificate.pem`.
  - a. Go to **Admin Settings > Interfaces**.
  - b. On the Interfaces page, click the overflow icon next to the **web** interface.
  - c. Click **Certificate Options....** The Interface Certificate Options wizard is displayed.
  - d. Select **Upload New Certificate** and click **Ok**. The Upload Certificate dialog box is displayed.
  - e. Click **Choose Certificate File** and upload the above created `web_certificate.pem` file.
  - f. Click **Upload Certificate**. The `web_certificate.pem` file is uploaded.
  - g. Click **Close**.
6. Go to **Admin Settings > Services**. The Services page shows the available services.
7. Scroll down to the **web** service.
8. Click **Restart** next to the web service.
9. Restart the CipherTrust Manager application. Click **System Restart** at the bottom of the page. The system will be unavailable during restart.
10. Download the caBundle (Local CA in CipherTrust Manager) for your CipherTrust Manager server certificate that will need to be uploaded to OCI later.

# System Info

The `system info` option retrieves the basic information of the CipherTrust Manager. It also allows you to set the name of the system.

| Attribute             | Description                                                                                                                   |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------|
| name                  | user-friendly name of the system.                                                                                             |
| version               | version details.                                                                                                              |
| version_suffix        | applicable only for the Long Term Support (LTS) releases. Displays the LTS release value as a suffix to the version.          |
| model                 | model number.                                                                                                                 |
| vendor                | vendor information.                                                                                                           |
| crypto_version        | FIPS version of the FIPS micro services of the CipherTrust manager.                                                           |
| uptime                | duration of time the system has been working and available.                                                                   |
| system_time           | date-time of the system.                                                                                                      |
| chassis_serial_number | the chassis serial number of the system. This field is applicable only for the CipherTrust Manager <i>physical</i> appliance. |

## Getting system info

To fetch the system information:

### Request example 1 (virtual CipherTrust Manager instance)

```
ksctl system info get
```

### Response

```
{
 "name": "",
 "version": "2.12.0-beta5+8905",
 "version_suffix": "",
 "model": "CipherTrust Manager k170v",
 "vendor": "Thales Group",
 "crypto_version": "1.7.0",
 "uptime": "0 days, 3 hours, 58 minutes",
 "system_time": "2023-02-28T05:58:56Z"
}
```

### Request example 2 (physical CipherTrust Manager appliance)

```
ksctl system info get
```

## Response

```
{
 "name": "",
 "version": "2.12.0-beta3+8382",
 "version_suffix": "",
 "model": "CipherTrust Manager k570",
 "vendor": "Thales Group",
 "crypto_version": "Unknown",
 "uptime": "19 days, 19 hours, 14 minutes",
 "system_time": "2023-02-28T07:11:27Z",
 "chassis_serial_number": "616-20061700100036"
}
```

## Change the Instance Name

Application Administrators, such as the `admin` user can change the instance name. This setting is only accessible in the root domain.

### Change the Instance Name in UI

1. Login to the root domain as an Application Administrator.
2. Navigate to **Admin Settings > Properties**.
3. In **Instance Properties**, enter a name in the **Name for this instance** field.
4. Click **Set Name** to finalize the change.

### Change the Instance Name in ksctl CLI

To set the user-friendly name for the system:

#### Syntax

```
ksctl system info set --name <user-friendly name for the system>
```

#### Example Request

```
ksctl system info set --name "virtual-box"
```

#### Example Response

```
{
 "name": "virtual-box",
 "version": "2.12.0-beta3+8382",
 "model": "CipherTrust Manager k570",
 "vendor": "Thales Group",
 "crypto_version": "Unknown",
 "uptime": "",

```



```
"system_time": ""
}
```

## System Properties

The CipherTrust Manager allows [Application Administrators](#), such as the `admin` user, to manage user configurable system properties. These settings are only accessible in the root [domain](#).

## Enable and Disable Activity Logs

You can enable or disable [activity logs](#) for [NAE and KMIP interfaces](#), to include them in debug log downloads. They are not enabled by default. When enabled, they are included in the downloadable LTSV debug log file, and can be forwarded to configured [log forwarders](#).

1. Login to the root domain as an Application Administrator.
2. Navigate to **Admin Settings > Properties**.
3. In **LOGS PROPERTIES**, select the **KMIP Activity Logs** toggle button to enable/disable KMIP activity logs.
4. In **LOGS PROPERTIES**, select the **NAE Crypto Activity Logs** toggle button to enable/disable NAE crypto activity logs.

## Set a Load Balancer Address

You can configure a load balancer address, so that CTE agents can communicate with CipherTrust Manager cluster nodes which are behind a load balancer.

1. Login to the root domain as an Application Administrator.
2. Navigate to **Admin Settings > Properties**.
3. In **Load Balancer Address**, add an IP address or hostname for the load balancer to the **Value** field.
4. Click **Update** to update it.

## Enable or Disable Data Key Caching for HSM-Anchored Domains

This caching can improve cryptographic and key export performance for [HSM-anchored domains](#).

1. Login to the root domain as an Application Administrator.
2. Navigate to **Admin Settings > Properties**.
3. In **HSM-ANCHORED DOMAINS DATA KEY MATERIAL CACHING**, select the **Enable Key Caching** toggle.
4. Do a system restart. For clustered CipherTrust Managers, restart every cluster node. This is required to fully apply the change.

## Set Cluster Certificate Renewal Threshold

This setting controls the number of days prior to certificate expiry that CipherTrust Manager auto-renews an internal certificate that is used for database access.

1. Login to the root domain as an Application Administrator.
2. Navigate to **Admin Settings > Properties**.
3. In **CLUSTER CERTIFICATE AUTO RENEW THRESHOLD**, in the **Value** field, enter an integer value, 30 or higher.
4. Click **Update** to apply the change.

## Properties Configurable from ksctl or REST API

| Property Name                           | Default Value | Auto Replication<br>(no need to update<br>on other cluster<br>nodes) | Service Restart<br>Required on all<br>Cluster Nodes | Description                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------|---------------|----------------------------------------------------------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ALLOW_UNKNOWN_FIELDS                    | False         | Yes                                                                  | Yes                                                 | Allows unknown fields in the JSON body from the API calls. If you set this property to <code>true</code> , restart is required for changes to take effect.                                                                                                                                                                                                                                                                      |
| ALLOW_USER_IMPERSONATION_ACR_OSS_DOMAIN | False         | Yes                                                                  | No                                                  | User impersonation is, by default, limited to impersonate users of the clients' domain only i.e. created users (users created in clients' domain) or assigned users (users created in clients' domain and assigned to some other domain). With this property enabled, clients belonging to the root domain can impersonate a user created inside a sub-domain only. The property is applicable to NAE and KMIP interfaces only. |
| CERT_REV_CHECK                          | 5sec          | Yes                                                                  | No                                                  | Configures the                                                                                                                                                                                                                                                                                                                                                                                                                  |

| Property Name                                       | Default Value | Auto Replication<br>(no need to update<br>on other cluster<br>nodes) | Service Restart<br>Required on all<br>Cluster Nodes | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------------------|---------------|----------------------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| K_TIMEOUT                                           |               |                                                                      |                                                     | revocation check request timeout for Web (cert user login) and NAE/KMIP clients. The timeout value must be in an integer from 0-600secs, where 0 refers to no timeout. This timeout is for each request. For example if a certificate contains 3 URLs for the calssuer and CRL/OCSP, and each URL gets failed in 5sec; then it will take 15sec to complete the authentication call. Additionally, this setting also determines the timeout for fetching a CRL while executing the <a href="#">Sync CRL schedule</a> . |
| CLUSTER<br>CERTIFICATE<br>AUTO RENEWAL<br>THRESHOLD | 30            | Yes                                                                  | No                                                  | Configures the number of days prior to certificate expiry that CipherTrust Manager auto-renews an internal certificate that is used for database access. Valid values are 30 or more, integers only.                                                                                                                                                                                                                                                                                                                  |
| ENABLE_CERT_RE<br>V_CHECK                           | True          | Yes                                                                  | No                                                  | Checks the client certificates on the NAE/KMIP interface for revocation status. This property is not applicable for user certificate login. For user certificate login, the CRL revocation status is always checked.                                                                                                                                                                                                                                                                                                  |
| ENABLE_KEY_CAC                                      | False         | Yes                                                                  | Yes                                                 | Allows caching on                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

| Property Name                   | Default Value | Auto Replication<br>(no need to update<br>on other cluster<br>nodes) | Service Restart<br>Required on all<br>Cluster Nodes | Description                                                                                                                                                                                                                                                            |
|---------------------------------|---------------|----------------------------------------------------------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HE                              |               |                                                                      |                                                     | CipherTrust Manager of key materials for HSM-anchored domains.                                                                                                                                                                                                         |
| ENABLE_KMIP_ACTIVITY_LOGS       | False         | Yes                                                                  | No                                                  | Enables the activity logging for KMIP operations.                                                                                                                                                                                                                      |
| ENABLE_NAE_ACTIVITY_LOGS        | False         | Yes                                                                  | No                                                  | Enables the activity logging for NAE operations.                                                                                                                                                                                                                       |
| ENABLE_RECORDS_DB_STORE         | False         | Yes                                                                  | No                                                  | Stores audit records in the database. If you set this property to <code>false</code> , it deletes the audit records. If syslog is added, it continues to push audit records to the syslog server.                                                                      |
| ENFORCE_NAE_CLIENT_VALIDATION   | false         | Yes                                                                  | No                                                  | Enforces client validation on the NAE. The NAE-KMIP service <i>do not</i> require system restart to reflect any changes made to the <code>ENFORCE_NAE_CLIENT_VALIDATION</code> property. Setting this property on one node bring it into effect for an entire cluster. |
| ENFORCE_NAE_CLIENT_REGISTRATION | False         | Yes                                                                  | No                                                  | Enforces registered NAE clients to perform the NAE operations.                                                                                                                                                                                                         |
| LOAD_BALANCER_ADDRESS           |               | Yes                                                                  | No                                                  | IP address/URL/Hostname of load balancer configured with CipherTrust Manager. This is to allow CTE agent communication.                                                                                                                                                |
| MAXIMUM_REFRESH_TOKEN_LIFETIME  | 5min          | Yes                                                                  | No                                                  | Describes the maximum lifetime of a refresh token. This                                                                                                                                                                                                                |

| Property Name           | Default Value | Auto Replication<br>(no need to update<br>on other cluster<br>nodes) | Service Restart<br>Required on all<br>Cluster Nodes | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------|---------------|----------------------------------------------------------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         |               |                                                                      |                                                     | can be an empty string (infinite lifetime) or a sequence of integers with a time unit as a suffix. The valid time units are <code>m</code> and <code>h</code> . For example, <code>300m</code> , <code>3h</code> , or <code>2h45m</code> . The value of the maximum lifetime of a refresh token should be at least 5min.                                                                                                                                                                                                                                                                                                                                                                                                     |
| UI_IDLE_SESSION_TIMEOUT | 10min         | Yes                                                                  | No                                                  | <p>Determines the session timeout for the UI when it remains idle. The UI Session timeout can be set as an empty string (which defaults to 10 minutes) or as a numerical value followed by a unit suffix, such as <code>7m</code>, <code>30m</code>, or <code>60m</code>. The only valid time unit is <code>m</code> for minutes. The <code>UI_IDLE_SESSION_TIMEOUT</code> must be set between 5 and 60 minutes and should not exceed the value of <code>MAXIMUM_REFRESH_TOKEN_LIFETIME</code>. If you try to exceed it, an error will occur stating, UI idle session timeout must be less than or equal to Maximum refresh token lifetime.</p> <p>There may also be cases, such as during a reset or upgrade, where the</p> |

| Property Name                  | Default Value | Auto Replication<br>(no need to update<br>on other cluster<br>nodes) | Service Restart<br>Required on all<br>Cluster Nodes | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------|---------------|----------------------------------------------------------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                |               |                                                                      |                                                     | <p>UI_IDLE_SESSION_TIMEOUT exceeds the MAXIMUM_REFRESH_TOKEN_LIFETIME. In such instances, the effective value will be the smaller of the two, and a warning will be displayed:</p> <p>Effective value is Maximum lifetime of a refresh token, because Maximum lifetime of a refresh token is less than UI Session Timeout.</p>                                                                                                                                                                                            |
| DISABLE_TLS_SESSION_RESUMPTION | False         | Yes                                                                  | Yes                                                 | <p>Disables the session ticket and PSK (resumption) support on NAE and KMIP interfaces. Set it to true to enable the property. Clients using OpenSSL version older than 1.0.2 may encounter 'ssl handshake error' on NAE and KMIP when making concurrent connections using TLS session resumption at client side. To mitigate this error, it is recommended for clients using <b>OpenSSL version 1.0.2 or lower to upgrade to version 1.0.2b or higher</b>. Alternatively, you can disable the TLS session resumption</p> |

| Property Name              | Default Value | Auto Replication<br>(no need to update<br>on other cluster<br>nodes) | Service Restart<br>Required on all<br>Cluster Nodes | Description                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------|---------------|----------------------------------------------------------------------|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                            |               |                                                                      |                                                     | on NAE and KMIP interfaces using the <code>DISABLE_TLS_SESSION_RESUMPTION</code> system property.                                                                                                                                                                                                                                                                                |
| KEY_CACHE_EXPIRES_DURATION | 2             | Yes                                                                  | Yes                                                 | Defines the NAE-KMIP key cache expiry duration in seconds. It's value ranges between 0 to 86400 seconds(24 hrs). If you set the value to 0, it disables the cache. Using a large value for <code>KEY_CACHE_EXPIRES_DURATION</code> improves the cryptographic and export operations' performance, but disables the Key Usage/Export audit events for the duration key is cached. |
| KEY_STATES_METRIC_INTERVAL | 3600s (1h)    | Yes                                                                  | Yes                                                 | Defines the interval for retrieving key states from the database. It's value ranges from 0-63072000 seconds. It displays the count of keys based on their state and algorithm on the Prometheus endpoint and Grafana dashboard.                                                                                                                                                  |
| PASSWORD_HASH_ITERATIONS   | 10000         | Yes                                                                  | No                                                  | Defines how many times a hashing algorithm is applied repeatedly to a password. This process enhances the password security by increasing its resistance to brute-                                                                                                                                                                                                               |

| Property Name              | Default Value | Auto Replication<br>(no need to update<br>on other cluster<br>nodes) | Service Restart<br>Required on all<br>Cluster Nodes | Description                                                                                                                                                                                                                                                                                                                                         |
|----------------------------|---------------|----------------------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                            |               |                                                                      |                                                     | force attacks, as it requires more computational effort. The default value is 10000. The permissible values are 10000 and 5000.                                                                                                                                                                                                                     |
| ENABLE_REST_CRYPTO_RECORDS | False         | Yes                                                                  | No                                                  | <p>Enables logging of REST crypto operations in audit records.</p> <p>You can also enable this property from UI using the Audit records for REST crypto operations toggle button. <b>(Admin Settings &gt; Properties)</b></p> <p>Enabling the ENABLE_REST_CRYPTO_RECORDS property may lead to a performance drop during bulk crypto operations.</p> |

## Managing user configurable system properties using ksctl

The following operations can be performed:

- Get details of a property
- List all properties
- Modify a property
- Reset a property

### Getting details of properties

To get details of a property, run:

#### Syntax



```
ksctl properties get --name <name of the system configuration>
```

### Example

```
ksctl properties get --name ENABLE_RECORDS_DB_STORE
```

## Getting list of properties

To list all the configurable system properties, run:

### Syntax

```
ksctl properties list
```

## Updating properties

To modify the value of a property, run:

### Syntax

```
ksctl properties modify --name <property name> --value <value to be set>
```

### Example 1

```
ksctl properties modify --name ENABLE_RECORDS_DB_STORE --value false
```

### Example 2

```
ksctl properties modify --name CERT_REV_CHECK_TIMEOUT --value 10
```

Changes doesn't reflect immediately. It may take up to 60 seconds on the current node as well as on the clustered nodes.

## Resetting properties

To reset the value of a property to default, run:

### Syntax

```
ksctl properties reset --name <property name>
```

### Example

```
ksctl properties reset --name ENABLE_RECORDS_DB_STORE
```

# Network Diagnostics

Users in the [admin](#) or [System Admins group](#) can run some network diagnostics to determine availability and latency of internet connections to external hosts. These network diagnostics emulate ping, netcat (checkport), traceroute, and nslookup. As well, you can return information about a network interface.

The following network diagnostics are available in the [CLI](#) and the [API](#):

- network interfaces list
- ping
- traceroute
- checkport
- lookup

## Note

[Connection Manager](#) also allows a basic connection test for new and existing connections, which indicates availability.

The `ksadmin` can manage network interfaces for some deployments using the [nmcli](#) tool through an SSH session to CipherTrust Manager. Use the cloud provider's tools to manage network interfaces for Virtual CipherTrust Manager appliances deployed on public clouds.

## network interfaces list

You can list all the network interfaces on the CipherTrust Manager. The interface `name` is returned, which can be used in the other network diagnostic commands. The method, IP address, gateway, and netmask IPv4 and IPv6 values are returned. The dns is returned for IPv4 only.

## network interfaces list Command Syntax

```
ksctl network interfaces list
```

## network interfaces list Example Response

```
{
 "skip": 0,
 "limit": 0,
 "total": 1,
 "resources": [
 {
 "name": "ens5",
 "inet": {
```

```

 "method": "dhcp",
 "ip": "172.30.1.79",
 "gateway": "172.30.1.1",
 "netmask": "255.255.255.0",
 "dns": [
 "172.30.0.2"
]
 },
 "inet6": {
 "method": "auto",
 "ip": "2600:1f18:4ad:3400:1d5a:c254:8197:e7d
9",
 "gateway": "fe80::c91:46ff:fe5c:d40d",
 "netmask": 128
 }
}
]
}

```

## ping

Ping a remote host. The `destination` is required as a domain name or IP address.

The response contains the IP address of the destination, number of sent, received, and lost packets, as well as minimum, maximum, and average round trip time (RTT).

You can optionally specify:

- `count` - The number of packets to send.
- `interface` - The CipherTrust Manager network interface to communicate on. Available interfaces can be found with `network interfaces list` command.
- `ipv4` - Use IPv4 only for communication. Cannot be used with `ipv6`.
- `ipv6` - Use IPv6 only for communication. Cannot be used with `ipv4`.
- `timeout-secs` - Time to wait for a response from the host system in seconds.

## ping Command Syntax

```
ksctl network ping --destination <IP_address_or_domain_name> [--interface <m_interface>] [--count <integer>] [--timeout-secs <integer>] [--ipv6|--ipv4]
```

## ping Example Response

```

{
 "ip_address": "2607:f8b0:4004:c09::68",
 "packets_sent": 1,
 "packets_received": 1,
 "packets_lost": 0,

```

```

 "min_rtt": 2.369,
 "max_rtt": 2.369,
 "avg_rtt": 2.369
 }

```

## traceroute

Return the traceroute to a specified destination. The `destination` is required as a domain name or IP address.

The response contains each hop number, the domain name and IP address of each hop, and the RTT values for probe for each hop.

You can optionally specify:

- `first` - Start from the `first_ttl` hop.
- `interface` - The CipherTrust Manager network interface to communicate on. Available interfaces can be found with `network interfaces list` command.
- `ipv4` - Use IPv4 only for communication. Cannot be used with `ipv6`.
- `ipv6` - Use IPv6 only for communication. Cannot be used with `ipv4`.
- `max-hops` - Maximum number of hops to search for in target.
- `port` - Port to access.
- `queries` - Set the number of probes per each hop.
- `send-wait` - Minimum time interval between probes in milliseconds.
- `tcp` - Use TCP SYN for tracerouting.
- `udp` - Use UDP to particular port for tracerouting.

## traceroute Command Syntax

```

ksctl network traceroute --destination <IP_address_or_domain_name> [--first
<integer>] [--interface <cm_interface>] [--ipv6|--ipv4] [--max-hop <number_of_hop
s>] [--port <port_number>] [--queries <number_of_probes_per_hop>] [--send-wait <m
inimum_time_interval_between_probes_in_ms>] [--tcp] [--udp]

```

## traceroute Example Response

```

{
 "response": [
 "traceroute to www.google.com (172.253.122.147), 5 hops max,
 60 byte packets",
 " 1 244.5.2.217 (244.5.2.217) 2.513 ms ec2-3-236-63-27.comp

```

```

ute-1.amazonaws.com (3.236.63.27) 7.604 ms ec2-3-236-62-5.compute-1.amazonaws.co
m (3.236.62.5) 4.526 ms",
 " 2 240.0.56.66 (240.0.56.66) 0.252 ms 240.0.56.98 (240.0.5
6.98) 0.331 ms 240.0.224.64 (240.0.224.64) 0.284 ms",
 " 3 240.0.224.80 (240.0.224.80) 0.299 ms 240.3.84.90 (24
0.3.84.90) 0.246 ms 240.3.84.81 (240.3.84.81) 0.267 ms",
 " 4 240.0.56.51 (240.0.56.51) 0.246 ms 240.3.84.52 (240.3.8
4.52) 0.219 ms 240.0.56.29 (240.0.56.29) 0.261 ms",
 " 5 240.0.224.34 (240.0.224.34) 0.275 ms 240.0.36.12 (24
0.0.36.12) 0.358 ms 240.0.40.13 (240.0.40.13) 0.784 ms"
]
}

```

## checkport

Attempt a connection to a specified port on the remote host.

The required flags are:

- `destination` - remote host destination provided as a domain name or IP address.
- `port` - remote host port number

You can optionally specify:

- `interface` - The CipherTrust Manager network interface to communicate on. Available interfaces can be found with `network interfaces list` command.
- `ipv4` - Use IPv4 only for communication. Cannot be used with `ipv6`.
- `ipv6` - Use IPv6 only for communication. Cannot be used with `ipv4`.
- `timeout-secs` - Time to wait for a response from the host system in seconds.

## checkport Command Syntax

```

ksctl network checkport --destination <destination_IP_or_domain_name> --port
<remote_host_port_to_access> [--interface <cm_interface>] [--ipv6|--ipv4] [--time
out-secs <time_to_wait_for_response>]

```

## checkport Example Response

```

{
 "response": "connection successful"
}

```

## lookup

Look up a specific IP address or hostname. The IP address can be IPv4 or IPv6. When looking up a host name, a

list of IP addresses associated with the host name are returned. When looking up an IP address, a list of host names associated with the IP address are returned.

The `target` IP address or hostname is required.

## lookup Command Syntax

```
ksctl network lookup --target <IP_address_or_hostname>
```

## lookup Example Response

```
"response": {
 "hosts": [
 "www.abc.com",
 "www.def.com"
]
}
```

# Integrating Logging with Splunk App

This section describes how to configure the CipherTrust Manager and Thales Security Intelligence app (a Splunk app) for the CM Dashboard of the app.

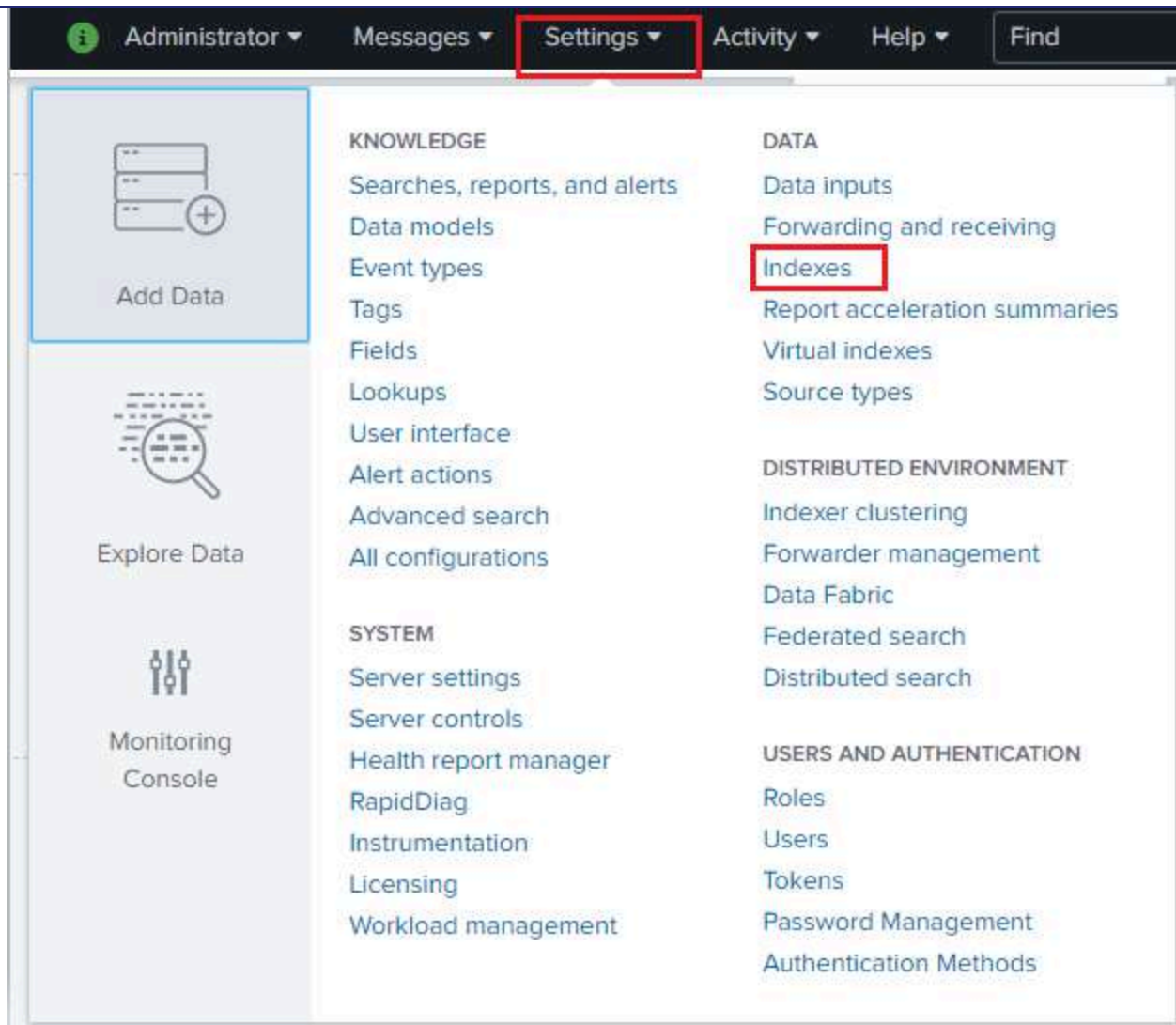
## App Configuration for CM Dashboard

The Thales Security Intelligence app is written for flexibility, however, this means, more configuration steps are required for the app to work correctly with the log data (for example, Splunk indexes and input).

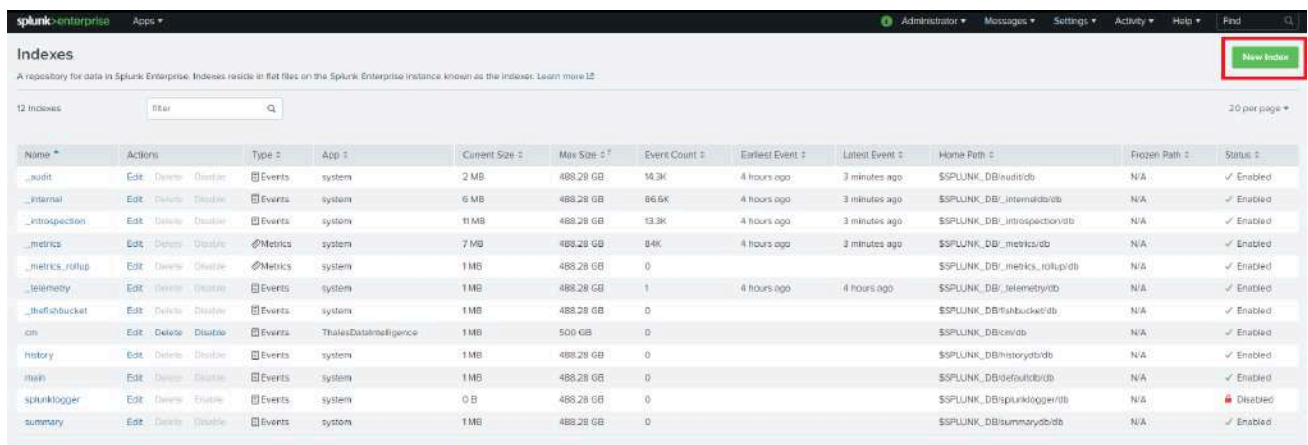
## Create the CipherTrust Manager Splunk Index

The Splunk index is for the general CipherTrust Manager log data. Perform the following steps to create indexes to specifically hold the CipherTrust Manager log data from your CipherTrust Manager appliances, both virtual and physical.

1. Log on to your Splunk server.
2. From the top menu, click **Settings > Indexes**.



3. At the top right, click the **New Index** button.



4. In the **New Index** dialog box, specify the **Index Name** (for example, `cm`).

5. From the **App** drop-down list, select **Thales Security Intelligence**.

New Index

General Settings

Index Name

cm

Set index name (e.g., INDEX\_NAME). Search using index=INDEX\_NAME.

Index Data Type

Events

Metrics

The type of data to store (event-based or metrics).

Home Path

optional

Hot/warm db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/db).

Cold Path

optional

Cold db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/colddb).

Thawed Path

optional

Thawed/resurrected db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/thaweddb).

Data Integrity Check

Enable

Disable

Enable this if you want Splunk to compute hashes on every slice of your data for the purpose of data integrity.

Max Size of Entire Index

500

GB ▼

Maximum target size of entire index.

Max Size of Hot/Warm/Cold Bucket

auto

GB ▼

Maximum target size of buckets. Enter 'auto\_high\_volume' for high-volume indexes.

Frozen Path

optional

Frozen bucket archive path. Set this if you want Splunk to automatically archive frozen buckets.

App

Thales Security Intelligence ▼

Storage Optimization

Save

Cancel

6. Click **Save**.

The Indexes page is refreshed with the new index in the list.

## Create Splunk Data Input (TCP/UDP)

By default, the Thales Security Intelligence app is not configured to obtain the CipherTrust Manager log data on any port. So, the data input ports must be added.

To add the data input ports:

1. From the **Splunk** menu, click **Settings > Data inputs**.



|                                                                                                         |                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <br>Add Data           | <b>KNOWLEDGE</b> <ul style="list-style-type: none"> <li>Searches, reports, and alerts</li> <li>Data models</li> <li>Event types</li> <li>Tags</li> <li>Fields</li> <li>Lookups</li> <li>User interface</li> <li>Alert actions</li> <li>Advanced search</li> <li>All configurations</li> </ul> | <b>DATA</b> <ul style="list-style-type: none"> <li>Data inputs</li> <li>Forwarding and receiving</li> <li>Indexes</li> <li>Report acceleration summaries</li> <li>Virtual indexes</li> <li>Source types</li> </ul> |
|                                                                                                         | <b>SYSTEM</b> <ul style="list-style-type: none"> <li>Server settings</li> <li>Server controls</li> <li>Health report manager</li> <li>RapidDiag</li> <li>Instrumentation</li> <li>Licensing</li> <li>Workload management</li> </ul>                                                           | <b>DISTRIBUTED ENVIRONMENT</b> <ul style="list-style-type: none"> <li>Indexer clustering</li> <li>Forwarder management</li> <li>Data Fabric</li> <li>Federated search</li> <li>Distributed search</li> </ul>       |
|                                                                                                         |                                                                                                                                                                                                                                                                                               | <b>USERS AND AUTHENTICATION</b> <ul style="list-style-type: none"> <li>Roles</li> <li>Users</li> <li>Tokens</li> <li>Password Management</li> <li>Authentication Methods</li> </ul>                                |
| <br>Explore Data       |                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                    |
| <br>Monitoring Console |                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                    |

- On the **Data inputs** page, click **+ Add new** for the **TCP** or **UDP** Type.

## Local inputs

| Type                                                                                                                                  | Inputs | Actions   |
|---------------------------------------------------------------------------------------------------------------------------------------|--------|-----------|
| <b>Files &amp; Directories</b><br>Index a local file or monitor an entire directory.                                                  | 11     | + Add new |
| <b>HTTP Event Collector</b><br>Receive data over HTTP or HTTPS.                                                                       | 0      | + Add new |
| <b>TCP</b><br>Listen on a TCP port for incoming data, e.g. syslog.                                                                    | 0      | + Add new |
| <b>UDP</b><br>Listen on a UDP port for incoming data, e.g. syslog.                                                                    | 1      | + Add new |
| <b>Scripts</b><br>Run custom scripts to collect or generate more data.                                                                | 9      | + Add new |
| <b>Systemd Journald Input for Splunk</b><br>This is the input that gets data from journald (systemd's logging component) into Splunk. | 0      | + Add new |
| <b>Splunk Secure Gateway</b><br>Initializes the Splunk Secure Gateway application to talk to mobile clients over websockets           | 1      | + Add new |
| <b>Splunk Secure Gateway Mobile Alerts TTL</b><br>Cleans up storage of old mobile alerts                                              | 1      | + Add new |
| <b>Splunk Secure Gateway Deleting Expired Tokens</b><br>Delete expired or invalid tokens created by Secure Gateway from Splunk        | 1      | + Add new |
| <b>Splunk Secure Gateway Role Based Notification Manager</b><br>Used for sending mobile alerts to users by role                       | 1      | + Add new |

3. Add the desired **Port** (for example, UDP/5514 or TCP/6514) and click **Next >**.

**Add Data**

Select Source
Input Settings
Review
Done

< Back
Next >

**Files & Directories**  
Upload a file, index a local file, or monitor an entire directory.

**HTTP Event Collector**  
Configure tokens that clients can use to send data over HTTP or HTTPS.

**TCP / UDP**  
Configure the Splunk platform to listen on a network port.

**Scripts**  
Get data from any API, service, or database with a script.

**Systemd Journald Input for Splunk**  
This is the input that gets data from journald (systemd's logging component) into Splunk.

**Splunk Secure Gateway**  
Initializes the Splunk Secure Gateway application to talk to mobile clients over websockets.

**Splunk Secure Gateway Mobile Alerts TTL**  
Cleans up storage of old mobile alerts.

**Splunk Secure Gateway Deleting Expired Tokens**  
Delete expired or invalid tokens created by Secure Gateway from Splunk.

**Splunk Secure Gateway Role Based Notification Manager**  
Used for sending mobile alerts to users by role.

**Splunk Secure Gateway Enable**  
Determine if Splunk Secure Gateway core modular inputs should be enabled.

**Splunk Secure Gateway Metrics Collector**  
Collects metrics for Splunk Secure Gateway.

Configure this instance to listen on any TCP or UDP port to capture data sent over the network (such as syslog). [Learn More](#)

TCP

UDP

Port ?   
Example: 514

Source name override ?   
host:port

Only accept connection from ?   
example: 10.1.2.3, lbadhost.splunk.com, \*.splunk.com

**FAQ**

- > How should I configure the Splunk platform for syslog traffic?
- > What's the difference between receiving data over TCP versus UDP?
- > Can I collect syslog data from Windows systems?
- > What is a source type?

#### 4. Specify the **Input Settings**.

- a. Next to the **Source type** section, click **Select**, then select **cm-st** from the drop-down list.

#### Note

The custom Splunk source type `cm-st` is created when you install the Splunk app.

- b. From the **App Context** drop-down list, select the **Thales Security Intelligence** app.
- c. Select **IP** as the Method.
- d. Select the **Index** created in the steps above.
- e. Click **Review >**.

Add Data

Select Source

Input Settings

Review

Done

< Back

Review >

Source type

The source type is one of the default fields that the Splunk platform assigns to all incoming data. It tells the Splunk platform what kind of data you've got, so that the Splunk platform can format the data intelligently during indexing. And it's a way to categorize your data, so that you can search it easily.

Select

New

cm-st ▼

App context

Application contexts are folders within a Splunk platform instance that contain configurations for a specific use case or domain of data. App contexts improve manageability of input and source type definitions. The Splunk platform loads all app contexts based on precedence rules. [Learn More](#)

App Context

Thales Security Intelligence (ThalesDataIntelligen... ▼)

Host

When the Splunk platform indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. [Learn More](#)

Method ?

IP

DNS

Custom

Index

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later. [Learn More](#)

Index

cm ▼

Create a new index

FAQ

- > How do indexes work?
- > How do I know when to create or use multiple indexes?

- On the **Review** page, review the information. To make any changes, click **Back** and update the details. If the information is correct, click **Submit >**.

**Add Data**

Select Source

Input Settings

Review

Done

< Back

Submit >

## Review

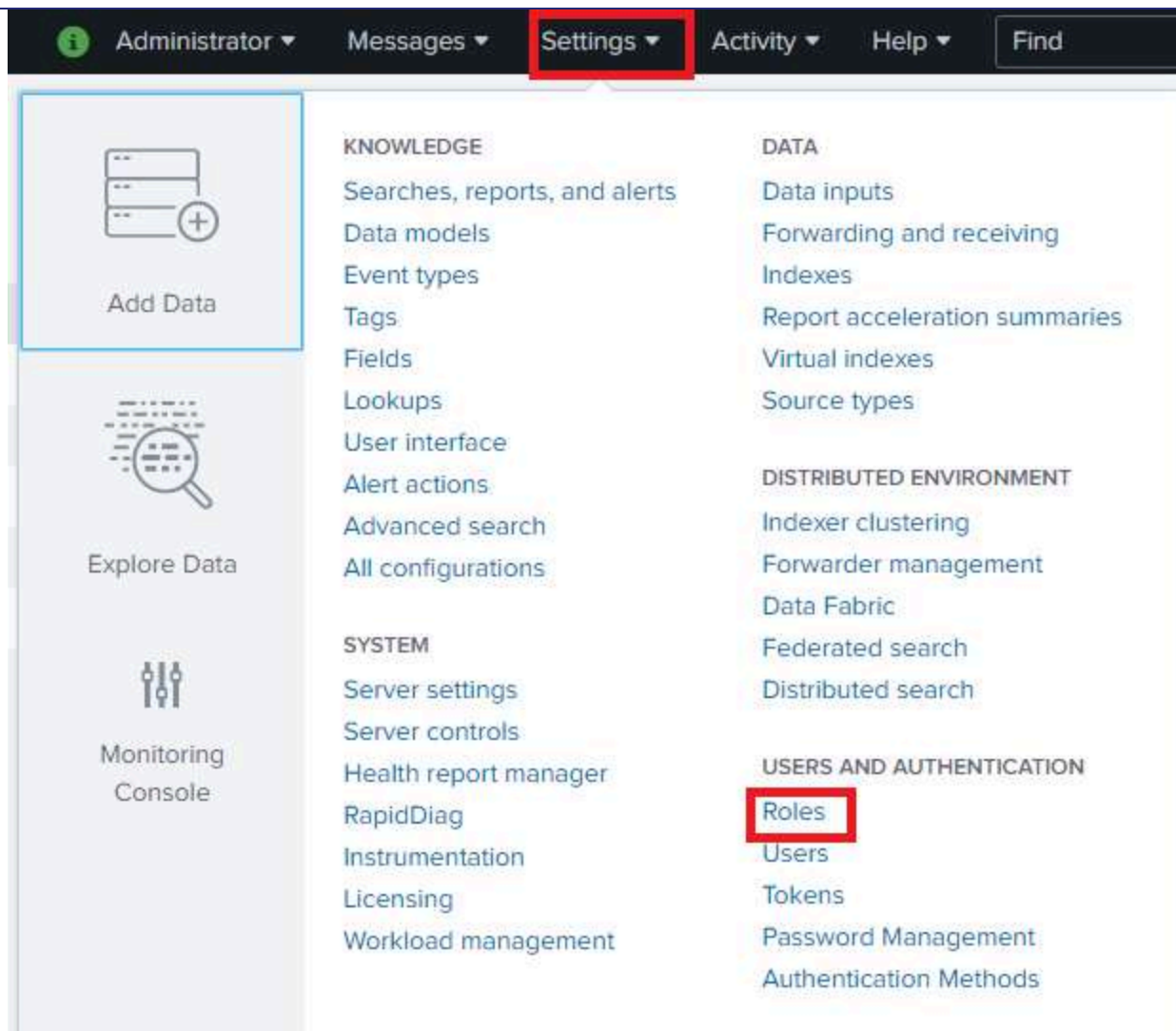
|                            |                                   |
|----------------------------|-----------------------------------|
| Input Type .....           | UDP Port                          |
| Port Number .....          | 5514                              |
| Source name override ..... | N/A                               |
| Restrict to Host .....     | N/A                               |
| Source Type .....          | cm-st                             |
| App Context .....          | ThalesDataIntelligence            |
| Host .....                 | (IP address of the remote server) |
| Index .....                | cm                                |

## Add User Role Access to Indexes

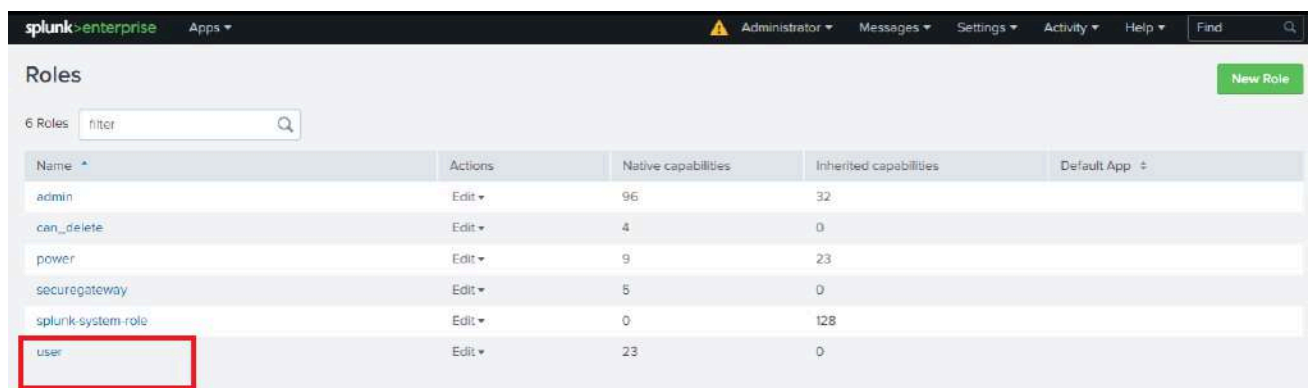
By default, Splunk uses the main or default index when performing searches if no index is specified. You can add the newly created CipherTrust index to the Splunk Role that will have access to view the Thales Security Intelligence App Dashboards.

To add the user role access to indexes:

1. From the **Splunk** menu, click **Settings > Roles**.



- On the **Roles** page, select the role to modify (for example, `user`).



**Note**

The next steps apply to Splunk 8 and higher versions.

3. On the **Edit Role user** page, click the **Indexes** tab.

**Edit Role user** ✕

Name \* ?

1. Inheritance

2. Capabilities

3. Indexes

4. Restrictions

5. Resources

Specify roles from which this role inherits capabilities and indexes. Inherited capabilities and indexes cannot be disabled. If multiple roles are specified, this role inherits capabilities and indexes from all selected roles.

| <input type="checkbox"/> | Role name          | <input type="text" value="filter"/> | Showing all ▼ |
|--------------------------|--------------------|-------------------------------------|---------------|
| <input type="checkbox"/> | admin              |                                     |               |
| <input type="checkbox"/> | can_delete         |                                     |               |
| <input type="checkbox"/> | power              |                                     |               |
| <input type="checkbox"/> | splunk-system-role |                                     |               |
| <input type="checkbox"/> | user               |                                     |               |

Cancel

Save

4. Scroll down the list of indexes until the **cm** entry is displayed.

**Edit Role user** ✕

Name ⓘ

1. Inheritance 2. Capabilities 3. **Indexes** 4. Restrictions 5. Resources

**Wildcards**  
Instead of selecting individual indexes, you can create a Wildcard Index to dynamically capture all indexes that match the Wildcard. After you add a Wildcard Index, it appears in the Indexes table. Wildcard indexes are limited to this role.

**Add**

**Indexes**  
Enable both the "Included" and "Default" checkboxes for an index to make that index searchable by default for this role. You must save this role before you can see its inherited wildcards.

| Index Name      | Included ⓘ                          | Default ⓘ                           | Showing all |
|-----------------|-------------------------------------|-------------------------------------|-------------|
| _metrics        | <input type="checkbox"/>            | <input type="checkbox"/>            |             |
| _metrics_rollup | <input type="checkbox"/>            | <input type="checkbox"/>            |             |
| _telemetry      | <input type="checkbox"/>            | <input type="checkbox"/>            |             |
| _thefishbucket  | <input type="checkbox"/>            | <input type="checkbox"/>            |             |
| cm              | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |             |
| history         | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |             |
| main            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |             |
| summary         | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |             |

**Cancel** **Save**

5. Select the **Default** check box to add the index to default searches.

6. Click **Save**.

## Configure the Syslog Server on CipherTrust Manager

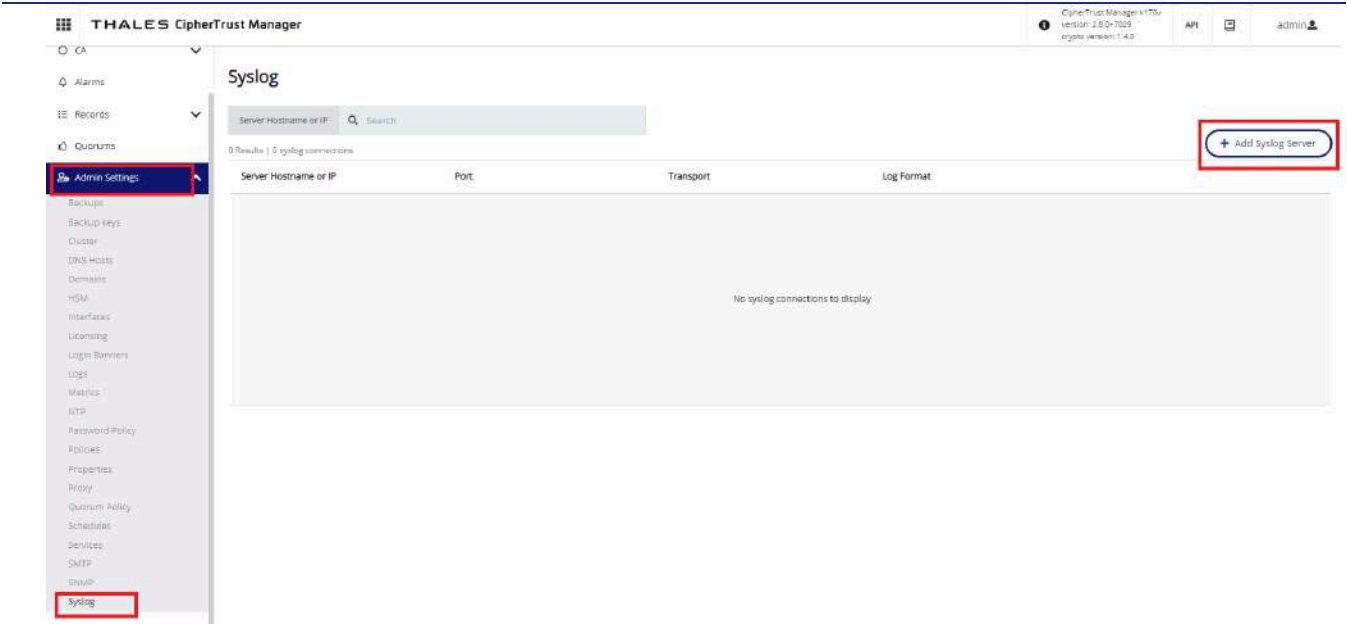
### Note

The legacy Syslog server available in "Admin Settings" will be removed in a future release. However, you can use the [Connection Manager](#) to configure the Syslog server.

Configure the CipherTrust Manager to send Syslog data to the Splunk server on the port specified earlier (for example, UDP/5514 or TCP/6514).

1. Log on to the **CipherTrust Manager** GUI.
2. Click **Admin Settings > Syslog > Add Syslog Server**.





3. Enter the Syslog server details.

Add Syslog Server

Hostname or IP Address \*

10.164.12.36

Port

5514

Log Format

RFC5424

Transport \*

UDP

Cancel

Add Syslog Server

4. Click **Add Syslog Server**.

## Return Material Authorization (RMA) Guidance

Thales ensures that all of its products are designed, manufactured, and tested to the highest level of quality. On occasion, a product may fail in the field after use by the customer. Products that fail in the field, when covered by a maintenance agreement or during the warranty period, may be eligible for an RMA (Return Material Authorization).

---

The first step to obtain an RMA is to contact [customer support](#) for your product.

For RMA of Thales CipherTrust Manager k570 or Thales CipherTrust Manager k470 appliances, contact Thales Customer Support for assistance.

For RMA of Thales TCT CipherTrust Manager k570 or Thales TCT CipherTrust Manager k160 appliances, contact Thales TCT Technical Support for assistance.

CipherTrust Manager physical appliances may contain sensitive customer key material. Thales recognizes that and assures the customers that our appliances are hardened. Follow the steps for your appliance to fully ensure data is protected before the appliance is sent for RMA.

This page covers the following appliance models:

- [Thales CipherTrust Manager k570](#)
- [Thales CipherTrust Manager k470](#)
- [Thales TCT CipherTrust Manager k570](#)
- [Thales TCT CipherTrust Manager k160](#)

## To Prepare a Thales CipherTrust Manager k570 or Thales CipherTrust Manager k470 Appliance for RMA

1. Ensure that all the sensitive information such as keys, backup keys, certificates, NAE users, and authorization policies are [backed up](#) at all times.
2. Login to the CipherTrust Manager as ksadmin via serial console or SSH.
3. Do one of the following:
  - a. Perform a **factory reset** of the CipherTrust Manager using the [system configuration utility](#). This command erases all configuration information, log files and any keys stored on the appliance.

```
kscfg system factory-reset
```

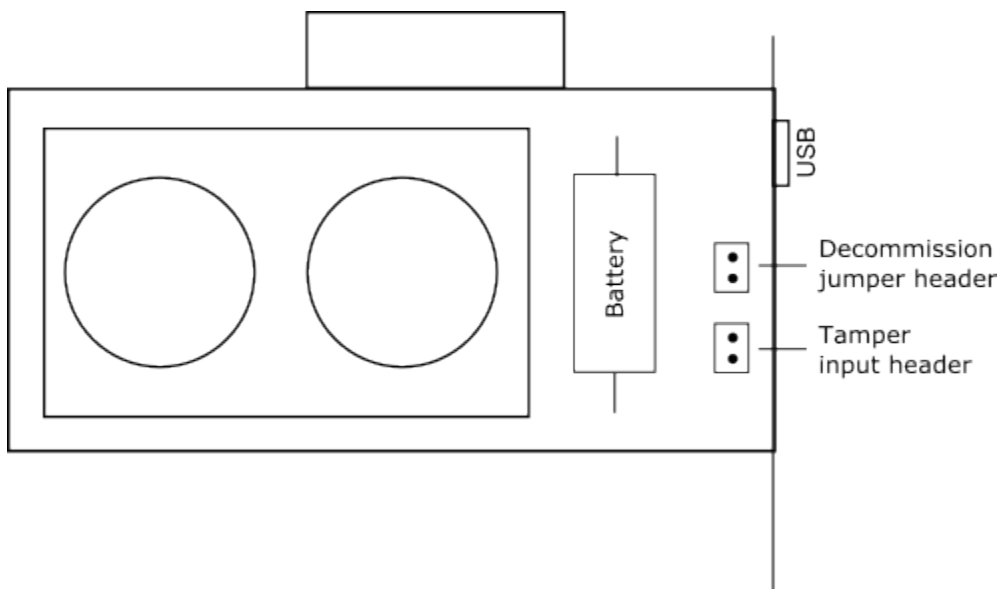
- b. Perform a **hard reset** of the CipherTrust Manager using the [system configuration utility](#). This command resets the appliance and removes data associated with CipherTrust Manager, such as keys and certificates. All log information and appliance configuration information remains intact. This remaining information can help us determine the possible cause of the failure.

```
kscfg system reset
```

4. For k570 devices, reset the on-board PCIe HSM card which stores the root keys. Do one of the following:
  - a. Login to the CipherTrust Manager as ksadmin via serial console or SSH, and run the lunaCM factory reset command.

```
lunacm:> hsm factoryReset
```

- b. Short circuit the decommission jumper header on the PCI card. You can use the blade of a screwdriver, or other conductive tool to short-circuit the two pins of the decommission header, or you can connect a switch to the decommission header if desired. Power is not required to decommission the HSM, that is, you can decommission the HSM after removing it from the chassis. The following image shows the two-pin decommission jumper header location on the PCI card:



## To Prepare a Thales TCT CipherTrust Manager k570 or Thales TCT CipherTrust Manager k160 Appliance for RMA

1. Ensure that all the sensitive information such as keys, backup keys, certificates, NAE users, and authorization policies are backed up at all times.
2. Login to the CipherTrust Manager as ksadmin via serial console or SSH.
3. Do the following:
  - a. Use the following command to delete a Root of Trust key from the Token HSM:

```
ksctl rot-keys delete --id <key_id> --force
```

- b. Remove the HSM from the CipherTrust Manager GUI interface. From Admin Settings, go into HSM and remove the HSM. This will remove all key material, etc. from the device.
4. For k160 devices, be sure to include all the Token HSMs that are associated with the device in your package when shipping the k160 unit to Thales TCT. These Tokens are keyed to the device and cannot be used on another device.

# Secrets Management

## ksctl secrets create

In the secrets create API, a new parameter `passwordConfig` is added with the option `passwordconfig`, which is of type "json". With this option, the password information is passed via file as shown below:

```
ksctl secrets create --name <name> --data-type "password" --passwordconfig <path of the json file>
```

### passwordconfig.json sample

```
{
 "passwordLength": 8,
 "lowercaseCount": 1,
 "uppercaseCount": 1,
 "symbolsCount": 1,
 "digitsCount": 1
}
```

#### Note

- Material is a mandatory field for:
  - all the non-licensed users.
  - the licensed users for all types other than "password".
- `passwordconfig` is applicable only for the licensed users and type "password".

## ksctl secrets list

The secrets list API adds a new filter version.

```
ksctl secrets list --secretversion 2
```

## ksctl secrets version

The secrets version API creates a new version of the secret.

```
ksctl secrets version --name <name>
ksctl secrets version --name <key-name> --secretmaterial <secret-material>
```

**Note**

- Material is a mandatory field for:
  - all the non-licensed users.
  - the licensed users for all types other than "password".
- If the secret password is created using password config, the same config will be used to create a new version.
- If the secret password is created using material, the user needs to provide a new material to create a new version.

## ksctl secrets list version

The secrets list version API lists all the versions of an existing secret.

```
ksctl secrets listversion--name <name>
```

## Reports

CipherTrust Manager provides easy-to-read reports containing details about cluster nodes, domains, certificates, certificate authorities (CA), clients, and license usage. You can download these reports in CSV and PDF formats.

The reports can provide insights into various questions, such as:

- What is the number of active and inactive cluster nodes, and what is their uptime?
- How many domains exist under the root, and who are their respective administrators?
- How many licenses are utilized at root and domain levels?
- How many local or external certificates are currently available and in use at root and domain levels?

## Report Types

Following types of reports are available in the CipherTrust Manager:

- [Cluster nodes report](#)
- [Domain report](#)
- [License usage by domain report](#)

- System license usage report
- Certificate reports
  - Local CA report
  - Local CA certificates report
  - External CA report
  - Clients report
  - Interface report
- Capacity report

#### Caution

This feature is a technical preview for evaluation in non-production environments. A technical preview introduces new, limited functionality for customer feedback as we work on the feature. Details and functionality are subject to change. This includes API endpoints, UI elements, and CLI commands. We cannot guarantee that data created as part of a technical preview will be retained after the feature is finalized.

- Orphaned resources report

#### Caution

This feature is a technical preview for evaluation in non-production environments. A technical preview introduces new, limited functionality for customer feedback as we work on the feature. Details and functionality are subject to change. This includes API endpoints, UI elements, and CLI commands. We cannot guarantee that data created as part of a technical preview will be retained after the feature is finalized.

## Cluster nodes report

This report provides a list of cluster nodes with the following details.

| Field   | Description                               |
|---------|-------------------------------------------|
| Node ID | Node ID of the cluster.                   |
| Status  | Active or inactive status of the cluster. |

| Field         | Description                                  |
|---------------|----------------------------------------------|
| <b>Host</b>   | Host IP of the cluster.                      |
| <b>Port</b>   | Port number.                                 |
| <b>Uptime</b> | Time for which the cluster has been running. |

To view **Cluster Nodes Report**:

1. Log in to CipherTrust Manager console as a root user.
2. Select **Admin Settings > Cluster**.
3. Click the download icon next to Cluster.
4. Select **Download As CSV** or **Download As PDF**.
5. (Optional) Rename the report file.
6. Click **Save**.

#### Note

Cluster Node Reports are available only for the root users.

## Domain report

This report provides a list of domains with the following details.

| Field                        | Description                                                         |
|------------------------------|---------------------------------------------------------------------|
| <b>Domain Name</b>           | Name of the domain.                                                 |
| <b>Domain ID</b>             | ID of the domain.                                                   |
| <b>Allow User Management</b> | Information on whether creation and management of users is allowed. |
| <b>Anchored Domain</b>       | Information on whether domain is anchored to HSM.                   |
| <b>Meta</b>                  | Additional reference information about the domain.                  |

The report helps in identifying first-level subdomains under the root domain and the respective admins who have access to those domains.

To view **Domain Report**:

1. Log in to CipherTrust Manager console as root user.
2. Select **Admin Settings > Domains**

3. Click the download icon next to Domains.
4. Select **Download As CSV** or **Download As PDF**.
5. (Optional) Rename the report file.
6. Click **Save**.

#### Note

Domain Reports are available only for the root users.

## License usage by domain report

This report provides the following license consumption details for a domain and all its subdomains. Only domains with at least one consumed license of any type are included in the report.

| Field                          | Description                                                         |
|--------------------------------|---------------------------------------------------------------------|
| <b>Domain ID</b>               | ID of the domain user using the license.                            |
| <b>Domain Name</b>             | Name of the domain user using the license.                          |
| <b>Parent Domain ID</b>        | ID of the parent domain.                                            |
| <b>Parent Domain Name</b>      | Name of the parent domain.                                          |
| <b>Transparent Encryption</b>  | Count of licenses used by CipherTrust Transparent Encryption (CTE). |
| <b>LDT</b>                     | Count of licenses used by LDT (Live Data Transformation).           |
| <b>CTE-KP</b>                  | Count of licenses used by CTE-KP (Kubernetes Protection).           |
| <b>CTE UserSpace</b>           | Count of licenses used by CTE UserSpace.                            |
| <b>CTE-RP</b>                  | Count of licenses used by CTE-RP (Ransomware Protection).           |
| <b>KMIP</b>                    | Count of licenses used by KMIP.                                     |
| <b>DPG</b>                     | Count of licenses used by DPG (Data Protection Gateway).            |
| <b>RESTful Data Protection</b> | Count of licenses used by Restful Data Protection.                  |
| <b>CADP</b>                    | Count of licenses used by CADP.                                     |
| <b>BDT</b>                     | Count of licenses used by BDT (Batch Data Transformation).          |
| <b>CCKM</b>                    | Count of licenses used by CCKM (CipherTrust Cloud Key Manager).     |



---

To view **License Usage by Domain Report**:

1. Log in to CipherTrust Manager console.
2. Select **Admin Settings > Licensing**.
3. Click the download icon next to Licensing, and select **License Usage By Domain Report**.
4. Select **Download As CSV** or **Download As PDF**.
5. (Optional) Rename the report file.
6. Click **Save**.

#### Note

License Usage By Domain Reports are available for the root and domain users.

## System license usage report

This report provides license usage details of the root domain. It shows the number of available and consumed licenses for all license types (including connectors).

| Field            | Description                                 |
|------------------|---------------------------------------------|
| <b>Connector</b> | Name of the connector.                      |
| <b>Total</b>     | Total number of licenses available for use. |
| <b>Used</b>      | Number of consumed licenses.                |

The report also contains usage breakdown for various cloud units of CCKM (CipherTrust Cloud Key Manager).

To view **System License Usage Report**:

1. Log in to CipherTrust Manager console as a root user.
2. Select **Admin Settings > Licensing**.
3. Click the download icon next to Licensing, and select **System License Usage Report**.
4. Select **Download As CSV** or **Download As PDF**.
5. (Optional) Rename the report file.
6. Click **Save**.

**Note**

System License Usage Reports are available only for the root users.

## Certificate reports

CipherTrust Manager provides reports related to the CA (Certificate Authority) certificates provisioned for different domains (including the root domain) and subdomains. You can use these reports to view details of the certificates issued by local and external CAs, and their usage details from the client and interface perspective.

### Local CA Report

This report provides a list of live and expired local CAs with the following details.

| Field                        | Description                                                     |
|------------------------------|-----------------------------------------------------------------|
| <b>Name</b>                  | Name of the local CA.                                           |
| <b>State</b>                 | State of the local CA (Active or Inactive).                     |
| <b>Serial Number</b>         | Serial number of the local CA.                                  |
| <b>Subject</b>               | Subject to identify the entity associated with the certificate. |
| <b>Issuer</b>                | Issuer details of the local CA.                                 |
| <b>Activation</b>            | Date and time of local CA activation.                           |
| <b>Expiry</b>                | Date and time of the local CA expiration.                       |
| <b>Client Authentication</b> | Whether client authentication is enabled.                       |
| <b>User Authentication</b>   | Whether user authentication is enabled.                         |

To view **Local CA Report**:

1. Log in to CipherTrust Manager console.
2. Select **CA > Local**.
3. Click the Download icon next to Local Certificate Authorities.
4. Click **Local CA Report** or **Expired Local CA Report**, and select **As CSV** or **As PDF**.
5. (Optional) Rename the report file.
6. Click **Save**.

### Local CA certificates report

This report provides a list of certificates corresponding to a local CA with the following details.

| Field                | Description                                                     |
|----------------------|-----------------------------------------------------------------|
| <b>Name</b>          | Name of the local CA certificate.                               |
| <b>State</b>         | State of the local CA certificate (Active or Inactive).         |
| <b>Serial Number</b> | Serial number of the local CA certificate.                      |
| <b>Subject</b>       | Subject to identify the entity associated with the certificate. |
| <b>Issuer</b>        | Issuer details of the local CA.                                 |
| <b>Activation</b>    | Date and time of certificate activation.                        |
| <b>Expiry</b>        | Date and time of certificate expiration.                        |

To view **Local CA Certificates Report**:

1. In the CipherTrust Manager console, select **CA > Local**.
2. Select any local CA from the list.
3. Click the Download icon next to Certificates issued by <local\_ca\_id>.
4. Click **Local CA Certificates Report** and select **As CSV** or **As PDF**.
5. (Optional) Rename the report file.
6. Click **Save**.

## External CA report

This report provides a list of external CAs with the following details.

| Field                        | Description                                                     |
|------------------------------|-----------------------------------------------------------------|
| <b>Name</b>                  | Name of the CA.                                                 |
| <b>Serial Number</b>         | Serial number of the CA.                                        |
| <b>Subject</b>               | Subject to identify the entity associated with the certificate. |
| <b>Issuer</b>                | Issuer details of the CA.                                       |
| <b>Activation</b>            | Date and time of the CA activation.                             |
| <b>Expiry</b>                | Date and time of the CA expiration.                             |
| <b>Client Authentication</b> | Whether client authentication is enabled.                       |
| <b>User Authentication</b>   | Whether user authentication is enabled.                         |

To view **External CA Report**:

1. Log in to CipherTrust Manager console.
2. Select **CA > External**.

3. Click the download icon next to External Certificate Authorities.
4. Select **Download As CSV** or **Download As PDF**.
5. (Optional) Rename the report file.
6. Click **Save**.

## Clients report

This report provides a list of clients and their corresponding certificates with the following details.

| Field                      | Description                                                                                       |
|----------------------------|---------------------------------------------------------------------------------------------------|
| <b>ID</b>                  | ID of the client.                                                                                 |
| <b>Name</b>                | Name of the client.                                                                               |
| <b>Created At</b>          | Date and time of client creation.                                                                 |
| <b>Certificate ID</b>      | Certificate associated with the client.                                                           |
| <b>Client State</b>        | Active or inactive status of the client.                                                          |
| <b>Client Type</b>         | Type of the registered connector client such as CTE, Application Data Protection, or kmip client. |
| <b>Certificate Issuer</b>  | Name of the certificate issuer .                                                                  |
| <b>Certificate Subject</b> | Additional information about the certificate.                                                     |
| <b>Client Hostname</b>     | Hostname of the client.                                                                           |
| <b>Certificate Expiry</b>  | Data and time of certificate expiration.                                                          |

To view **Clients Report**:

1. Log in to CipherTrust Manager console.
2. Select **Access Management > Client Hub**.
3. Click the download icon next to Client Hub.
4. Select **Download As CSV** or **Download As PDF**.
5. (Optional) Rename the report file.
6. Click **Save**.

## Interface report

This report provides a list of interfaces and their corresponding CA details with the following details.

| Field       | Description            |
|-------------|------------------------|
| <b>Name</b> | Name of the interface. |

| Field                          | Description                                                                                                                |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Mode</b>                    | Interface mode to identify whether TLS is enabled and whether the password is required or is optional for TLS connections. |
| <b>Certificate User Field</b>  | Certificate field that identifies a user or a connector client connected with an interface.                                |
| <b>Autogen CA</b>              | Auto-generated Certificate Authority associated with the interface.                                                        |
| <b>External Trusted CAs</b>    | External CA associated with the interface.                                                                                 |
| <b>Local Trusted CAs</b>       | Local CA associated with the interface.                                                                                    |
| <b>Port</b>                    | Port number.                                                                                                               |
| <b>Network Interface</b>       | NIC card or virtual interface in use for communication.                                                                    |
| <b>Type</b>                    | Type of the registered connector client such as CTE, Application Data Protection, or kmip client.                          |
| <b>Certificate Expiry Date</b> | Data and time of certificate expiration.                                                                                   |
| <b>Minimum TLS Version</b>     | Minimum supported TLS version.                                                                                             |
| <b>Maximum TLS Version</b>     | Maximum supported TLS version.                                                                                             |

To view **Interface Report**:

1. Log in to CipherTrust Manager console.
2. Select **Admin Settings > Interfaces**.
3. Click the download icon next to Interfaces.
4. Select **Download As CSV** or **Download As PDF**.
5. (Optional) Rename the report file.
6. Click **Save**.

## Capacity report

### Caution

This feature is a technical preview for evaluation in non-production environments. A technical preview introduces new, limited functionality for customer feedback as we work on the feature. Details and functionality are subject to change. This includes API endpoints, UI elements, and CLI commands. We cannot guarantee that data created as part of a technical preview will be retained after the feature is finalized.

Capacity reports provide resource counts in the current domain and its ten direct child domains with the most

resources. This helps users in the `admins` group such as `admin`, or the `Domain Admins` group, monitor if resources are approaching maximum appliance capacity limits and make decisions on resource management. Consult the [Keys](#) and [Domains](#) page for capacity limits. This report can be run in any domain the logged in user has permission to access, through the `ksctl` CLI or REST API.

To access this report, run the command `ksctl reports capacity-report`.

### Note

It can take up to an hour for changes to resources to appear in this report.

| Returned field                                    | Description                                                                                                                                                                                                                                                  |
|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>domain_name</code>                          | The first instance is the name of the current domain. Subsequent instances are the names of direct child domains.                                                                                                                                            |
| <code>domain_id</code>                            | The first instance is the identifier of the current domain. Subsequent instances are the identifiers of direct child domains.                                                                                                                                |
| <code>key_usage_count_this_this</code>            | The first instance is the number of keys belonging to the current domain. Subsequent instances are the numbers of keys belonging to a child domain.                                                                                                          |
| <code>subdomain_count_this_domain</code>          | The first instance is the number of child domains of the current domain. Subsequent instances are the number of child domains for each of the child domains, or counts of grandchild domains for the current domain.                                         |
| <code>subdomain_count_including_subdomains</code> | The first instance is the number of child and grandchild domains for the current domain. Subsequent instances are the number of child and grandchild domains for child domains, or counts of grandchild and great grandchild domains for the current domain. |
| <code>subdomains_info_by_key_usage_count</code>   | An array of direct child domain resource usage counts, sorted in order of highest key usage. This field is only returned for the current domain.                                                                                                             |
| <code>subdomains_info_by_subdomain_count</code>   | An array of direct child domain resource usage counts, sorted in order of highest number of child domains, or most grandchild domains of the current domain. This field is only returned for the current domain.                                             |

## Orphaned resources report

### Caution

This feature is a technical preview for evaluation in non-production environments. A technical preview introduces new, limited functionality for customer feedback as we work on the feature. Details and

functionality are subject to change. This includes API endpoints, UI elements, and CLI commands. We cannot guarantee that data created as part of a technical preview will be retained after the feature is finalized.

Orphaned resources report provide counts of objects that are associated with a deleted domain. If a domain is deleted without first deleting its resources, the resources remain in the database. Orphaned resources count toward maximum capacity limits for the CipherTrust Manager appliance or Virtual CipherTrust Manager instance. This report is only available in the root domain to users in the `admin` group, such as `admin`, through the `ksctl` CLI or REST API.

To access this report, run the command `ksctl reports orphaned-resources`.

#### Note

It can take up to an hour for newly orphaned resources to appear in this report.

| Returned field                         | Description                                                              |
|----------------------------------------|--------------------------------------------------------------------------|
| <code>total_orphaned_keys_count</code> | A count of all keys on the appliance belonging to a deleted domain.      |
| <code>orphaned_keys_by_account</code>  | An array with an itemized count of keys with an associated user account. |
| <code>orphaned_key_count</code>        | A count of orphaned keys for the account.                                |
| <code>account</code>                   | The domain account that originated the orphaned keys.                    |

## Key Templates

The CipherTrust Manager provides the support of templates for creating Keys. This template is used to specify the attributes of a key for various operations. Attributes (`key_attributes`) specified in a template are applied to all the keys that reference the template by a name.

#### Note

- Members only of `admin` and `Key Admins` groups are allowed to manage templates.
- Members of `Key Users` and `Restricted Key Users` groups can only read templates.

**Note**

Currently, templates are not included in domain or partial domain backup. Only option is for customer to re-create them in case of disaster recovery.

## Managing templates

The following operations can be performed on templates:

- Create templates
- Get template
- list templates
- Update template
- Delete templates

## Creating templates

To create a template:

### Syntax

```
ksctl templates create --name <name> --description <description> --labels <comma-separated-key:value> --meta <comma-separated-key:value> --key_attributes <comma-separated-key:value>
```

Here,

- `name` - name of template.
- `description` - description of template.
- `labels` - key/value pairs to group templates.
- `meta` - end-user/service data defined in the template.
- `key_attributes` - attributes of a key defined the template.

### Example Request

```
ksctl templates create --name 'Test Template New' --desc 'Symmetric' --labels 'sale=HR' --meta '{ "color":"Red"}' --key_attributes '{ "algorithm": "AES", "size": 256 }'
```



## Example Response

```
{
 "id": "211d1451-5280-491b-ae17-00543bfaa32a",
 "uri": "kylo:kylo:vault:templates:test-template-new-211d1451-5280-491b-ae17-00543bfaa32a",
 "account": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2024-05-01T10:16:21.486296Z",
 "updatedAt": "2024-05-01T10:16:21.486296Z",
 "name": "Test Template New",
 "description": "Symmetric",
 "meta": {
 "color": "Red"
 },
 "labels": {
 "sale": "HR"
 },
 "key_attributes": {
 "algorithm": "AES",
 "size": 256
 }
}
```

## Getting details of templates

To get detail of a template:

### Syntax

```
ksctl templates get --id <name/ID-of-a-template>
```

### Example Request

```
ksctl templates get --id "211d1451-5280-491b-ae17-00543bfaa32a"
```

### Example Response

```
{
 "id": "211d1451-5280-491b-ae17-00543bfaa32a",
 "uri": "kylo:kylo:vault:templates:test-template-new-211d1451-5280-491b-ae17-00543bfaa32a",
 "account": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2024-05-01T10:16:21.486296Z",
 "updatedAt": "2024-05-01T10:16:21.486296Z",
 "name": "Test Template New",
 "description": "Symmetric",
 "meta": {
 "color": "Red"
 },
 "labels": {
 "sale": "HR"
 },
 "key_attributes": {
 "size": 256,

```

```

 "algorithm": "AES"
 }
}

```

## Getting list of templates

To fetch a list of template:

### Syntax

```
ksctl templates list --labels-query <query-filter>
```

Here, `labels-query` filters results that match label selector expressions. Multiple values are logically ANDed.

### Example Request

```
ksctl templates list --labels-query "team=HR"
```

### Example Response

```

{
 "skip": 0,
 "limit": 10,
 "total": 1,
 "resources": [
 {
 "id": "8d6dd37a-1183-4e92-8c5f-5ba3d4714c41",
 "uri": "kylo:kylo:vault:templates:8d6dd37a-1183-4e92-8c5f-5ba3d4714c4
1",
 "account": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2024-05-01T07:17:26.836585Z",
 "updatedAt": "2024-05-01T07:43:13.092583Z",
 "name": "tp-8d6dd37a-1183-4e92-8c5f-5ba3d4714c41",
 "description": "test template create",
 "meta": {
 "color": "red"
 },
 "labels": {
 "team": "HR"
 },
 "key_attributes": {
 "meta": {
 "cte": {
 "cte_versioned": false,
 "encryption_mode": "CBC",
 "unique_to_client": true,
 "persistent_on_client": true
 },
 "kmip": {
 "custom": [
 {
 "type": "TextString",
 "x-attr1": "test_12"
 }
]
 }
 }
 }
 }
]
}

```

```

 {
 "type": "TextString",
 "x-attr2": "test_13"
 }
],
 "alternative_names": [],
 "app_specific_info": [
 {
 "application_data": "thalesdocs",
 "application_namespace": "namespace"
 },
 {
 "application_data": "thalesdocs_ew",
 "application_namespace": "namespace_1"
 }
],
 "contact_information": "Thales"
},
"versionedKey": true,
"customAttributes": [
 {
 "name": "x-nae-attr",
 "value": "test"
 }
]
},
"size": 256,
"algorithm": "AES"
}
]
}
}

```

## Updating templates

To update a template:

### Syntax

```

ksctl templates update --id <template id> --name <name> --description <description> --labels <comma-separated-key:value> --meta <comma-separated-key:value> --key_attributes <comma-separated-key:value>

```

Here,

- `name` - name of template.
- `description` - description of template.
- `labels` - key/value pairs to group templates.
- `meta` - end-user/service data defined in the template.
- `key_attributes` - attributes of a key defined the template.

## Example Request

```
ksctl templates update --id "211d1451-5280-491b-ae17-00543bfaa32a" --key_attributes '{ "algorithm": "AES", "size": 256, "objectType": "Symmetric Key", "format": "raw" }'
```

## Example Response

```
{
 "id": "211d1451-5280-491b-ae17-00543bfaa32a",
 "uri": "kylo:kylo:vault:templates:test-template-new-211d1451-5280-491b-ae17-00543bfaa32a",
 "account": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2024-05-01T10:16:21.486296Z",
 "updatedAt": "2024-05-01T10:18:46.321065Z",
 "name": "Test Template New",
 "description": "Symmetric",
 "meta": {
 "color": "Red"
 },
 "labels": {
 "sale": "HR"
 },
 "key_attributes": {
 "algorithm": "AES",
 "size": 256,
 "objectType": "Symmetric Key",
 "format": "raw"
 }
}
```

## Deleting templates

To delete a template:

### Syntax

```
ksctl templates delete --id <id/name>
```

## Example Request

```
ksctl templates delete --id "8d6dd37a-1183-4e92-8c5f-5ba3d4714c41"
```

## Example Response

There will be no response if template is deleted successfully.

## Managing public SSH keys for ksadmin

The [ksadmin user](#), is a specialized operating system administrator that can access the CipherTrust Manager via SSH or via password authentication on a physical server console port. Additionally, you can also enhance the access by adding more public SSH keys for the ksadmin user to connect securely to the CipherTrust Manager.

To add additional public SSH keys:

1. From ksadmin home, list the directories using `-a` option. Since the `.ssh` folder is hidden.

Alternatively, use WinSCP. Go to **Options > Preferences > Panels** and select **Show Hidden Files**.

2. Modify the `authorized_keys` file directly in Windows to add the new SSH public key. Or, use WinSCP.
3. Reboot the system.

## Example

Below is an example of file modification.

### Original file

**Command:**

```
ksadmin@keysecure:~/.ssh$ cat authorized_keys.original
```

**Output:**

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACW0up8bpe+f3fB7R8/JbpnMASDcF+UiNdUDAMYAtiE
2DXPW2ZKqyggOolpYc4FR6aovogyWo2q+MhXk+UUdBa3RrdOprRUedP0ZcoHnKyvimGscUUSzFRQPi7Tl
PVl60zfSoxAZQCa2YH1R4JCUv0hzmL3XAY9JoLnPwML+eq6uLUzYXvaoVqVNNZd0X5zX4O/YdbaDvmIO4
IBc6f2Er+Rzp07mMN61Y6XOA2G4ULLUV1APnVwmAoFElnBxjyFYT7liP/uizI6dxviU4SDXyBWxvL54kw
5+1IHjGztXdddeqB2hv7ZvvEl7XdNanVUYefTJ019j82TDz+7i/p9Nel PV-Linux-keypair_PUBlIC
```

### Modified file

**Command:**

```
ksadmin@keysecure:~/.ssh$ cat authorized_keys
```

**Output:**

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACW0up8bpe+f3fB7R8/JbpnMASDcF+UiNdUDAMYAtiE
2DXPW2ZKqyggOolpYc4FR6aovogyWo2q+MhXk+UUdBa3RrdOprRUedP0ZcoHnKyvimGscUUSzFRQPi7Tl
PVl60zfSoxAZQCa2YH1R4JCUv0hzmL3XAY9JoLnPwML+eq6uLUzYXvaoVqVNNZd0X5zX4O/YdbaDvmIO4
IBc6f2Er+Rzp07mMN61Y6XOA2G4ULLUV1APnVwmAoFElnBxjyFYT7liP/uizI6dxviU4SDXyBWxvL54kw
5+1IHjGztXdddeqB2hv7ZvvEl7XdNanVUYefTJ019j82TDz+7i/p9Nel PV-Linux-keypair_PUBlIC

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAC9o5irVdzR1GNQwt+oQP/5t9IzsdwTROw3Q2qBG+OSk
bExAdSRnNydAGmsJ6H5kAaKUYMnOYX7xPphgGYLmocz/eba7fAqFfnAlYYWL91bQ61PDtnV1S/HJ0tLyn
RWHUOVGZInzMfCrRuHqqNMpuhQ7OopisY4wwQ57XtUi8sRrLNjEYiQRL7oJc5EajFe5Mgr47KGCs2o1E
X2VgxLdmpu4XVWrA/9YbqMjE+HMXrP1MURofluompiT/GKjMZpbHUMQfHlHK9b7YYTtUGixhDVI46pFm
3YQ50ETO5HZAwweP7ySqNFJ4tbBT86kpexG+8ktHw5PYaFshlgk4nlY5 SEG-kylo_public
```

# CipherTrust Data Security Platform Architecture

A technical introduction to the CipherTrust Data Security Platform and Sample Use Cases



White Paper

# Contents

|           |                                                                  |
|-----------|------------------------------------------------------------------|
| <b>3</b>  | <b>Executive Summary</b>                                         |
| <b>3</b>  | <b>Platform Capabilities and Benefits</b>                        |
| 3         | Platform Capabilities                                            |
| 3         | Environments and Technologies Supported                          |
| 4         | Compliance Regulations Supported                                 |
| <b>4</b>  | <b>CipherTrust Platform – Products Summary</b>                   |
| <b>5</b>  | <b>CipherTrust Platform Connectors</b>                           |
| 5         | CipherTrust Manager                                              |
| 6         | CipherTrust Enterprise Key Management                            |
| 8         | CipherTrust Data Discovery and Classification                    |
| 9         | CipherTrust Transparent Encryption                               |
| 10        | CipherTrust Application Data Protection                          |
| 11        | CipherTrust Tokenization                                         |
| 13        | CipherTrust Database Protection                                  |
| <b>14</b> | <b>CipherTrust Data Security Platform Sample Use Cases</b>       |
| 14        | Centralized Key Management – Enterprise and Cloud Key Management |
| 15        | Securely Migrating Data to Hybrid Cloud Environments             |
| 16        | Protecting Big Data Environments                                 |
| 17        | Satisfying Data Privacy and Security Compliance Regulations      |
| <b>18</b> | <b>Comprehensive Data Security from Thales</b>                   |
| <b>18</b> | <b>About Thales</b>                                              |

# Executive Summary



With the widespread adoption of cloud services, big data environments, and IoT technologies, organizations are moving huge amounts of sensitive data rapidly to third party and partner infrastructures. All of this makes today's data environments increasingly complex to manage and secure. So, it comes as no surprise that almost half of the respondents to the Thales [2021 Data Threat Report](#) survey, see multi-cloud data security management as more complex to manage.

The CipherTrust Data Security Platform from Thales integrates data discovery, classification, and industry-leading data protection solutions across diverse IT environments to provide adaptive data-centric security. The platform provides powerful tools to combat external threats, guard against insider abuse, and establish persistent controls, even when data is stored in the cloud or any external provider's infrastructure and supports an evolving regulatory landscape. This white paper provides an overview of the deployment architecture of the CipherTrust Platform products and the compelling use-cases that they enable for customers along their data protection journey.

## Platform Capabilities and Benefits

### Platform Capabilities

- Centralized management console
- Data discovery and classification
  - Risk analysis with data visualizations
- Data Protection Connectors
  - Transparent encryption for files, databases and containers
  - Application-layer data protection
  - Format preserving encryption
  - Tokenization with static and dynamic data masking
  - Database protection
  - Privileged user access controls
- Proactive Data Protection
  - Integrated data discovery, classification and automated protection
- Centralized enterprise key management
  - FIPS 140-2 compliant
  - Multi-cloud key management
  - Unparalleled ecosystem of KMIP partner integrations
  - Transparent Data encryption (TDE) key management
- Monitoring and reporting

### Environments and Technologies Supported

- IaaS, PaaS and SaaS: Amazon Web Services, Google Cloud Platform, Microsoft Azure, IBM Cloud, Salesforce, Microsoft Office365 , Oracle Cloud Infrastructure, Alibaba Cloud
- OS: Linux, Windows and AIX
- Big data: Hadoop, NoSQL, SAP HANA and Teradata
- Databases: IBM DB2, Microsoft SQL Server, MongoDB, MySQL, NoSQL, Oracle, Sybase and others
- Containers: Docker, Red Hat OpenShift
- Any storage environment



## Compliance Regulations Supported

GDPR

PCI-DSS

HIPAA

SOX/GLBA

ISO/IEC 27002:2013

FIPS 140-2

FISMA, FedRAMP

NIST 800-53 Rev 4

California Consumer Privacy Act

South Africa POPI Act

Japan My Number Compliance

South Korea's PIPA

India's Aadhaar Act

Philippine's Data Privacy Act

Monitory Act of Singapore

Australia Privacy Amendment

# CipherTrust Platform – Products Summary

The CipherTrust Data Security Platform features the following products to discover, classify and protect sensitive data to enable organizations to securely extract value from sensitive data and confidently adopt digital transformation technologies

- **CipherTrust Manager**

CipherTrust Manager is the central management point for the platform. It is an industry-leading enterprise key management solution that enables organizations to centrally manage encryption keys, provide granular access controls and configure security policies. CipherTrust Manager manages key lifecycle tasks including generation, rotation, destruction, import and export, provides role-based access control to keys and policies, supports robust auditing and reporting, and offers development- and management-friendly REST APIs. CipherTrust Manager is available in physical and virtual form-factors that are FIPS 140-2 compliant up to level 3. The CipherTrust Manager can also be rooted to a hardware security module (HSM) such as Thales Luna and Luna Cloud HSM.

- **CipherTrust Data Discovery and Classification**

CipherTrust Data Discovery and Classification locates regulated data, both structured and unstructured, across the cloud, big data, and traditional data stores. A single pane of glass delivers understanding of sensitive data and its risks, enabling better decisions about closing security gaps, compliance violations, and prioritizing remediation. The solution provides a streamlined workflow all the way from policy configuration, discovery, and classification, to risk analysis and reporting, helping to eliminate security blind spots and complexities.

- **CipherTrust Transparent Encryption**

CipherTrust Transparent Encryption delivers data-at-rest encryption, privileged user access controls and detailed data access audit logging. Agents protect data in files, volumes and databases on Windows, AIX and Linux OS's across physical and virtual servers and in cloud and big data environments. The Live Data Transformation extension is available for CipherTrust Transparent Encryption, providing zero-downtime encryption and data rekeying. In addition, security intelligence logs and reports streamline compliance reporting and speed up threat detection using leading security information and event management (SIEM) systems.

- **CipherTrust Intelligent Protection**

CipherTrust Intelligent Protection enables organizations to rapidly discover and classify data based on sensitivity, vulnerability, and other risk profiles and pro-actively protect at-risk data using encryption and access controls. It integrates CipherTrust Data Discovery with Classification and CipherTrust Transparent Encryption to improve operational efficiencies, accelerate time to compliance, and pro-actively close security gaps.

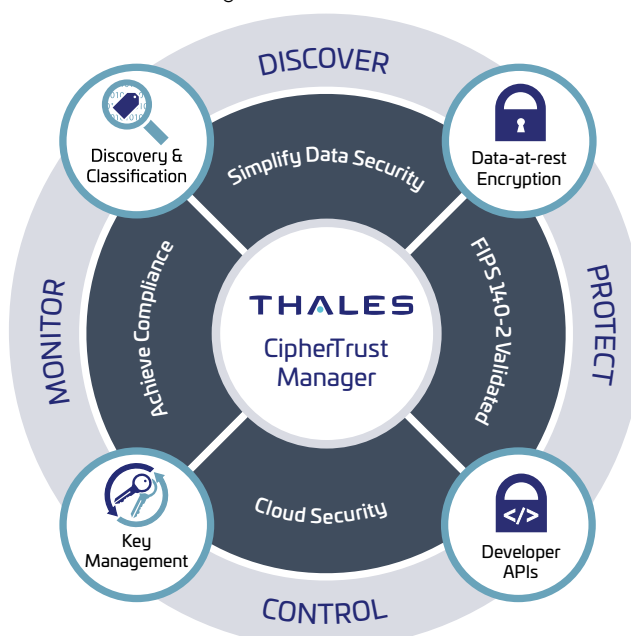


Figure 1: CipherTrust Data Security Platform

- **CipherTrust Application Data Protection**

CipherTrust Application Data Protection delivers crypto functions such as key management, signing, hashing and encryption services through APIs, so that developers can easily secure data at the application server or big data node. The solution comes with supported sample code so that developers can move quickly to securing data processed in their applications. CipherTrust Application Data Protection accelerates development of customized data security solutions, while removing the complexity of key management from developer responsibility and control. In addition, it enforces strong separation of duties through key management policies that are managed only by security operations.

- **CipherTrust Tokenization**

CipherTrust Tokenization is offered both vaulted and vaultless tokenization and can help reduce the cost and complexity of complying with data security mandates such as PCI-DSS. The vaultless offering includes policy-based dynamic data masking. Tokenization replaces sensitive data with a representative token, so that the sensitive data is kept separate and secure from the database and unauthorized users and systems. Both offerings make it easy to add tokenization to applications.

- **CipherTrust Database Protection**

CipherTrust Database Protection solutions integrate data encryption for sensitive fields in databases with secure, centralized key management and without the need to alter database applications. CipherTrust Database Protection solutions support Oracle, Microsoft SQL Server, IBM DB2 and Teradata databases.

- **CipherTrust Key Management**

CipherTrust Key Management delivers a robust, standards-based solutions for managing encryption keys across the enterprise. It simplifies administrative challenges around encryption key management to ensure that keys are secure and always provisioned to authorized encryption services. CipherTrust Key Management solutions support a variety of use cases including:

- **CipherTrust Cloud Key Manager** streamlines bring your own key (BYOK) management for Amazon Web Services, Microsoft Azure, Salesforce and IBM Cloud. The solution provides comprehensive cloud key lifecycle management and automation to enhance security team efficiency and simplify cloud key management.
- **CipherTrust TDE Key Management** supports a broad range of database solutions such as Oracle, Microsoft SQL, and Microsoft Always Encrypted.
- **CipherTrust KMIP Server** centralizes management of KMIP clients, such as full disk encryption (FDE), big data, IBM DB2, tape archives, VMware vSphere and vSAN encryption, etc.

# CipherTrust Platform Connectors

## CipherTrust Manager

CipherTrust Manager enables organizations to centrally manage encryption keys, provide granular access control and configure security policies. It manages key lifecycle tasks including generation, rotation, destruction, import and export, provides role-based access control to keys and policies, supports robust auditing and reporting, and offers developer friendly APIs.

- **Unified Management Console**

CipherTrust Manager provides a single pane of glass for discovering and classifying sensitive data integrated with a comprehensive set of data protection connectors to encrypt or tokenize data to reduce business risk and satisfy compliance regulations. It streamlines provisioning of connector licenses through a new customer facing licensing portal and provides better visibility and control of licenses in use.

- **Centralized Key Management and Access Control**

It offers centralized key lifecycle, certificate and policy management with role-based access control and provides full audit log review. It authenticates and authorizes administrators and key users using existing AD and LDAP credentials.

- **Cloud Friendly Deployment Options**

CipherTrust Manager offers several options to securely migrate applications to multiple cloud environments. It offers support for AWS, Azure, Google Cloud, VMware, HyperV, Oracle VM and more. In addition, CipherTrust Cloud Key Manager supports bring your own key (BYOK) across multiple cloud infrastructures and SaaS applications.

- **Developer Friendly APIs**

CipherTrust Manager offers new REST interfaces, in addition to KMIP and NAE-XML APIs, for developers to simplify deployment of applications integrated with key management capabilities and automate development and testing of administrative functions

- **Improved Monitoring and Alerting**

It includes tracking of all administrator access, encryption key state and policy changes in multiple log formats (RFC-5424, CEF, and LEEF) for easy integration with SIEM tools. In addition, customers can generate pre-configured and customizable email alerts (SNMP v1, v2c, v3).

- **Hybrid High-Availability Clustering**

It offers a choice of clustering CipherTrust Manager physical appliances (k470, k570) and a virtual appliances (k170v, k470v) for high-availability environments to ensure optimum processing regardless of the workload location (data center or cloud).

## Robust Separation of Duties and Multitenancy Support

CipherTrust Manager can enforce strong separation of duties by requiring the assignment of key and policy management privileges to one or more data security administrators for different departments within a large enterprise. It provides capabilities to create multiple domains to support large enterprises with distributed locations.

## FIPS 140-2 Compliant

CipherTrust Manager provides several options to integrate with a FIPS 140-2 validated physical or virtual HSMs as a secure root of trust for better key entropy.

- Built-in HSM crypto accelerator card on a CipherTrust Manager k570 appliance.
- Network attached Luna HSM with HA clustering
- Luna Cloud HSM on Data Protection on Demand and other cloud HSMs like AWS CloudHSM, Azure Dedicated HSM, and IBM Cloud HSM.

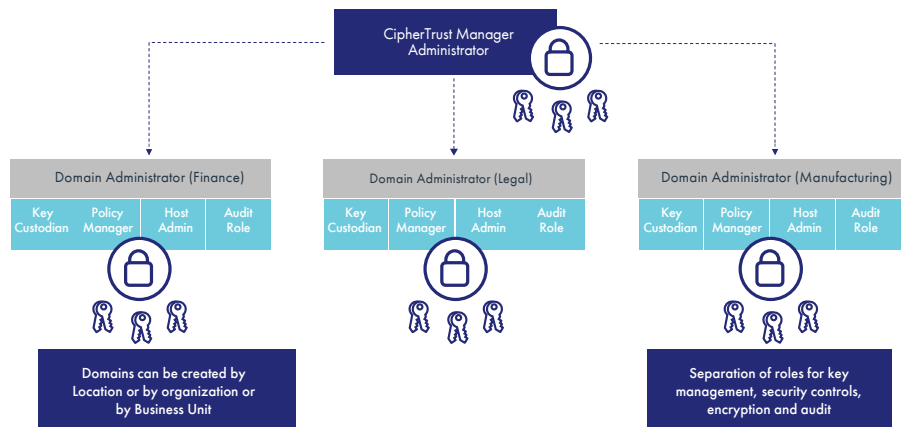


Figure 2: Multi-Tenancy Support with Strong Separation of Duties

## CipherTrust Enterprise Key Management

CipherTrust Enterprise Key Management solutions enable organizations to centrally manage and store cryptographic keys and policies for third party devices including Microsoft SQL TDE, Oracle TDE, and KMIP-compliant encryption products. CipherTrust Enterprise Key Management delivers a robust, standards-based platform for managing encryption keys from disparate devices 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.

- CipherTrust Cloud Key Manager** simplifies Bring Your Own Keys (BYOK) and provider created keys for multiple cloud service providers and SaaS applications, while addressing enterprise needs to meet compliance and best practices for managing encryption key life cycles outside of their native environments, without the need for enterprises to become cryptographic experts. It uses CipherTrust Manager as the key source
- CipherTrust KMIP Server** supports interoperability with a broad range of KMIP compatible data storage devices including SAN and NAS storage arrays, self encrypting drives and hyper-converged infrastructure solutions and follow best practices for strong and secure enterprise key management. CipherTrust Manager has already been certified with a number of 3rd party applications utilizing KMIP.
- CipherTrust TDE Key Management** provides key lifecycle management for Oracle TDE master encryption keys and Microsoft SQL Server database encryption keys to meet compliance requirements and follow best practices.
- CipherTrust LUKS Key Management** provides transparent disk encryption for Linux. The LUKS Key Agents enables you to centrally manage encryption keys for Linux disk partitions.

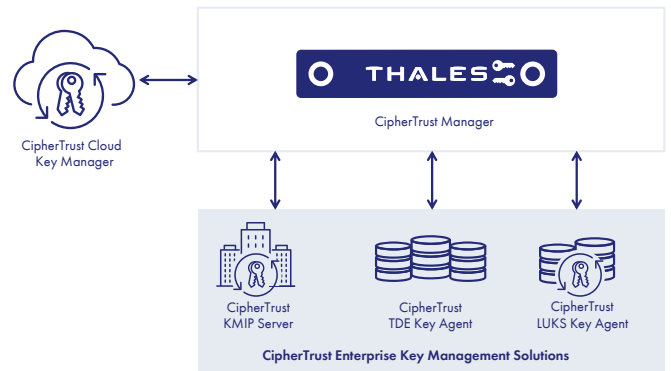


Figure 3: CipherTrust Enterprise Key Management Solutions

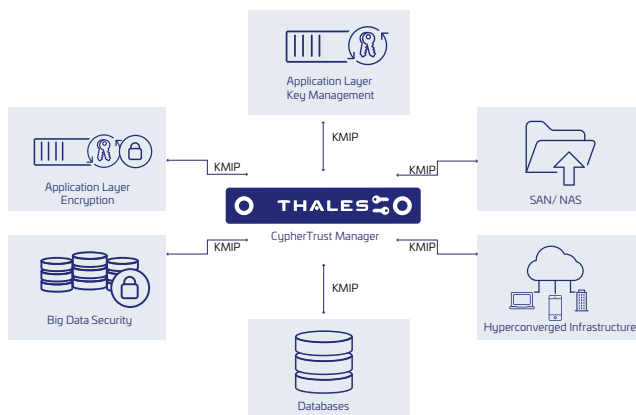


Figure 4: KMIP Client Types Supported by CipherTrust Manager

## CipherTrust KMIP Server

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.

How it works:

1. CipherTrust Manager needs to be setup for KMIP support, which entails adding the KMIP clients on the CipherTrust Manager.

2. Trust between the CipherTrust Manager and the KMIP client needs to be established.
3. For a mutually authenticated Transport Layer Security (TLS) connection between KMIP clients and servers, the client needs to be registered with the CipherTrust Manager at which point the certificate for the KMIP client gets created.

## CipherTrust TDE Key Management

Managing encryption keys for native Transparent Data Encryption (TDE) solutions presents challenges such as isolating keys from the assets they protect and storing them securely - not only a best practice for key management, but a common industry data protection mandate. CipherTrust TDE 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

- **Microsoft SQL Server Transparent Data Encryption**

CipherTrust 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**

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 table spaces within the databases. CipherTrust Enterprise Key Management solutions interface with the Oracle Wallet to protect and manage these MEKs within a secure FIPS-certified boundary.

## CipherTrust Cloud Key Manager

CipherTrust Cloud Key Manager provides "Bring Your Own Key" (BYOK) services to enable customers to control the keys used to encrypt their data in multiple cloud service providers. Customer key control allows for the separation, creation, ownership and control, including revocation, of encryption keys or tenant secrets used to create them. Leveraging cloud provider BYOK API's, the CipherTrust Cloud Key Manager reduces key management complexity and operational costs by giving customers lifecycle control of encryption keys with centralized management and visibility.

- **Multi-cloud Key Control:** CipherTrust Cloud Key Manager provides comprehensive key lifecycle management to fulfill requirements for safe, comprehensive key management across multiple cloud providers. It satisfies industry best practices and data security standards that mandate encryption keys to be stored and managed centrally, remotely from the cloud service provider and the associated encryption operations.
- **Enhanced IT Efficiency:** Centralized key management gives you access to each cloud provider from a single browser window, including across multiple accounts or subscriptions. Automated key rotation offers IT efficiency and enhanced data security. For workloads that require it, CipherTrust Cloud Key Manager can request creation of native cloud provider keys and provide full lifecycle management for them.
- **Strong Encryption Key Security:** CipherTrust Cloud Key Manager leverages the security of the virtual or physical CipherTrust Manager appliances, as a key source. It can also use Thales hardware security modules (HSMs) to create keys and store with a highest root of trust.

### Supported Cloud Environments:

- Amazon Web Services
- Google Cloud
- Google Workspace
- Microsoft Azure
- Microsoft Office365
- Microsoft Azure Stack
- Microsoft Azure China National Cloud
- IBM Cloud
- Salesforce.com
- Salesforce Sandbox



Figure 5: Support for Multiple Cloud Environments

## CipherTrust Data Discovery and Classification

CipherTrust Data Discovery and Classification enables organizations to get complete visibility of their sensitive data with efficient data discovery, classification, and risk analysis across cloud, big data, and traditional environments. It enables organizations to uncover and mitigate data privacy gaps and proactively respond to growing data privacy and security regulations, such as GDPR, CCPA, PCI DSS, HIPAA and more.

CipherTrust Data Security Platform unifies data discovery, classification and data protection, as well as granular access controls and centralized key management – all on a single platform. CipherTrust Data Discovery and Classification enables IT organizations to manage data privacy and security with centralized management. From a single pane of glass organizations can set policies, discover data that adheres or violates those policies, classify data, and rank risks. They can then automate remediation using CipherTrust data protection solutions to protect sensitive data and respond to regulatory challenges with a single platform. To achieve this clear view, organizations need to:

- 1. Define Policies:** for data privacy, classification profiles, and scans for multiple data stores.
- 2. Discover:** Locate both structured and unstructured sensitive data across the entire enterprise in multi-cloud, big data, relational databases, or file storage systems.
- 3. Classify:** Sensitive data such as national IDs, financial data, and personal data, based on built-in templates or market-proven classification techniques.
- 4. Analyze Risk:** Understand the risks with rich visualizations and risk scores.
- 5. Remediate:** Proactively protect sensitive data soon after it is discovered and classified using risk-based policies
- 6. Generate Reports:** Leverage charts and reports for risk analysis and to support compliance programs.



Figure 6: Streamline Data Discovery, Classification and Remediation Actions

Technical Specifications

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div><b>Data Stores</b><ul style="list-style-type: none"><li>Local storage and local memory on the host</li><li>Network storage<ul style="list-style-type: none"><li>Windows Share (CIS/SMB)</li><li>Unix File System (NFS)</li></ul></li><li>Databases<ul style="list-style-type: none"><li>IBM DB2</li><li>Oracle</li><li>SQL</li><li>PostgreSQL</li><li>SAP HANA</li><li>Mongo DB</li><li>MySQL</li></ul></li><li>Big Data<ul style="list-style-type: none"><li>Hadoop Clusters</li><li>Teradata</li></ul></li><li>Cloud<ul style="list-style-type: none"><li>AWS S3 buckets</li><li>Azure Blob</li><li>Azure Table</li><li>Google Apps (Gmail and Gdrive)</li><li>Office 365 (Exchange, SharePoint)</li></ul></li></ul></div> | <div><b>Type of data identified</b><ul style="list-style-type: none"><li>Health (Australian Medicare Card, European EHIC, US Health Insurance Claim number, etc.)</li><li>Financial (American Express, Diners Club, Mastercard, VISA card numbers, bank account number, etc.)</li><li>Personal (name, last name, address, DOB, email, etc.)</li><li>National ID (social security number, Spanish DNI, etc.)</li></ul></div> |
| <div><b>Type of files supported:</b><ul style="list-style-type: none"><li>Databases: Access, DBase, SQLite, MSSQL MDF &amp; LDF</li><li>Images: BMP, FAX, GIF, JPG, PDF (embedded), PNG, TIF</li><li>Compressed: bzip2, Gzip (all types), TAR, Zip (all types)</li><li>Microsoft Backup Archive: Microsoft Binary / BKF</li><li>Microsoft Office: v5, 6, 95, 97, 2000, XP, 2003 onwards</li><li>Open Source: Star Office / Open Office</li><li>Open Standards: PDF, HTML, CSV, TXT</li></ul></div>                                                                                                                                                                                                                                | <div><b>Pre-built templates</b><p>The solution includes a wide range of ready-to-use templates that can help organizations meet common regulatory and business policy needs.</p><ul style="list-style-type: none"><li>CCPA</li><li>GDPR</li><li>HIPAA</li><li>PCI DSS</li><li>PII</li><li>PHI</li></ul></div>                                                                                                               |

CipherTrust Transparent Encryption

CipherTrust Transparent Encryption delivers data-at-rest encryption with centralized key management, privileged user access control and detailed data access audit logging. It transparently encrypts data wherever it resides -- on-premises file systems/servers, across multiple clouds and within big data, and container environments.

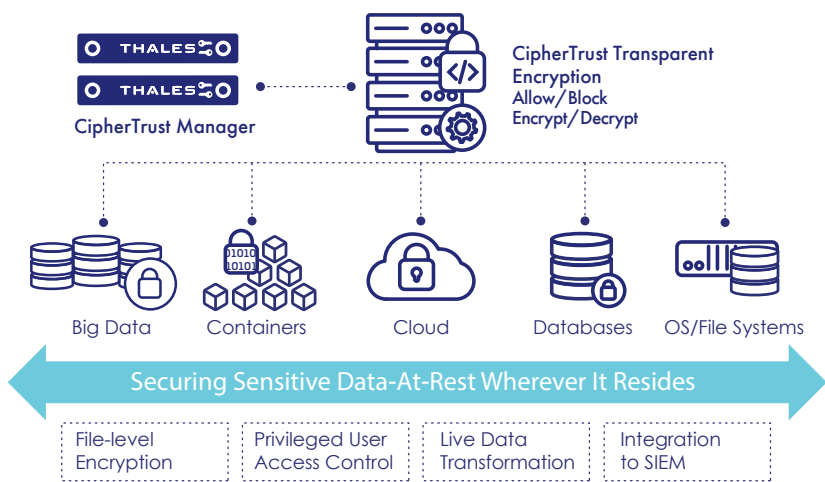


Figure 7: CipherTrust Transparent Encryption

The deployment is simple, scalable and fast, with Transparent Encryption agents installed at the operating file-system or device layer, and encryption and decryption is transparent to all applications that run above it. CipherTrust Transparent Encryption is designed to meet data security compliance and best practice requirements with minimal disruption, effort, and cost. Implementation of the encryption software is seamless keeping both business and operational processes working without changes even during deployment and roll out.

Transparent Encryption agents are kernel level drivers that sit above file systems or volumes in the OS stack. Agents perform policy-based access control, encryption/ decryption and data access auditing. The solution enables organizations to define access policies that are checked before granting access to “GuardPoints” - resources that are protected by fine-grained access policies as follows.

- A GuardPoint can encompass a complete disk drive volume, or a specific directory or an AWS S3 bucket under which all the unstructured files or structured database files reside.
- An access policy defines WHO (user) or WHAT (process) can access the GuardPoint, WHEN, specific file ACTION permitted (read/write/delete/rename) and EFFECT (permit/deny, apply key for encryption/decryption, and audit user access).

### Protection Against Privileged User Abuse

Fine-grained access control policies can be setup to restrict privileged users (root, domain admins) to perform administrative tasks only (such as system backups, updates, and hardware maintenance), without giving them access to decrypt sensitive data. Any malware that covertly elevates privileges to gain admin access and gain access to sensitive data is prevented from doing so if access policies are setup correctly.

### Preventing Ransomware from Encrypting Sensitive Files

Access policies can be defined to create a whitelist of “trusted” applications to prevent any untrusted binaries (e.g. ransomware) from accessing data stores (GuardPoints) protected by the Transparent Encryption agent and to prevent privileged users from accessing user data in files and databases. The granular access policies can enable organizations to block any rogue binaries from encrypting files/databases, even if the intruder has execute permissions for that binary and read/write permission to the target file that contains business critical data.

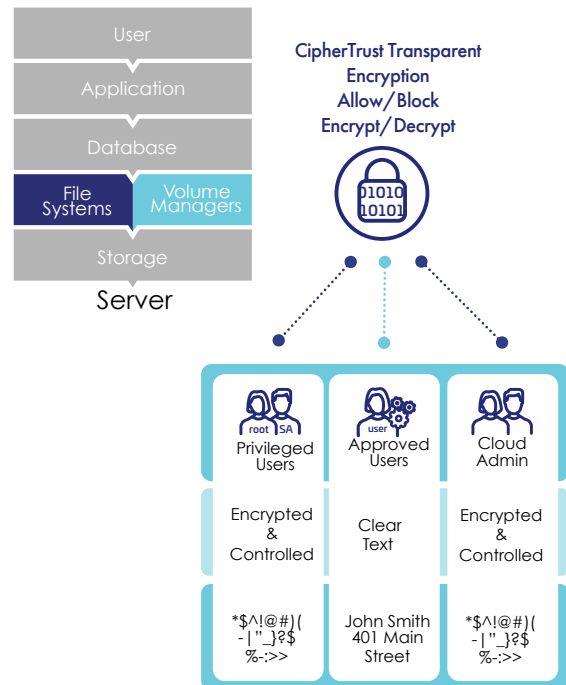


Figure 8: CipherTrust Transparent Encryption Access Control Policies

### Integration with SIEM Solutions

CipherTrust security intelligence logs and reports streamline compliance reporting and speed up threat detection using leading security information and event management (SIEM) systems.

### CipherTrust Transparent Encryption Extensions

- **Live Data Transformation:** Eliminates the downtime required for initial encryption and scheduled rekeying operations. This patented technology allows for databases or files to be encrypted or re-keyed with a new encryption key while the data is in use without taking applications off-line.
- **SAP HANA Qualified:** SAP has qualified CipherTrust Transparent Encryption with HANA v2.0 to deliver data encryption, key management, privileged user access control, and granular file access audit logs. This solution can be quickly deployed, requiring no changes to SAP HANA or the underlying database or hardware infrastructure.

## CipherTrust Application Data Protection

CipherTrust Application Data Protection delivers crypto functions such as key management, signing, hashing and encryption services through APIs so that developers can easily secure data on application servers or big data nodes. The solution comes with supported sample code so that developers can move quickly to securing data processed in their applications. CipherTrust Application Data Protection accelerates development of customized data security solutions, while removing the complexity of key management from the developers’ responsibility and control.

### Flexible, Efficient Implementation

CipherTrust Application Data Protection reduces the time, complexity, and risk of developing and implementing an in-house encryption and key management solution with support for a range of encryption providers and KMIP. Developers can utilize Java, C and C++ to integrate between applications and CipherTrust Application Data Protection.

### Robust Centralized Key Management

Enterprises must not only protect against data theft, but they must also protect their encryption keys from theft, misplacement, or accidental destruction. To facilitate these safeguards, the CipherTrust Application Data Protection crypto library supports enterprise key management through CipherTrust Manager. Depending on a range of architectural, performance and risk assessment choices, customers can choose to retain keys and perform all encryption operations in CipherTrust Manager, similar to an HSM. A configuration change, not requiring any code changes, enables temporary use of keys for encryption or decryption on the CipherTrust Application Data Protection servers. Keys on servers are encrypted when not in use and obfuscated in memory when in use.



## Strong Safeguards

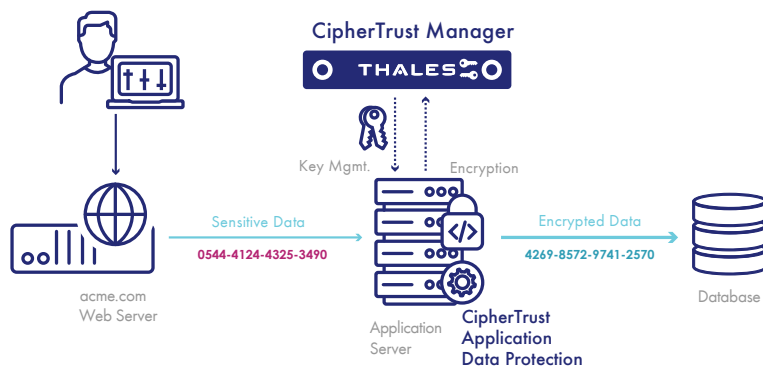


Figure 9: CipherTrust Application Data Protection

CipherTrust Application Data Protection delivers the controls needed to address security policies and compliance mandates, both for data on premises and in PaaS environments. With the solution, you can stop malicious DBAs, cloud administrators, hackers and, in cloud environments, even authorities with subpoenas from accessing valuable data.

## Technical Specifications

### Development Libraries and APIs

- Java, C/C++, .NET, .Net Core
- XML open interface, KMIP standard
- Web services: REST

### Encryption Algorithms:

- 3DES, AES, AES-XTS, SHA, HMAC, RSA, ECC
- Format-preserving: FF1/FF3, Tokenization

### Web Application Servers

- Apache Tomcat, IBM WebSphere, JBoss, Microsoft IIS, Oracle WebLogic, SAP NetWeaver, Sun ONE, and more

### Cloud and Virtual Infrastructures

- Works with all major cloud platforms, including AWS, Azure, IBM Cloud, Google and VMware

### Supported Platforms for ICAPI Provider

- Red Hat Enterprise Linux 5.4 and above
- Microsoft Windows 2003, 2008 R2, and 7 in both 32-bit and 64-bit

## CipherTrust Tokenization

CipherTrust Tokenization enables organizations to mask sensitive data and address their compliance objectives, while gaining breakthroughs in operational efficiency. It provides a single platform that offers database tokenization and dynamic display security. With this solution, security teams can meet PCI DSS requirements and secure data in cloud, big data, and data center environments—and do so with minimal disruption and administrative overhead. Any organization can replace sensitive assets with tokens, so attackers and malicious contractors and employees can't exploit the information to further their agendas.

### Dynamic Data Masking

Security administrators can establish policies to return an entire field tokenized or dynamically mask parts of a field, enabling role-based display security. Administrators can establish settings to have an entire field tokenized or may dynamically mask data so that only portions of a field are visible, such as the first few digits of a Social Security Number or credit card. It can be integrated with your existing LDAP and Active Directory based identity and access management systems, so your security teams can efficiently set granular policies for specific users and groups. For example, a user with customer service representative credentials can see a credit card number with the last four digits visible for customer identification purposes, while a customer service supervisor may be able to see the entire credit card number.

- **Comprehensive Support:** CipherTrust Tokenization supports the following options:
  - Format preserving tokenization
  - Random and sequential tokens



- Either irreversible tokens or the option to delete tokens
- Single and multi-use tokens
- Partial tokenization
- Dynamic data masking, alpha-numeric, and custom mask characters.
- **Efficient Implementation:** This solution offers easy-to-use REST APIs for integration with the CipherTrust Tokenization Server, so application developers can simply and quickly add tokenization and dynamic data masking to applications. Developers don't have to manually institute identity management or redaction policies.
- **Agility and Scalability:** CipherTrust Tokenization delivers the high performance needed to address the operational demands of the most processing-intensive environments. It runs on virtual machines and can be quickly and efficiently scaled up and scaled down to accommodate changing workloads.

## Technical Specifications

### Tokenization capabilities:

- Format-preserving tokens with irreversible option
- Random tokens data length up to 128K
- Date tokenization
- Luhn checking option for FPE and random tokens

### Dynamic data masking capabilities:

- Policy based, number of left and/or right characters exposed, with customizable mask character
- Authentication with Lightweight Directory Access Protocol (LDAP) or Active Directory (AD)

### Deployment Form Factors and Options: Open Virtualization Format (.OVA) and International

- Organization for Standardization (.iso)
- Microsoft Hyper-V VHD
- Amazon Machine Image (.ami)
- Microsoft Azure Marketplace
- Google Cloud Platform

### System requirements:

- Minimum hardware: 4 CPU cores, 16–32 GB RAM
- Minimum disk: 80GB

### Application integration:

- RESTful APIs

### Performance:

More than 1 million credit card size tokenization transactions per second, per token server (using multiple threads and batch (or vector) mode) on a 32-core server (dual-socket Xeon E5- 2630v3) with 16 GB RAM

# CipherTrust Database Protection

CipherTrust Database Protection provides transparent column-level encryption of structured, sensitive data residing in databases, such as credit card, social security numbers, national ID numbers, passwords, and email addresses. This solution enables large amounts of sensitive data to be moved in and out of data stores by efficiently encrypting and decrypting specific columns in databases. No changes are required to applications. It can scale-up to support multiple data centers in on-premises, virtual, and public cloud environments.

CipherTrust Database Protection solutions work with CipherTrust Manager for centralized key and policy management.

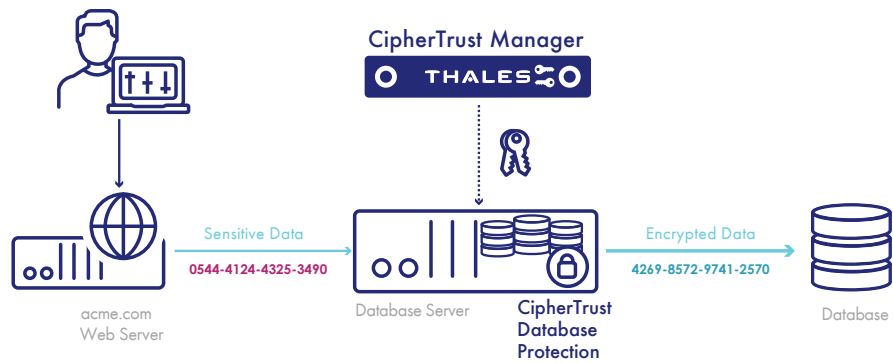


Figure 10: CipherTrust Database Protection

This solution enables organizations to seamlessly encrypt or tokenize sensitive database columns.

## Technical Specifications

|                                                                                                                                       |                                                                                                                                                                              |
|---------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Supported Databases:</b> <ul style="list-style-type: none"><li>• Oracle</li><li>• Microsoft SQL Server</li><li>• IBM DB2</li></ul> | <b>Supported Platforms:</b> <ul style="list-style-type: none"><li>• Microsoft Windows</li><li>• Linux</li><li>• Solaris</li><li>• HP-UX</li><li>• AIX</li></ul>              |
| <b>Encryption Algorithms:</b> <ul style="list-style-type: none"><li>• AES, 3DES, FF3, FF1, RSA, ECC</li></ul>                         | <b>Cloud and Virtual Infrastructures:</b> <ul style="list-style-type: none"><li>• Works with all major cloud platforms, including AWS, Microsoft Azure, and VMware</li></ul> |

The CipherTrust Platform also offers the following database protection solutions.

### CipherTrust Protection for Teradata Database

Delivers fast, efficient and robust data-at-rest security capabilities for Teradata environments, securing sensitive assets in both Teradata Database and the Teradata Integrated Big Data Platform. It integrates with Teradata with simple user-defined functions (UDF) for encryption and decryption. Both Cipher-Block Chaining (CBC) and format-preserving encryption (FPE) modes are available. Use of FPE enables dynamic data masking for decryption operations on a per-user basis. Teradata database and big data analytics solutions enable organizations to more fully leverage information to fuel improved decisions, products, services and business results.

### Static Data Masking with CipherTrust Batch Data Transformation

CipherTrust Batch Data Transformation is a high-speed data protection tool offering both encryption and tokenization with static data masking. It leverages CipherTrust Manager for key management and CipherTrust Application Data Protection and CipherTrust Tokenization to facilitate the encryption or tokenization of high volumes of sensitive records without lengthy maintenance windows and downtime. This solution can either encrypt or tokenize sensitive data on a per-column basis. Tokenization and encryption may be used concurrently on different database columns.

Batch Data Transformation is commonly used for a wide range of static data masking use cases:

- Fast and efficient re-keying of existing encrypted data.
- Masking sensitive data before or loading into a data warehouse or data lake.
- Initial encryption or tokenization of existing data in production databases prior to deployment of applications that encrypt or tokenize new data.
- Enable third-party data analysis and clean-up without exposing sensitive or private information
- Enable data science or data analysis team members to utilize accurately represented data without exposure of sensitive content

# CipherTrust Data Security Platform

## Sample Use Cases

### Centralized Key Management – Enterprise and Cloud Key Management

#### The Challenge

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.

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, according to this [Aberdeen Report](#). 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. In addition, each separate encryption system requires specialized training to learn the unique processes that are specific to that system.

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.

#### Solution: CipherTrust Enterprise Key Management

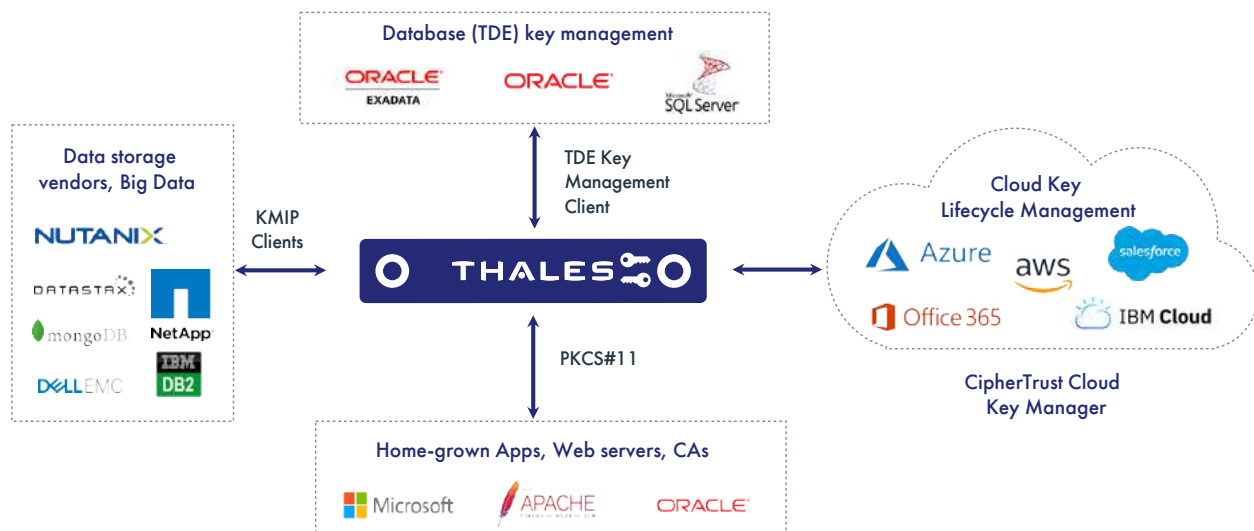


Figure 11: CipherTrust Enterprise Key Management Solutions

CipherTrust Enterprise Key Management solutions centralize key management for internal as well as commercial off the shelf (COTS) applications and storage solutions. This gives customers greater command over your keys while increasing encryption data security. CipherTrust Key Management products connect with your applications through standard interfaces and deliver access to robust key management and encryption functions.

CipherTrust Manager- the foundation for Thales enterprise key management solutions, is a high-availability appliance that centralizes encryption key management for the CipherTrust Data Security Platform 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.

It supports the following three major interoperability standards that are supported by most server, storage and device vendors currently deployed in existing IT environments.

- 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.
- 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 (TDE).
- OASIS KMIP** – Key Management Interoperability Protocol (KMIP), 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.
- CipherTrust Cloud Key Manager:** supports bring your own keys (BYOK) and provider created keys for multiple cloud service providers and SaaS applications, while addressing enterprise needs to meet compliance and best practices for managing encryption key life cycles outside of their native environments, without the need for enterprises to become cryptographic experts. It uses CipherTrust Manager as the key source.

## Securely Migrating Data to Hybrid Cloud Environments

### The Challenge

Cloud computing is transforming the way enterprises, government agencies, and small businesses manage their company data. Elastic public cloud services are enabling agile, cost-effective methods to run business-critical applications and store information. And, while some enterprises aren't yet ready to let go of the traditional on-premises data center, they are exploring and evaluating all available options.

Bring your own encryption (BYOE) is a security model that gives cloud customers complete control over the encryption of their data by allowing them to deploy a virtualized instance of their own encryption software in tandem with the application they are hosting in the cloud. It is possible in this scenario for the end user to manage their encryption keys within the cloud; however, given the legal pressures that a

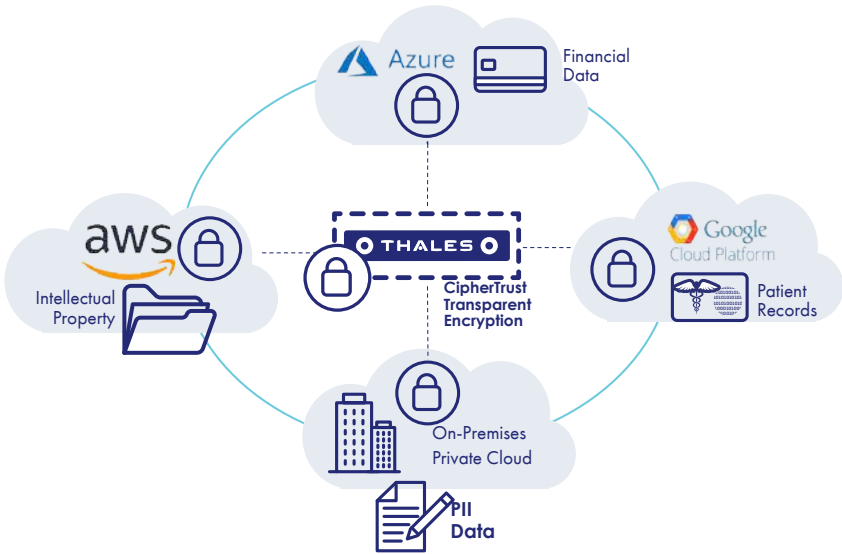


Figure 12: Support for Hybrid Clouds with CipherTrust Transparent Encryption

cloud service provider (CSP) could potentially face, it would make little sense to encrypt data and then store the encryption keys in the same environment. To fully secure data in an untrusted and multi-tenant cloud environment, organizations must maintain complete governance and control of their data.

### Solution: CipherTrust Transparent Encryption

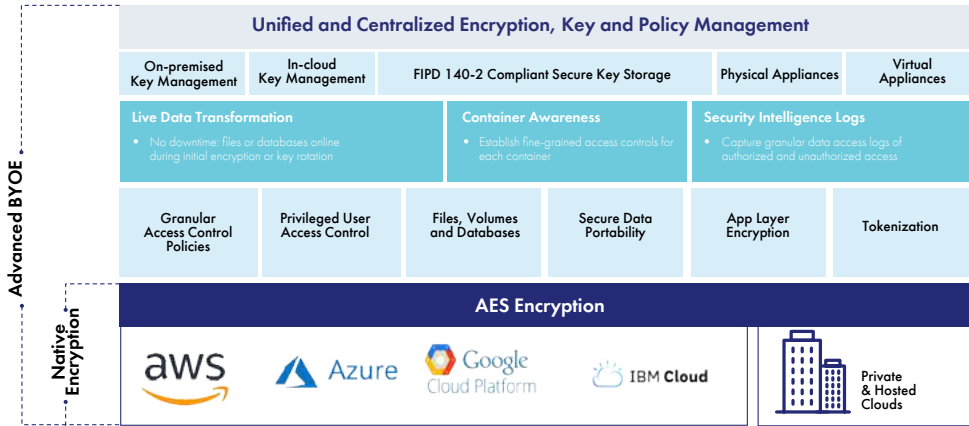


Figure 13: Centralized Encryption, Key and Policy Management with CipherTrust Transparent Encryption

Most cloud service and leading encryption providers use the same encryption technique - Advanced Encryption Standard, or AES. However, encrypting data is only a starting point. To truly protect your data you need to consider the threats you're protecting against, managing encryption keys and access controls across multiple cloud providers. Compared to the native encryption solutions available from cloud providers, CipherTrust BYOE solutions gives you higher confidence that your data is secure and that you are in compliance with mandates by delivering the following solutions.

- CipherTrust Transparent Encryption encrypts sensitive data (such as credit card numbers, personal information, logs, passwords, configurations, and more) on servers in a broad range of files, databases, containers, as well as big data implementations in the cloud. It features granular access controls, which ensures only authorized users or processes can view protected data and prevent rogue administrators from impersonating another user who has access to sensitive data.
- CipherTrust Transparent Encryption extensions enable use of data in the cloud during encryption and rekeying operations with patented Live Data Transformation. CipherTrust Transparent Encryption monitors and logs file access to accelerate threat detection.
- Simplified key management across on-premises and multi-cloud deployments by centralizing control on the FIPS 140-2 compliant CipherTrust Manager.

## Protecting Big Data Environments

### The Challenge

With the explosive growth of data in every aspect of our lives and in enterprises around the world, there is growing demand to derive value from this data and provide business intelligence. Enterprises depend on this intelligence so they can meet their customers' needs in a timely manner and with greater precision. Along with traditional sources of data such as transactional systems and data warehouses, new sources of data, such as those from the "Internet of Things" (click logs, social media interactions and sensors), have emerged. Collectively, these vastly larger information volumes and new assets are known as Big Data. With nearly every enterprise embracing big data environments, and with large numbers of these environments implemented in the cloud, the security of the sensitive data within the data lake, source data environments, and the reports that hold high-value correlated results have become an insistent concern. Unfortunately, many organizations hesitate looking at security – and more specifically, encryption – when it comes to big data solutions because they are concerned about deploying at scale or impeding the analytics tools that make these solutions so valuable in the first place.

### Lack of effective access control

Unauthorized access could not only result in financial loss, identity theft, and reputational damage, but could also run your organization afoul of regulatory compliance. Privileged users are granted substantial access to corporate network resources to be able to perform their routine duties. However, if these users are malicious, or if their credentials are stolen, it can lead to a major data breach.

### Data privacy violations

Big data comes from multiple sources at a high velocity, volume, variety, and degree of complexity. It is no secret that privacy violations from internationally-originated data is a huge concern for companies that deal with big data.

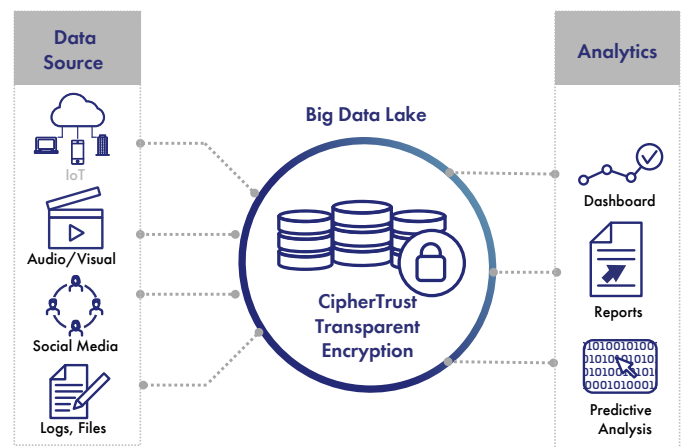


Figure 14: CipherTrust Products Support for Big Data Environments

### Solution: CipherTrust Transparent Encryption

CipherTrust Transparent Encryption offers the granular controls, robust encryption, and comprehensive coverage that organizations need to secure sensitive data across their big data environments— including data sources, infrastructure, and analytics. The solution can be used to protect data at the file system level within compute nodes (and underlying storage), source data locations, as well as the repositories used for logs and reports. And, this protection extends beyond the system level users/ groups and LDAP/AD users and groups that are enforced by Transparent Encryption agent on a typical server. The solution also enforces policy-based encryption, access controls and data access logging by Hadoop users, groups and zones. This capability provides further protection against privileged users within the big data lake or users within the environment.

A typical deployment includes agents installed on compute nodes, source data servers, and servers accessing log/report repositories. Data is encrypted throughout the environment with appropriate access policies and data access logging controls provided by the CipherTrust Manager. Further, the use of hardware encryption capabilities in underlying compute infrastructure results in minimal overhead from encrypt/decrypt operations. This makes it possible to use the solution even where speed and compute capability are critical. By leveraging the CipherTrust Data Security Platform to secure big data lake environments, organizations can realize the following benefits:

- **Compliance:** CipherTrust Data Security Platform addresses a broad set of use cases to discover, classify, and protect sensitive data across their big data environments. The platform delivers the comprehensive capabilities that enable organizations to address the demands of evolving regional data protection and privacy laws.
- **Prevent privileged-user threats:** CipherTrust Transparent Encryption provides the fine-grained, policy-based access controls, including Hadoop granular user access controls, which restrict access to encrypted data. These allow only approved access to data by processes and users as required to meet strict compliance requirements.
- **Achieve robust security:** Make the most of big data analytics with confidence that the collected and mined data, including that which is sensitive, is protected.

## Satisfying Data Privacy and Security Compliance Regulations

### The Challenge

As large datasets amounts of sensitive data is migrating to SaaS applications, big data and hybrid-cloud environments business risks are growing because of the following reasons.

- Lack of visibility to where sensitive data resides is leaving gaps in implementing data protection
- Encrypting everything is not an ideal option, since that can impact operational efficiencies
- Compliance gaps are occurring frequently since all sensitive data is not protected everywhere in the enterprise

### Solution: CipherTrust Data Security Platform

The CipherTrust Data Security Platform provides an integrated solution for the entire data security journey starting from discovering and classifying sensitive data, analyzing data at risk based on regulatory mandates, and then protecting sensitive data wherever it resides using a variety of CipherTrust Data Protection Connectors.

The CipherTrust Platform enables organizations to meet the following types of compliance requirements that are common across most global compliance regulations and standards, such as GDPR, PCI DSS, HIPAA, NIST and many more.

| Compliance Requirement                                                                                                  | CipherTrust Platform Product                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identify all assets including sensitive data that needs to be protected                                                 | CipherTrust Data Discovery and Classification                                                                                                                                                                                                                                                                                                                                                                                           |
| Protect sensitive data against unauthorized access and restrict access to sensitive data                                | CipherTrust Transparent Encryption                                                                                                                                                                                                                                                                                                                                                                                                      |
| Monitor (audit) all authorized/unauthorized user and administrative access to sensitive data                            | CipherTrust Security Intelligence                                                                                                                                                                                                                                                                                                                                                                                                       |
| Remediate/mitigate risks of sensitive data exposure with strong encryption and data masking (tokenization) technologies | <p>CipherTrust Platform Connectors:</p> <ul style="list-style-type: none"> <li>• Transparent Encryption</li> <li>• Application Data Protection</li> <li>• Database Protection</li> <li>• Tokenization</li> </ul> <p>Intelligent Protection: Automatically mitigates risks by discovering unstructured sensitive data and pro-actively protecting it using Data Discovery and Classification integrated with Transparent Encryption.</p> |
| Provide a strong encryption key management system that is FIPS certified with elevated root of trust                    | CipherTrust Manager integrated with Thales Hardware Security Modules                                                                                                                                                                                                                                                                                                                                                                    |

# Comprehensive Data Security from Thales

Thales offers a comprehensive portfolio of data protection products that include data discovery and classification, data encryption, tokenization, and centralized key management capabilities, which enable customers to protect business critical data wherever it resides -- in file servers, databases, applications, on-premises, or in multi-cloud environments. The industry leading CipherTrust Data Security Platform from Thales enables you to simplify security, improve operational efficiency, and accelerate time to compliance.

## 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](https://cpl.thalesgroup.com/contact-us)

> [cpl.thalesgroup.com](https://cpl.thalesgroup.com) <





# CipherTrust Manager

[cpl.thalesgroup.com](http://cpl.thalesgroup.com)

**THALES**  
Building a future we can all trust

CipherTrust Manager simplifies key lifecycle management tasks, including secure key generation, backup/restore, clustering, deactivation, and deletion by enabling organizations to centrally manage encryption keys for Thales CipherTrust Data Security Platform and third-party products.

Role-based access provides control to keys and policies, multi-tenancy support, and robust auditing and reporting of all key management and encryption operations.

As the central management point for the [CipherTrust Data Security Platform \(CDSP\)](#), CipherTrust Manager provides a unified management console that makes it easy to discover and classify data, and to protect sensitive data wherever it resides. CDSP makes available a comprehensive set of CipherTrust Data Protection Connectors from Thales, including Secrets Management and Ransomware Protection, and REST, KMIP, and NAE XML APIs for custom solutions.

## Key Capabilities

- **Full Key Lifecycle Management and Automated Operations:** CipherTrust Manager simplifies management of encryption keys across their entire lifecycle, including secure key generation, backup/restore, clustering, deactivation, and deletion. It makes automated, policy-driven operations easy to perform, and generates alarms for events of interest.
- **Quorum Authorization:** Allows an administrator to require multiple approvers for a sensitive operation.
- **Centralized Administration and Access Controls:** Unifies key management operations with role-based access controls. Authenticates and authorizes administrators and key users using existing AD and LDAP credentials. Prevents unauthorized password change and alerts on simultaneous logins by same user.
- **Self-service Licensing:** Streamlines provisioning of connector licenses through a new customer facing licensing portal. The new management console offers better visibility and control of licenses in use.
- **Secrets Management:** Provides the ability to create and manage secrets and opaque objects for usage on the platform.
- **Multi-tenancy Support:** Supports separation of duties with delegated user management within multiple domains.
- **Developer Friendly REST APIs:** Offers new REST interfaces, in addition to Key Management Interoperability Protocol (KMIP) and NAE-XML APIs, allows customers to remotely generate and manage keys.
- **Flexible HA Clustering and Intelligent Key Sharing:** Provides the option of clustering physical and / or virtual appliances together to assure high availability as well as increased encryption transaction throughput.
- **Robust Auditing and Reporting:** Includes tracking of all key state changes, administrator access, and policy changes in multiple log formats (RFC-5424, CEF, LEEF) for easy integration with SIEM tools.

- **Broad Partner Ecosystem:** CipherTrust Manager provides centralized key management for a wide variety of storage partners via KMIP and database partners via Transparent Database Encryption (TDE).

## Benefits

- Centralized key and policy management for on-premises data stores and cloud infrastructures
- Reduced business risk with unified data discovery, classification and sensitive data protection
- Simplified management with self-service licensing portal and visibility into licenses in use
- Cloud-friendly deployment options with support for public, private and hybrid clouds. Public: AWS, Azure, Google Cloud, Oracle Cloud and Alibaba Cloud. Private image files: VMware vSphere OVA, Microsoft Hyper-V VHDX, Nutanix AHV VMDK and OpenStack QCOW2. Hybrid cloud image files: Azure Stack HCI, Azure Stack Hub
- Expanded Hardware Security Module (HSM) support for superior key control
- Unparalleled partner ecosystem of integrations with leading enterprise storage, server, database, application and SaaS vendors



**CipherTrust Manager**

## Deployment Options

CipherTrust Manager is available in both virtual and physical form-factors that integrate with FIPS 140-2 validated Thales Luna Network or Cloud HSM Hardware Security Modules (HSMs) to securely store master keys with the highest root of trust. These appliances can be deployed on-premises as well as in private or public cloud infrastructures. This allows customers to address compliance requirements, regulatory mandates and industry best practices for data security.

## CipherTrust Manager Features

| Features                                | Virtual Appliances                                                                                                                                                                                       |                                                                                                                                                                                                          | Physical Appliances                                                                                                                                                                                      |                                                                                                                                                                                                                                |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                         | k170v                                                                                                                                                                                                    | k470v                                                                                                                                                                                                    | k470                                                                                                                                                                                                     | k570                                                                                                                                                                                                                           |
| <b>Administrative Interfaces</b>        | Management Console, REST API, kscfg (system configuration), (ksctl (Command Line Interface)                                                                                                              |                                                                                                                                                                                                          |                                                                                                                                                                                                          |                                                                                                                                                                                                                                |
| <b>Network Management</b>               | SNMP v1, v2c, v3, NTP, Syslog-TCP                                                                                                                                                                        |                                                                                                                                                                                                          |                                                                                                                                                                                                          |                                                                                                                                                                                                                                |
| <b>Monitoring</b>                       | Prometheus, Splunk                                                                                                                                                                                       |                                                                                                                                                                                                          |                                                                                                                                                                                                          |                                                                                                                                                                                                                                |
| <b>API Support</b>                      | REST, NAE-XML, KMIP, PKCS#11, JCE, .NET, MCCAPI, MS CNG                                                                                                                                                  |                                                                                                                                                                                                          |                                                                                                                                                                                                          |                                                                                                                                                                                                                                |
| <b>Secure Authentication</b>            | Local User , AD/LDAP, LDAPS, Certificate based authentication, Supports Open ID Connect (OIDC)                                                                                                           |                                                                                                                                                                                                          |                                                                                                                                                                                                          |                                                                                                                                                                                                                                |
| <b>System Formats</b>                   | RFC-5424, CEF, LEEF                                                                                                                                                                                      |                                                                                                                                                                                                          |                                                                                                                                                                                                          |                                                                                                                                                                                                                                |
| <b>Supported HSMs for Root of Trust</b> | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM, nShield Network HSM, Google Cloud HSM | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM, nShield Network HSM, Google Cloud HSM | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM, nShield Network HSM, Google Cloud HSM | <b>Built-in HSM</b> , Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM, nShield Network HSM, Google Cloud HSM |
| <b>Automated Deployment Support</b>     | Yes (via Terraform, Cloud-Init)                                                                                                                                                                          | Yes (via Terraform, Cloud-Init)                                                                                                                                                                          | No                                                                                                                                                                                                       | Yes (via Secure Transport Mode)                                                                                                                                                                                                |
| <b>Maximum Number of Keys</b>           | Tested up to 1M Keys (more possible with appropriately sized virtual environments)                                                                                                                       | Tested up to 1M Keys (more possible with appropriately sized virtual environments)                                                                                                                       | 1 Million Keys                                                                                                                                                                                           | 1 Million Keys                                                                                                                                                                                                                 |
| <b>Maximum Domains (multi-tenancy)</b>  | 100                                                                                                                                                                                                      | 1000                                                                                                                                                                                                     | 1000                                                                                                                                                                                                     | 1000                                                                                                                                                                                                                           |
| <b>FIPS Support</b>                     | FIPS 140-2 L1 <a href="#">[Certificate #4430]</a>                                                                                                                                                        |                                                                                                                                                                                                          |                                                                                                                                                                                                          |                                                                                                                                                                                                                                |
|                                         | Integrates with an external FIPS Certified Physical or Cloud HSM as Secure Root of Trust                                                                                                                 |                                                                                                                                                                                                          |                                                                                                                                                                                                          | Embedded PCI-HSM<br>FIPS140-2 Level 3&<br>FIPS140-3 Level 3 Certified                                                                                                                                                          |

# Appliance Specifications

| Physical Appliances       | k470                                                                    | k570          |
|---------------------------|-------------------------------------------------------------------------|---------------|
| Dimensions                | "19" x 21" x 1.75" (482.6mm x 533.4mm x 44.45mm)"                       |               |
| Hard Drive                | 1x 2TB SATA SE (Spinning Disk)                                          |               |
| CPU                       | Cores: 4, Threads: 8, Processor Base Frequency: 3.8 GHz                 |               |
| RAM                       | 16 GB *                                                                 |               |
| NIC Support               | 4x 1 GB or 2x 10Gb/2x 1 Gb (NIC Bonding capable)                        |               |
| Rack Mount                | Standard 1U rack mountable<br>Sliding rails can be optionally purchased |               |
| Reliability               | Dual hot swappable power supplies                                       |               |
| Safety and Compliance     | CSA C-US, FCC, CE, VCCI, C-TICK, KC Mark, BIS                           |               |
| Mean Time Between Failure | 165,279 hours                                                           | 153,583 hours |

| Virtual Appliances           | k170v                                                                                                                                                                                                                                                                                                                                                                                                                                      | K470v                                                                                                                                                         |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System Requirements          | <ul style="list-style-type: none"> <li>RAM (GB): 16</li> <li>Hard Disk (GB): 100</li> <li>NICs: 1 or more</li> <li>CPUs: up to 4 CPU max</li> </ul>                                                                                                                                                                                                                                                                                        | <ul style="list-style-type: none"> <li>RAM (GB): 16 or more</li> <li>Hard Disk (GB): 200 or more</li> <li>NICS: 2 or more</li> <li>CPUs: 5 or more</li> </ul> |
| Clouds/Hypervisors Supported | <ul style="list-style-type: none"> <li>Public Clouds: AWS Cloud, Microsoft Azure, Google Cloud Enterprise (GCE), Oracle Cloud Infrastructure (OCI), Alibaba Cloud</li> <li>Private Clouds/Hypervisors: VMware vSphere (6.5, 6.7 and 7.0), Microsoft Hyper-V, Nutanix AHV, OpenStack (QCOW2)</li> <li>* AWS GovCloud, Azure Government Cloud also supported</li> <li>Hybrid Clouds/Hypervisors: Azure Stack HCI, Azure Stack Hub</li> </ul> |                                                                                                                                                               |

| Safety Certifications                                                           | Applicable Administrative Unit |
|---------------------------------------------------------------------------------|--------------------------------|
| CB Scheme                                                                       | 50 countries                   |
| CSA-UL                                                                          | Canada/US                      |
| Emissions Certifications                                                        |                                |
| FCC Part 15, Subpart B, Class B                                                 | US                             |
| EN55032:2010, EN55035:2017, EN61000-3-2:2006 +A1:2009 +A2:2009 EN61000-3-3:2008 | EU                             |
| ICES-003 Issue 7 - October 2020                                                 | Canada                         |
| AS/NZ CISPR 32:2015                                                             | Australia/NZ                   |
| VCCI V-3/2009.04                                                                | Japan                          |
| KN22, KN24, KC Mark                                                             | South Korea                    |
| NOM                                                                             | Mexico                         |
| BIS                                                                             | India                          |

\* 16 GB or more

# CipherTrust Manager



## Overview

CipherTrust Manager enables organizations to centrally manage encryption keys for Thales CipherTrust Data Security Platform and third party products. 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, and robust auditing and reporting of all key management and encryption operations.

CipherTrust Manager is the central management point for the [CipherTrust Data Security Platform](#). It provides a unified management console that makes it easy to discover and classify data, and to protect sensitive data wherever it resides using a comprehensive set of CipherTrust Data Protection Connectors from Thales.

CipherTrust Manager is available in both virtual and physical form-factors that integrate with FIPS 140-2 validated Thales Luna Network or Cloud HSM, and third-party Hardware Security Modules (HSMs) for securely storing master keys with highest root of trust. These appliances can be deployed on-premises as well as in private or public cloud infrastructures. This allows customers to address compliance requirements, regulatory mandates and industry best practices for data security.

## Benefits

- Centralized key and policy management for on-premises data stores and cloud infrastructures
- Reduced business risk with unified data discovery, classification and sensitive data protection
- Simplified management with self-service licensing portal and visibility into licenses in use
- Cloud friendly deployment options with support for AWS, Azure, Google Cloud, VMware, Oracle Cloud Infrastructure and more
- Expanded Hardware Security Module (HSM) support for superior key control
- Unparalleled partner ecosystem of integrations with leading enterprise storage, server, database, application and SaaS vendors



Key Management



Access Policies



Auditing Reporting



Flexible APIs



CipherTrust Manager



## Key Capabilities

- **Full Key Lifecycle Management and Automated Operations:** CipherTrust Manager simplifies management of encryption keys across their entire lifecycle, including secure key generation, backup/restore, clustering, deactivation, and deletion. It makes automated, policy-driven operations easy to perform, and generates alarms for events of interest.
- **Unified Management Console:** Provides a unified console for discovering and classifying sensitive data integrated with a comprehensive set of CipherTrust Data Protection Connectors to encrypt or tokenize data to reduce business risk and satisfy compliance regulations.
- **Centralized Administration and Access Controls:** Unifies key management operations with role-based access controls. Authenticates and authorizes administrators and key users using existing AD and LDAP credentials. Prevents unauthorized password change and alerts on simultaneous logins by same user.
- **Self-service Licensing:** Streamlines provisioning of connector licenses through a new customer facing licensing portal. The new management console offers better visibility and control of licenses in use.
- **Secrets Management:** Provides the ability to create and manage secrets and opaque objects for usage on the platform.
- **Multi-tenancy Support:** Supports separation of duties with delegated user management within multiple domains.
- **Developer Friendly REST APIs:** Offers new REST interfaces, in addition to Key Management Interoperability Protocol (KMIP) and NAE-XML APIs, allows customers to remotely generate and manage keys.
- **Flexible HA Clustering and Intelligent Key Sharing:** Provides the option of clustering physical and / or virtual appliances together to assure high availability as well as increased encryption transaction throughput.
- **Robust Auditing and Reporting:** Includes tracking of all key state changes, administrator access, and policy changes in multiple log formats (RFC-5424, CEF, LEEF) for easy integration with SIEM tools.
- **Broad Partner Ecosystem:** CipherTrust Manager provides centralized key management for wide variety of storage partners via KMIP and database partners via Transparent Database Encryption (TDE).

## CipherTrust Manager Features

| Features                         | Virtual Appliances                                                                                                                                                |                                                                                                                                                                   | Physical Appliances                                                                                                                                               |                                 |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
|                                  | k170v                                                                                                                                                             | k470v                                                                                                                                                             | k470                                                                                                                                                              | k570                            |
| Administrative Interfaces        | Management Console, REST API, kscfg (system configuration), (ksctl (Command Line Interface)                                                                       |                                                                                                                                                                   |                                                                                                                                                                   |                                 |
| Network Management               | SNMP v1, v2c, v3, NTP, Syslog-TCP                                                                                                                                 |                                                                                                                                                                   |                                                                                                                                                                   |                                 |
| API Support                      | REST, NAE-XML, KMIP, PKCS#11, JCE, .NET, MCCAPI, MS CNG                                                                                                           |                                                                                                                                                                   |                                                                                                                                                                   |                                 |
| Secure Authentication            | Local User , AD/LDAP, LDAPS, Certificate based authentication, Supports Open ID Connect (OIDC)                                                                    |                                                                                                                                                                   |                                                                                                                                                                   |                                 |
| System Formats                   | RFC-5424, CEF, LEEF                                                                                                                                               |                                                                                                                                                                   |                                                                                                                                                                   |                                 |
| Supported HSMs for Root of Trust | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM | N/A (has built in HSM)          |
| Automated Deployment Support     | Yes (via Cloud-Init)                                                                                                                                              | Yes (via Cloud-Init)                                                                                                                                              | No                                                                                                                                                                | Yes (via Secure Transport Mode) |
| Maximum Number of Keys           | Tested up to 1M Keys (more possible with appropriately sized virtual environments)                                                                                | Tested up to 1M Keys (more possible with appropriately sized virtual environments)                                                                                | 1 Million Keys                                                                                                                                                    | 1 Million Keys                  |
| Maximum Domains (multi-tenancy)  | 100                                                                                                                                                               | 1000                                                                                                                                                              | 1000                                                                                                                                                              | 1000                            |

## Appliance Specifications

| Physical Appliances          | k470                                                                                                                                                                                                                                                                                                                                                                    | k570                                                                                                                                                                  |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dimensions                   | 19" x 21" x 1.725" (482.6mm x 533.4mm x 43.815mm)                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                       |
| Hard Drive                   | 1x 2TB SATA SE (Spinning Disk)                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                       |
| CPU                          | Xeon E3- 1275v6 Processor                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                       |
| RAM                          | 16GB                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                       |
| NIC Support                  | 4x1GB or 2x10Gb/2x1Gb (NIC Bonding capable)                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                       |
| Rack Mount                   | Standard 1U rack mountable<br>Sliding rails can be optionally purchased                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                       |
| Reliability                  | Dual hot swappable power supplies                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                       |
| Safety and Compliance        | CSA C-US, FCC, CE, VCCI, C-TICK, KC Mark, BIS                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                       |
| Mean Time Between Failure    | 165,279 hours                                                                                                                                                                                                                                                                                                                                                           | 153,583 hours                                                                                                                                                         |
| FIPS Support                 | Integrates with an external FIPS Certified Physical or Cloud HSM as Secure Root of Trust                                                                                                                                                                                                                                                                                | Embedded PCI-HSM FIPS 140-2 Level 3 certified – password and multi-factor (PED) (Certificate #3205)                                                                   |
| Virtual Appliances           | k170v                                                                                                                                                                                                                                                                                                                                                                   | K470v                                                                                                                                                                 |
| System Requirements          | <ul style="list-style-type: none"> <li>• RAM (GB): 16</li> <li>• Hard Disk (GB): 100</li> <li>• NICs: 1 or more</li> <li>• CPUs: up to 4 CPU max</li> </ul>                                                                                                                                                                                                             | <ul style="list-style-type: none"> <li>• RAM (GB): 16 or more</li> <li>• Hard Disk (GB): 200 or more</li> <li>• NICs: 2 or more</li> <li>• CPUs: 5 or more</li> </ul> |
| Clouds/Hypervisors Supported | <ul style="list-style-type: none"> <li>• Public Clouds: AWS Cloud, Microsoft Azure, Google Cloud Enterprise (GCE), Oracle Cloud Infrastructure (OCI), Alibaba Cloud</li> <li>• Private Clouds/Hypervisors: VMware vSphere (6.5, 6.7 and 7.0), Microsoft Hyper-V, Nutanix AHV, OpenStack (QCOW2)</li> </ul> <p>* AWS GovCloud, Azure Government Cloud also supported</p> |                                                                                                                                                                       |

# CipherTrust Manager

## Centralize and simplify data security policies and key management anywhere



### Overview

CipherTrust Manager enables organizations to centrally manage encryption keys for Thales CipherTrust Data Security Platform and third party products. 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, and robust auditing and reporting of all key management and encryption operations.

CipherTrust Manager is the central management point for the CipherTrust Data Security Platform. It provides a unified management console that makes it easy to discover and classify data, and to protect sensitive data wherever it resides using a comprehensive set of CipherTrust Data Protection Connectors from Thales.

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. These appliances can be deployed on-premises as well as in private or public cloud infrastructures. This allows customers to address compliance requirements, regulatory mandates and industry best practices for data security.

### Benefits

- Centralized key management for multiple on-premises data stores and cloud infrastructures
- Reduced risk with unified data discovery, classification and sensitive data protection
- Simplified management with self-service licensing portal and visibility into licenses in use
- Cloud friendly deployment options with support for AWS, Azure, Google Cloud, VMware, Oracle Cloud Infrastructure and more
- Support for superior key control with Thales TCT's T-Series HSM
- Unparalleled partner ecosystem of integrations with leading enterprise storage, server, database, application and SaaS vendors



Key  
Management



Access  
Control



Auditing  
Reporting



REST  
APIs



CipherTrust Manager



## Key Capabilities

- **Full Key Lifecycle Management and Automated Operations:** CipherTrust Manager simplifies management of encryption keys across their entire lifecycle, including secure key generation, backup/restore, clustering, deactivation, and deletion. It makes automated, policy-driven operations easy to perform, and generates alarms for events of interest.
- **Unified Management Console:** Provides a unified console for discovering and classifying sensitive data integrated with a comprehensive set of CipherTrust Data Protection Connectors to encrypt or tokenize data to reduce business risk and satisfy compliance regulations.
- **Centralized Administration and Access Control:** Unifies key management operations with role-based access controls and provides full audit log review. Authenticates and authorizes administrators and key users using existing AD and LDAP credentials.
- **Secrets Management:** Provides the ability to create and manage secrets and opaque objects for usage on the platform.
- **Multi-tenancy Support:** Provides capabilities required to create multiple domains with separation of duties to support large organizations with distributed locations.
- **Developer Friendly REST APIs:** Offers new REST interfaces, in addition to Key Management Interoperability Protocol (KMIP) and NAE-XML APIs, allows customers to remotely generate and manage keys.
- **Flexible HA Clustering and Intelligent Key Sharing:** Provides the option of clustering physical and / or virtual appliances together to assure high availability as well as increased encryption transaction throughput.
- **Robust Auditing and Reporting:** Includes tracking of all key state changes, administrator access, and policy changes in multiple log formats (RFC-5424, CEF, LEEF) for easy integration with SIEM tools.
- **Broad Partner Ecosystem:** CipherTrust Manager provides centralized key management for wide variety of storage partners via KMIP and database partners via Transparent Database Encryption (TDE).
- **Root of Trust:** CipherTrust Manager can use Thales TCT's Luna T-Series HSMs as root of trust. Meeting government mandates for U.S. Supply Chain, the high-assurance, tamper-resistant Luna T-Series HSMs are designed, developed, manufactured, sold, and supported in the United States. CipherTrust k160 uses a removable FIPS 140-2 certified token or high assurance token as a root of trust.

## CipherTrust Manager Features

|                                  | Virtual Appliances                                                                                                                                                |                                                                                                                                                                   | Physical Appliances                                                                                 |                                                             |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| Features                         | k170v                                                                                                                                                             | k470v                                                                                                                                                             | k160                                                                                                | k570*                                                       |
| Administrative Interfaces        | Management Console, REST API, kscfg (system configuration), (ksctl (Command Line Interface)                                                                       |                                                                                                                                                                   |                                                                                                     |                                                             |
| Network Management               | SNMP v1, v2c, v3, NTP, Syslog-TCP                                                                                                                                 |                                                                                                                                                                   |                                                                                                     |                                                             |
| API Support                      | REST, NAE-XML, KMIP, PKCS# 11, JCE, .NET, MCCAPI, MS CNG                                                                                                          |                                                                                                                                                                   |                                                                                                     |                                                             |
| Security Authentication          | Local User , AD/LDAP , Certificate based authentication                                                                                                           |                                                                                                                                                                   |                                                                                                     |                                                             |
| System Formats                   | RFC-5424, CEF, LEEF                                                                                                                                               |                                                                                                                                                                   |                                                                                                     |                                                             |
| Supported HSMs for Root of Trust | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM | Removable token HSM using either a FIPS 140-2 Certified Token or High Assurance Token               | Embedded FIPS 140 level 3 & CNSS approved Luna T-series HSM |
| Maximum Number of Keys           | Tested up to 1M Keys (more possible with appropriately sized virtual environments)                                                                                | Tested up to 1M Keys (more possible with appropriately sized virtual environments)                                                                                | Maximum capacity of 10,000 symmetric keys.<br><br>Maximum of 100 keys using concurrent connections. | 1 Million Keys                                              |
| Maximum Domains (multi-tenancy)  | 100                                                                                                                                                               | 1000                                                                                                                                                              | 100                                                                                                 | 1000                                                        |

\*Thales TCT can also offer CipherTrust k470 physical appliance. CipherTrust k470 offers the same features and specifications as CipherTrust k570 but does not include an embedded HSM. CipherTrust k470 utilizes an external FIPS Certified Physical or Cloud HSM as secure root of trust.

## Appliance Specifications

| Physical Appliances       | k160                                                                                                                               | k570                                                                    |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| Dimensions                | 6.5" x 4.0" x 1.5" (165.1mm x 101.6mm x 38.1mm)                                                                                    | 19" x 21" x 1.725" (482.6mm x 533.4mm x 43.815mm)                       |
| Hard Drive                | 1x 128GB mSATA SSD SE                                                                                                              | 1x 2TB SATA SE (Spinning Disk)                                          |
| CPU                       | Atom E3845 Processor SoC                                                                                                           | Xeon E3-1275v6 Processor                                                |
| RAM                       | 8GB                                                                                                                                | 16GB                                                                    |
| NIC Support               | 1x 1GB                                                                                                                             | 4x1GB or 2x10Gb/2x1Gb (NIC Bonding capable)                             |
| Rack Mount                | Standard 1U shelf mount can be optionally purchased (can house up to two k160s)                                                    | Standard 1U rack mountable<br>Sliding rails can be optionally purchased |
| Power                     | External power supply included, locking DC connector                                                                               | Dual hot swappable power supplies                                       |
| Safety and Compliance     | FCC, CE                                                                                                                            | CSA C-US, FCC, CE, VCCI, C-TICK, KC Mark, BIS                           |
| Mean Time Between Failure | 170,869 hours                                                                                                                      | 153,583 hours                                                           |
| FIPS Support              | Virtual appliance is FIPS 140 level 1 (pending). Integrates with 3rd party HSMs including FIPS 140 Level 3 validated Luna T-Series | Embedded FIPS 140 Level 3 & CNSS approved Luna T-series HSM             |

| Virtual Appliances           | k170v                                                                                                                                                                                                                                                                                                                                               | k470v                                                                                                                                                         |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System Requirements          | <ul style="list-style-type: none"> <li>RAM (GB): 16</li> <li>Hard Disk (GB): 100</li> <li>NICs: 1 or more</li> <li>CPUs: up to 4 CPU max</li> </ul>                                                                                                                                                                                                 | <ul style="list-style-type: none"> <li>RAM (GB): 16 or more</li> <li>Hard Disk (GB): 200 or more</li> <li>NICS: 2 or more</li> <li>CPUs: 5 or more</li> </ul> |
| Clouds/Hypervisors Supported | <ul style="list-style-type: none"> <li>Public Clouds: AWS Cloud, Microsoft Azure, Google Cloud Enterprise (GCE), Oracle Cloud Infrastructure (OCI)</li> <li>Private Clouds/Hypervisors: VMware vSphere (6.5, 6.7 and 7.0), Microsoft Hyper-V, Nutanix AHV, OpenStack (QCOW2)</li> </ul> <p>*AWS GovCloud, Azure Government Cloud also supported</p> |                                                                                                                                                               |

## 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](http://www.thalestct.com)

# Acelerando a Conformidade com a Lei Geral de Proteção de Dados do Brasil (LGPD)



# Contents

- 3** **LGPD: Um divisor de águas para os direitos de privacidade brasileiros**
- 4** **Artigos relevantes a proteção de dados**
- 6** **Transformação digital, nuvem, big data e o desafio da conformidade**
- 7** **Melhores práticas e as soluções da Thales Cloud Protection & Licensing**
- 7** **Entre as melhores práticas de conformidade e segurança para uma empresa com infraestrutura de TI híbrida incluímos quatro aspectos que se tornam essenciais no processo de conformidade com a LGPD**

# LGPD: Um divisor de águas para os direitos de privacidade brasileiros

A Lei Geral de Proteção de Dados (LGPD) do Brasil foi sancionada em 14 de agosto de 2018, e entrou em vigor em agosto de 2020.



A LGPD cria uma nova estrutura legal para o uso de dados pessoais no Brasil, tanto online quanto off-line, nos setores público e privado. A LGPD está substituindo ou complementando várias leis e regulamentações já existentes, algumas das quais estavam em conflito ou não tinham segurança jurídica. A LGPD torna o país mais competitivo no contexto de uma sociedade cada vez mais impulsionada pela transformação digital.

## De acordo com o Artigo 1 da Lei:

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.

Ao promulgar a Lei Geral de Proteção de Dados, o Brasil pode ser considerado como um país com nível adequado de proteção à privacidade e ao uso de dados pessoais.

## Ampla aplicabilidade e pesadas penalidades

De maneira semelhante ao General Data Protection Regulation (GDPR) da União Europeia, a LGPD terá aplicação extraterritorial, ou seja, o dever de cumprimento excederá os limites geográficos do país. Qualquer empresa estrangeira que tenha pelo menos uma filial no Brasil, ou ofereça serviços para o mercado brasileiro e colete e processe dados pessoais de titulares de dados localizados no país, independentemente da nacionalidade, estará sujeita à nova lei.

E como o GDPR, as penalidades pesadas podem ser aplicadas às organizações que não cumprem a lei. Entre as sanções, há autuações e multas, que podem variar de 2% do faturamento da empresa no Brasil (limitado no total a 50 milhões de reais) por infração. Há também a possibilidade de multa diária e a proibição parcial ou total do exercício de atividades relacionadas ao tratamento de dados.

# Artigos relevantes a proteção de dados

Os trechos a seguir foram extraídos da Lei Brasileira nº 13.709 (LGPD), de 14 de agosto de 2018.<sup>1</sup>

## Aplicação

Art. 3. Esta Lei aplica-se a qualquer operação de tratamento realizada por pessoa natural

ou por pessoa jurídica de direito público ou privado, independentemente do meio, do país de sua sede ou do país onde estejam localizados os dados ....

## Dados anonimizados

Art. 5. Para fins desta Lei, aplicam-se as seguintes definições:

ou por pessoa jurídica de direito público ou privado, independentemente do meio, do país de sua sede ou do país onde estejam localizados os dados ....

III - dado anonimizado: dado relativo ao titular que não possa ser identificado, considerando a utilização de meios técnicos razoáveis e disponíveis na ocasião de seu tratamento;

XI - anonimização: 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;

## Tratamento de dados

Art. 6. As atividades de tratamento de dados pessoais deverão observar a boa-fé e os seguintes princípios:

VII - segurança: utilização 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 difusão;

VIII - prevenção: adoção de medidas para prevenir a ocorrência de danos em virtude do tratamento de dados pessoais;

X - responsabilização e prestação de contas: demonstração, pelo agente, da adoção de medidas eficazes e capazes de comprovar a observância e o cumprimento das normas de proteção de dados pessoais e, inclusive, da eficácia dessas medidas.

## Dados anonimizados não considerados dados pessoais

Art. 12. Os dados anonimizados não serão considerados dados pessoais para os fins desta Lei, salvo quando o processo de anonimização ao qual foram submetidos for revertido, utilizando exclusivamente meios próprios, ou quando, com esforços razoáveis, puder ser revertido.

§ 1º A determinação do que seja razoável deve levar em consideração fatores objetivos, tais como custo e tempo necessários para reverter o processo de anonimização, de acordo com as tecnologias disponíveis, e a utilização exclusiva de meios próprios.

Art. 13. §4º. "Para os efeitos deste artigo, a pseudonimização é o tratamento por meio do qual um dado perde a possibilidade de associação, direta ou indireta, a um indivíduo, senão pelo uso de informação adicional mantida separadamente pelo controlador em ambiente controlado e seguro"

## Responsabilidade e registro de acesso

Art. 37. O controlador e o operador devem manter registro das operações de tratamento de dados pessoais que realizarem, especialmente quando baseado no legítimo interesse.

Art. 38. A autoridade nacional poderá determinar ao controlador que elabore relatório de impacto à proteção de dados pessoais, inclusive de dados sensíveis, referente a suas operações de tratamento de dados, nos termos de regulamento, observados os segredos comercial e industrial

Parágrafo único. Observado o disposto no caput deste artigo, o relatório deverá conter, no mínimo, uma descrição dos tipos de dados coletados, a metodologia utilizada para a coleta e para a garantia da segurança das informações e análise do controlador com relação às medidas, salvaguardas e mecanismos de mitigação de risco adotados.

# Artigos relevantes a proteção de dados

## Responsabilidade do controlador e dos processadores

Art. 42. O controlador ou o operador que, em razão do exercício de atividade de tratamento de dados pessoais, causar a outrem dano patrimonial, moral, individual ou coletivo, em violação à legislação de proteção de dados pessoais, é obrigado a repará-lo.

## Da segurança e do sigilo de dados

Art. 46. Os agentes de tratamento devem adotar medidas de segurança, 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.

## Notificação de vazamento de dados

Art. 48. O controlador deverá comunicar à autoridade nacional e ao titular a ocorrência de incidente de segurança que possa acarretar risco ou dano relevante aos titulares.

## Melhores práticas de segurança de dados

Art. 49. Os sistemas utilizados para o tratamento de dados pessoais devem ser estruturados de forma a atender aos requisitos de segurança, aos padrões de boas práticas e de governança e aos princípios gerais previstos nesta Lei e às demais normas regulamentares.

## Sanções administrativas

Art. 52. Os agentes de tratamento de dados, em razão das infrações cometidas às normas previstas nesta Lei, ficam sujeitos às seguintes sanções administrativas aplicáveis pela autoridade nacional:

- advertência, com indicação de prazo para adoção de medidas corretivas;
- multa simples, de até 2% (dois por cento) do faturamento da pessoa jurídica de direito privado, grupo ou conglomerado no Brasil no seu último exercício, excluídos os tributos, limitada, no total, a 50 milhões de reais por infração;
- multa diária, observado o limite total a que se refere o inciso II;
- publicização da infração após devidamente apurada e confirmada a sua ocorrência;
- bloqueio dos dados pessoais a que se refere a infração até a sua regularização;
- eliminação dos dados pessoais a que se refere a infração;

# Transformação digital, nuvem, big data e o desafio da conformidade



A Lei Geral de Proteção de Dados (LGPD) do Brasil exige as melhores práticas em segurança para dados pessoais e observa que os dados pessoais que tiverem sido anonimizados não serão considerados no escopo da Lei, se não puderem ser revertidos facilmente para o seu estado original por aqueles que possam obtê-los.

O desafio para a empresa moderna, é como equilibrar a necessidade de usar estes dados para processos, análises de negócios e ultimamente o sucesso no mercado, com a proteção aos dados confidenciais contra hackers, pessoas de dentro da organização ou vazamentos acidentais.

Por exemplo, os dados têm que fluir entre várias plataformas na nuvem e nas dependências da empresa, além de serem acessados por funcionários remotos e analisados em plataformas de big data. Durante todo o processo, os dados devem estar seguros para permanecer em conformidade com a LGPD e leis semelhantes.

Diante da transformação digital, as atuais estratégias de segurança usadas por empresas se tornam insuficientes. À medida que as empresas implementam novas tecnologias como nuvem e big data, muitas vezes escolhem soluções pontuais para proteger dados em uma plataforma ou outra. Isso leva a uma abordagem de segurança fragmentada e a falhas na proteção de dados.



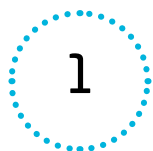
# Melhores práticas e as soluções da Thales Cloud Protection & Licensing

Entre as melhores práticas de conformidade e segurança para uma empresa com infraestrutura de TI híbrida incluímos quatro aspectos que se tornam essenciais no processo de conformidade com a LGPD

- Busca e proteção de dados sensíveis em ambientes de infraestrutura híbridos.
- Anonimização e / ou pseudonimização de dados pessoais
- Autenticação segura e centralizada
- Controle de acesso a dados sensíveis
- Monitoramento e registros de acessos aos dados

A Thales Cloud Protection & Licensing tem ampla experiência em conformidade com legislações como GDPR ou regulamentos como PCI. Além disso, a Thales é extremamente experiente em implementações abrangentes para corporações globais, em múltiplas plataformas de TI híbridas.

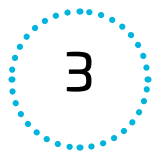
Nossas soluções oferecem aos clientes recursos para:



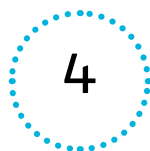
Busca e classificação de dados confidenciais



Anonimização e pseudonimização de dados



Gerenciamento das chaves usadas para criptografia de dados



Controle de acesso do usuário a dados



Registros de eventos de acesso a dados



Proteção de dados em ambientes de infraestrutura híbridos

# 1. Busca e classificação de dados confidenciais

## **Descoberta de dados, classificação e análise de risco**

Thales CipherTrust Data Discovery and Classification ajuda sua organização a obter visibilidade completa de seus dados confidenciais com eficiente descoberta de dados, classificação e análise de risco em armazenamentos de dados heterogêneos - nuvem, big data e ambientes tradicionais - em sua empresa.

Simples de implantar e usar, ele oferece um único painel de controle que permite obter uma compreensão clara de quais dados confidenciais você possui, onde estão localizados e os riscos de exposição. Com visualizações ricas e relatórios detalhados, você pode descobrir e fechar suas lacunas com mais facilidade, tomar melhores decisões sobre o compartilhamento de dados de terceiros e migração para a nuvem.

# 2. Anonimização e pseudonimização de dados

## **Criptografia transparente de dados em repouso**

A solução CipherTrust Transparent Encryption da Thales protege os dados com criptografia para dados em repouso de arquivo e nível de volume, controles de acesso e registro de auditoria de acesso a dados sem aplicativos de reengenharia, banco de dados ou infraestrutura. A instalação do software de criptografia transparente de arquivos é simples, escalável e rápida, com agentes instalados acima do sistema de arquivo em servidores ou máquinas virtuais para assegurar políticas de segurança e conformidade de dados. O gerenciamento de política e criptografia de chaves é fornecido pela CipherTrust Data Security Manager.

## **Tokenização com mascaramento dinâmico de dados**

A solução CipherTrust Vaultless Tokenization com Mascaramento Dinâmico de dados reduz drasticamente os custos e os esforços necessários para cumprir com as políticas de segurança e normas regulatórias, como a LGPD. A solução oferece capacidades para tokenização de banco de dados e segurança de tela dinâmica. As empresas podem eficientemente realizar seus objetivos de proteger e pseudoanonimizar ativos sensíveis — estejam eles em ambiente de data center, big data, em container ou na nuvem.

## **Módulos de Segurança de Hardware (HSMs)**

Os Módulos de Segurança de Hardware (HSMs) Thales Luna são certificados FIPS 140-2 nível 3, e INMETRO (ITI ICP-BRASIL), com uma ampla gama de usos para acelerar operações criptográficas, proteger o ciclo de vida de chave criptográfica e fornecer uma base de confiança para toda a sua infraestrutura de criptografia. A premiada solução Thales Data Protection On Demand é uma plataforma baseada em nuvem que fornece uma ampla gama de serviços de gerenciamento de chaves e HSM na nuvem por meio de um simples mercado on-line. Isso inclui o HSM como serviço e o Key Management como serviço.

# 3. Gerenciamento das chaves usadas para criptografia de dados

## **Gerenciamento de chaves centralizado**

A solução Enterprise Key Management unifica e centraliza o gerenciamento de criptografia de chaves no local e fornece gerenciamento seguro de chaves para soluções de armazenamento de dados. Os produtos para gerenciamento de chave na nuvem incluem o CipherTrust Cloud Key Manager.

# 4. Controle de acesso do usuário e autenticação

## **Gerenciamento de acesso baseado em nuvem como serviço**

O SafeNet Trusted Access simplifica o login dos usuários, pois é um serviço de gerenciamento de acesso que protege e gerencia centralmente o acesso a aplicativos baseados na web ou na nuvem. Empresas podem escalar os controles de acesso à nuvem enquanto atendem às necessidades dos negócios, de gerenciamento de riscos e de conformidade ao aplicar políticas flexíveis baseadas em risco, SSO na nuvem e métodos de autenticação universal.

## **Gerenciamento de identidades e acesso**

Com as soluções Thales Identity and Access Management, empresas podem proteger facilmente os aplicativos na nuvem ou no centro de dados, e atender às necessidades de gerenciamento de risco com base em suas estruturas de segurança atuais e utilizando os métodos de autenticação existentes para acesso à informação.

## Gerenciamento de acesso a dados baseado em criptografia

A solução CipherTrust Data Security Manager da Thales permite que a organização limite os privilégios de acesso de usuários a sistemas de informações que contenham informações sensíveis.

## 5. Registro de acesso ao banco de dados

### Security Intelligence Logs

A solução Security Intelligence Logs da Plataforma CipherTrust permite que sua organização identifique tentativas de acesso não autorizado e desenvolva parâmetros de modelos de acesso do usuário autorizado. A CipherTrust Security Intelligence integra-se aos principais sistemas de gerenciamento de eventos e informações de segurança (SIEM) que tornam essas informações acionáveis.

## 6. Proteção de dados em ambientes de infraestrutura híbridos

### Soluções de segurança BYOE e BYOK para nuvem

A Thales oferece várias maneiras de proteger dados em plataformas na nuvem. Organizações podem trazer sua própria criptografia para a nuvem (BYOE - Bring Your Own Encryption) usando nossa solução CipherTrust Transparent Encryption. Clientes também podem adicionar camada de segurança às suas implementações BYOK (Bring Your Own Key), usando o CipherTrust Cloud Key Manager (CCKM) para obter visibilidade e gerenciamento centralizados de ciclo de vida de várias nuvens, com armazenamento seguro de chaves FIPS-140-2.

### Plataforma de proteção de dados para infraestrutura híbrida

A CipherTrust Data Security Platform é uma plataforma extremamente flexível e escalonável, oferecendo proteção e controle de acesso a bancos de dados, arquivos e contêineres - e pode proteger dados que residem em ambientes de nuvem, virtuais, big data e físicos. As plataformas e tecnologias suportadas inclui:

- IaaS, PaaS e SaaS: Amazon Web Services, Google Cloud Platform, Microsoft Azure, Salesforce, Microsoft Office365
- OSs: Linux, Windows and Unix
- Big data: Hadoop, NoSQL, SAP HANA e Teradata
- Database: IBM DB2, Microsoft SQL Server, MongoDB, MySQL, NoSQL, Oracle, Sybase entre outros

Aonde quer que você opere e qualquer que seja o regulamento, você pode confiar na Thales para ajudá-lo a gerenciar o seu risco. A Thales pode ajudar a sua organização a cumprir com muitos dos requisitos da LGPD.

## Recursos relacionados

CipherTrust Data Security Platform Overview<sup>2</sup>

CipherTrust Transparent Encryption<sup>3</sup>

CipherTrust Tokenization com Mascaramento Dinâmico de dados<sup>4</sup>

Gerenciamento Centralizado de Chaves<sup>5</sup>

CipherTrust Cloud Key Manager<sup>6</sup>

Information and Access Management Solutions<sup>7</sup>

Thales Trusted Access: Cloud-based Access Management<sup>8</sup>

Thales Data Protection on Demand<sup>9</sup>

Thales Luna Network HSMs<sup>10</sup>

CipherTrust Security Intelligence com Integração SIEM<sup>11</sup>

<sup>2</sup> <https://cpl.thalesgroup.com/es/encryption/data-security-platform>

<sup>3</sup> <https://cpl.thalesgroup.com/es/encryption/transparent-encryption>

<sup>4</sup> <https://cpl.thalesgroup.com/es/encryption/tokenization>

<sup>5</sup> <https://cpl.thalesgroup.com/es/encryption/key-management>

<sup>6</sup> <https://cpl.thalesgroup.com/encryption/key-management/ciphertrust-cloud-key-manager>

<sup>7</sup> <https://cpl.thalesgroup.com/es/access-management>

<sup>8</sup> <https://cpl.thalesgroup.com/es/access-management/safenet-trusted-access>

<sup>9</sup> <https://cpl.thalesgroup.com/encryption/data-protection-on-demand>

<sup>10</sup> <https://cpl.thalesgroup.com/encryption/hardware-security-modules/network-hsms>

<sup>11</sup> <https://cpl.thalesgroup.com/encryption/security-intelligence-logs>

## About Thales

As pessoas em quem você confia para proteger sua privacidade confiam na Thales para proteger sua nuvem, seus dados e software. Em termos de segurança de dados, as organizações enfrentam cada vez mais momentos decisivos. Seja na criação de uma estratégia de criptografia, licenciamento de software, fornecimento de acesso seguro à nuvem ou cumprimento de normas de conformidade, você pode confiar na Thales para proteger sua transformação digital.

Tecnologia decisiva para momentos decisivos.

# THALES

### Thales – Latin America / Brazil

Rua Surubim, 577, 12º. andar, cjs.  
123/124 Brooklin  
São Paulo - SP - BRASIL 04571-050

Tel: +55 11 3198-5250 | Email: [CPL\\_Sales\\_AMS\\_TG@thalesgroup.com](mailto:CPL_Sales_AMS_TG@thalesgroup.com)

[> cpl.thalesgroup.com <](https://cpl.thalesgroup.com)



# Acelerando a Conformidade com a Lei Geral de Proteção de Dados do Brasil (LGPD)



# Contents

- 3** **LGPD: Um divisor de águas para os direitos de privacidade brasileiros**
- 4** **Artigos relevantes a proteção de dados**
- 6** **Transformação digital, nuvem, big data e o desafio da conformidade**
- 7** **Melhores práticas e as soluções da Thales Cloud Protection & Licensing**
- 7** **Entre as melhores práticas de conformidade e segurança para uma empresa com infraestrutura de TI híbrida incluímos quatro aspectos que se tornam essenciais no processo de conformidade com a LGPD**

# LGPD: Um divisor de águas para os direitos de privacidade brasileiros

A Lei Geral de Proteção de Dados (LGPD) do Brasil foi sancionada em 14 de agosto de 2018, e entrou em vigor em agosto de 2020.



A LGPD cria uma nova estrutura legal para o uso de dados pessoais no Brasil, tanto online quanto off-line, nos setores público e privado. A LGPD está substituindo ou complementando várias leis e regulamentações já existentes, algumas das quais estavam em conflito ou não tinham segurança jurídica. A LGPD torna o país mais competitivo no contexto de uma sociedade cada vez mais impulsionada pela transformação digital.

## De acordo com o Artigo 1 da Lei:

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.

Ao promulgar a Lei Geral de Proteção de Dados, o Brasil pode ser considerado como um país com nível adequado de proteção à privacidade e ao uso de dados pessoais.

## Ampla aplicabilidade e pesadas penalidades

De maneira semelhante ao General Data Protection Regulation (GDPR) da União Europeia, a LGPD terá aplicação extraterritorial, ou seja, o dever de cumprimento excederá os limites geográficos do país. Qualquer empresa estrangeira que tenha pelo menos uma filial no Brasil, ou ofereça serviços para o mercado brasileiro e colete e processe dados pessoais de titulares de dados localizados no país, independentemente da nacionalidade, estará sujeita à nova lei.

E como o GDPR, as penalidades pesadas podem ser aplicadas às organizações que não cumprem a lei. Entre as sanções, há autuações e multas, que podem variar de 2% do faturamento da empresa no Brasil (limitado no total a 50 milhões de reais) por infração. Há também a possibilidade de multa diária e a proibição parcial ou total do exercício de atividades relacionadas ao tratamento de dados.

# Artigos relevantes a proteção de dados

Os trechos a seguir foram extraídos da Lei Brasileira nº 13.709 (LGPD), de 14 de agosto de 2018.<sup>1</sup>

## Aplicação

Art. 3. Esta Lei aplica-se a qualquer operação de tratamento realizada por pessoa natural

ou por pessoa jurídica de direito público ou privado, independentemente do meio, do país de sua sede ou do país onde estejam localizados os dados ....

## Dados anonimizados

Art. 5. Para fins desta Lei, aplicam-se as seguintes definições:

ou por pessoa jurídica de direito público ou privado, independentemente do meio, do país de sua sede ou do país onde estejam localizados os dados ....

III - dado anonimizado: dado relativo ao titular que não possa ser identificado, considerando a utilização de meios técnicos razoáveis e disponíveis na ocasião de seu tratamento;

XI - anonimização: 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;

## Tratamento de dados

Art. 6. As atividades de tratamento de dados pessoais deverão observar a boa-fé e os seguintes princípios:

VII - segurança: utilização 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 difusão;

VIII - prevenção: adoção de medidas para prevenir a ocorrência de danos em virtude do tratamento de dados pessoais;

X - responsabilização e prestação de contas: demonstração, pelo agente, da adoção de medidas eficazes e capazes de comprovar a observância e o cumprimento das normas de proteção de dados pessoais e, inclusive, da eficácia dessas medidas.

## Dados anonimizados não considerados dados pessoais

Art. 12. Os dados anonimizados não serão considerados dados pessoais para os fins desta Lei, salvo quando o processo de anonimização ao qual foram submetidos for revertido, utilizando exclusivamente meios próprios, ou quando, com esforços razoáveis, puder ser revertido.

§ 1º A determinação do que seja razoável deve levar em consideração fatores objetivos, tais como custo e tempo necessários para reverter o processo de anonimização, de acordo com as tecnologias disponíveis, e a utilização exclusiva de meios próprios.

Art. 13. §4º. "Para os efeitos deste artigo, a pseudonimização é o tratamento por meio do qual um dado perde a possibilidade de associação, direta ou indireta, a um indivíduo, senão pelo uso de informação adicional mantida separadamente pelo controlador em ambiente controlado e seguro"

## Responsabilidade e registro de acesso

Art. 37. O controlador e o operador devem manter registro das operações de tratamento de dados pessoais que realizarem, especialmente quando baseado no legítimo interesse.

Art. 38. A autoridade nacional poderá determinar ao controlador que elabore relatório de impacto à proteção de dados pessoais, inclusive de dados sensíveis, referente a suas operações de tratamento de dados, nos termos de regulamento, observados os segredos comercial e industrial

Parágrafo único. Observado o disposto no caput deste artigo, o relatório deverá conter, no mínimo, uma descrição dos tipos de dados coletados, a metodologia utilizada para a coleta e para a garantia da segurança das informações e análise do controlador com relação às medidas, salvaguardas e mecanismos de mitigação de risco adotados.



# Artigos relevantes a proteção de dados

## Responsabilidade do controlador e dos processadores

Art. 42. O controlador ou o operador que, em razão do exercício de atividade de tratamento de dados pessoais, causar a outrem dano patrimonial, moral, individual ou coletivo, em violação à legislação de proteção de dados pessoais, é obrigado a repará-lo.

## Da segurança e do sigilo de dados

Art. 46. Os agentes de tratamento devem adotar medidas de segurança, 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.

## Notificação de vazamento de dados

Art. 48. O controlador deverá comunicar à autoridade nacional e ao titular a ocorrência de incidente de segurança que possa acarretar risco ou dano relevante aos titulares.

## Melhores práticas de segurança de dados

Art. 49. Os sistemas utilizados para o tratamento de dados pessoais devem ser estruturados de forma a atender aos requisitos de segurança, aos padrões de boas práticas e de governança e aos princípios gerais previstos nesta Lei e às demais normas regulamentares.

## Sanções administrativas

Art. 52. Os agentes de tratamento de dados, em razão das infrações cometidas às normas previstas nesta Lei, ficam sujeitos às seguintes sanções administrativas aplicáveis pela autoridade nacional:

- advertência, com indicação de prazo para adoção de medidas corretivas;
- multa simples, de até 2% (dois por cento) do faturamento da pessoa jurídica de direito privado, grupo ou conglomerado no Brasil no seu último exercício, excluídos os tributos, limitada, no total, a 50 milhões de reais por infração;
- multa diária, observado o limite total a que se refere o inciso II;
- publicização da infração após devidamente apurada e confirmada a sua ocorrência;
- bloqueio dos dados pessoais a que se refere a infração até a sua regularização;
- eliminação dos dados pessoais a que se refere a infração;

# Transformação digital, nuvem, big data e o desafio da conformidade



A Lei Geral de Proteção de Dados (LGPD) do Brasil exige as melhores práticas em segurança para dados pessoais e observa que os dados pessoais que tiverem sido anonimizados não serão considerados no escopo da Lei, se não puderem ser revertidos facilmente para o seu estado original por aqueles que possam obtê-los.

O desafio para a empresa moderna, é como equilibrar a necessidade de usar estes dados para processos, análises de negócios e ultimamente o sucesso no mercado, com a proteção aos dados confidenciais contra hackers, pessoas de dentro da organização ou vazamentos acidentais.

Por exemplo, os dados têm que fluir entre várias plataformas na nuvem e nas dependências da empresa, além de serem acessados por funcionários remotos e analisados em plataformas de big data. Durante todo o processo, os dados devem estar seguros para permanecer em conformidade com a LGPD e leis semelhantes.

Diante da transformação digital, as atuais estratégias de segurança usadas por empresas se tornam insuficientes. À medida que as empresas implementam novas tecnologias como nuvem e big data, muitas vezes escolhem soluções pontuais para proteger dados em uma plataforma ou outra. Isso leva a uma abordagem de segurança fragmentada e a falhas na proteção de dados.

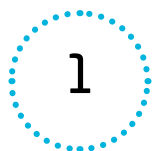
# Melhores práticas e as soluções da Thales Cloud Protection & Licensing

Entre as melhores práticas de conformidade e segurança para uma empresa com infraestrutura de TI híbrida incluímos quatro aspectos que se tornam essenciais no processo de conformidade com a LGPD

- Busca e proteção de dados sensíveis em ambientes de infraestrutura híbridos.
- Anonimização e / ou pseudonimização de dados pessoais
- Autenticação segura e centralizada
- Controle de acesso a dados sensíveis
- Monitoramento e registros de acessos aos dados

A Thales Cloud Protection & Licensing tem ampla experiência em conformidade com legislações como GDPR ou regulamentos como PCI. Além disso, a Thales é extremamente experiente em implementações abrangentes para corporações globais, em múltiplas plataformas de TI híbridas.

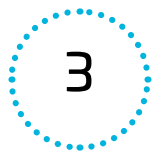
Nossas soluções oferecem aos clientes recursos para:



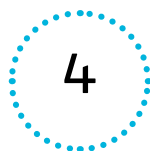
Busca e classificação de dados confidenciais



Anonimização e pseudonimização de dados



Gerenciamento das chaves usadas para criptografia de dados



Controle de acesso do usuário a dados



Registros de eventos de acesso a dados



Proteção de dados em ambientes de infraestrutura híbridos

# 1. Busca e classificação de dados confidenciais

## **Descoberta de dados, classificação e análise de risco**

Thales CipherTrust Data Discovery and Classification ajuda sua organização a obter visibilidade completa de seus dados confidenciais com eficiente descoberta de dados, classificação e análise de risco em armazenamentos de dados heterogêneos - nuvem, big data e ambientes tradicionais - em sua empresa.

Simples de implantar e usar, ele oferece um único painel de controle que permite obter uma compreensão clara de quais dados confidenciais você possui, onde estão localizados e os riscos de exposição. Com visualizações ricas e relatórios detalhados, você pode descobrir e fechar suas lacunas com mais facilidade, tomar melhores decisões sobre o compartilhamento de dados de terceiros e migração para a nuvem.

# 2. Anonimização e pseudonimização de dados

## **Criptografia transparente de dados em repouso**

A solução CipherTrust Transparent Encryption da Thales protege os dados com criptografia para dados em repouso de arquivo e nível de volume, controles de acesso e registro de auditoria de acesso a dados sem aplicativos de reengenharia, banco de dados ou infraestrutura. A instalação do software de criptografia transparente de arquivos é simples, escalável e rápida, com agentes instalados acima do sistema de arquivo em servidores ou máquinas virtuais para assegurar políticas de segurança e conformidade de dados. O gerenciamento de política e criptografia de chaves é fornecido pela CipherTrust Data Security Manager.

## **Tokenização com mascaramento dinâmico de dados**

A solução CipherTrust Vaultless Tokenization com Mascaramento Dinâmico de dados reduz drasticamente os custos e os esforços necessários para cumprir com as políticas de segurança e normas regulatórias, como a LGPD. A solução oferece capacidades para tokenização de banco de dados e segurança de tela dinâmica. As empresas podem eficientemente realizar seus objetivos de proteger e pseudoanonimizar ativos sensíveis — estejam eles em ambiente de data center, big data, em container ou na nuvem.

## **Módulos de Segurança de Hardware (HSMs)**

Os Módulos de Segurança de Hardware (HSMs) Thales Luna são certificados FIPS 140-2 nível 3, e INMETRO (ITI ICP-BRASIL), com uma ampla gama de usos para acelerar operações criptográficas, proteger o ciclo de vida de chave criptográfica e fornecer uma base de confiança para toda a sua infraestrutura de criptografia. A premiada solução Thales Data Protection On Demand é uma plataforma baseada em nuvem que fornece uma ampla gama de serviços de gerenciamento de chaves e HSM na nuvem por meio de um simples mercado on-line. Isso inclui o HSM como serviço e o Key Management como serviço.

# 3. Gerenciamento das chaves usadas para criptografia de dados

## **Gerenciamento de chaves centralizado**

A solução Enterprise Key Management unifica e centraliza o gerenciamento de criptografia de chaves no local e fornece gerenciamento seguro de chaves para soluções de armazenamento de dados. Os produtos para gerenciamento de chave na nuvem incluem o CipherTrust Cloud Key Manager.

# 4. Controle de acesso do usuário e autenticação

## **Gerenciamento de acesso baseado em nuvem como serviço**

O SafeNet Trusted Access simplifica o login dos usuários, pois é um serviço de gerenciamento de acesso que protege e gerencia centralmente o acesso a aplicativos baseados na web ou na nuvem. Empresas podem escalar os controles de acesso à nuvem enquanto atendem às necessidades dos negócios, de gerenciamento de riscos e de conformidade ao aplicar políticas flexíveis baseadas em risco, SSO na nuvem e métodos de autenticação universal.

## **Gerenciamento de identidades e acesso**

Com as soluções Thales Identity and Access Management, empresas podem proteger facilmente os aplicativos na nuvem ou no centro de dados, e atender às necessidades de gerenciamento de risco com base em suas estruturas de segurança atuais e utilizando os métodos de autenticação existentes para acesso à informação.

## Gerenciamento de acesso a dados baseado em criptografia

A solução CipherTrust Data Security Manager da Thales permite que a organização limite os privilégios de acesso de usuários a sistemas de informações que contenham informações sensíveis.

## 5. Registro de acesso ao banco de dados

### Security Intelligence Logs

A solução Security Intelligence Logs da Plataforma CipherTrust permite que sua organização identifique tentativas de acesso não autorizado e desenvolva parâmetros de modelos de acesso do usuário autorizado. A CipherTrust Security Intelligence integra-se aos principais sistemas de gerenciamento de eventos e informações de segurança (SIEM) que tornam essas informações acionáveis.

## 6. Proteção de dados em ambientes de infraestrutura híbridos

### Soluções de segurança BYOE e BYOK para nuvem

A Thales oferece várias maneiras de proteger dados em plataformas na nuvem. Organizações podem trazer sua própria criptografia para a nuvem (BYOE - Bring Your Own Encryption) usando nossa solução CipherTrust Transparent Encryption. Clientes também podem adicionar camada de segurança às suas implementações BYOK (Bring Your Own Key), usando o CipherTrust Cloud Key Manager (CCKM) para obter visibilidade e gerenciamento centralizados de ciclo de vida de várias nuvens, com armazenamento seguro de chaves FIPS-140-2.

### Plataforma de proteção de dados para infraestrutura híbrida

A CipherTrust Data Security Platform é uma plataforma extremamente flexível e escalonável, oferecendo proteção e controle de acesso a bancos de dados, arquivos e contêineres - e pode proteger dados que residem em ambientes de nuvem, virtuais, big data e físicos. As plataformas e tecnologias suportadas inclui:

- IaaS, PaaS e SaaS: Amazon Web Services, Google Cloud Platform, Microsoft Azure, Salesforce, Microsoft Office365
- OSs: Linux, Windows and Unix
- Big data: Hadoop, NoSQL, SAP HANA e Teradata
- Database: IBM DB2, Microsoft SQL Server, MongoDB, MySQL, NoSQL, Oracle, Sybase entre outros

Aonde quer que você opere e qualquer que seja o regulamento, você pode confiar na Thales para ajudá-lo a gerenciar o seu risco. A Thales pode ajudar a sua organização a cumprir com muitos dos requisitos da LGPD.

## Recursos relacionados

CipherTrust Data Security Platform Overview<sup>2</sup>

CipherTrust Transparent Encryption<sup>3</sup>

CipherTrust Tokenization com Mascaramento Dinâmico de dados<sup>4</sup>

Gerenciamento Centralizado de Chaves<sup>5</sup>

CipherTrust Cloud Key Manager<sup>6</sup>

Information and Access Management Solutions<sup>7</sup>

Thales Trusted Access: Cloud-based Access Management<sup>8</sup>

Thales Data Protection on Demand<sup>9</sup>

Thales Luna Network HSMs<sup>10</sup>

CipherTrust Security Intelligence com Integração SIEM<sup>11</sup>

<sup>2</sup> <https://cpl.thalesgroup.com/es/encryption/data-security-platform>

<sup>3</sup> <https://cpl.thalesgroup.com/es/encryption/transparent-encryption>

<sup>4</sup> <https://cpl.thalesgroup.com/es/encryption/tokenization>

<sup>5</sup> <https://cpl.thalesgroup.com/es/encryption/key-management>

<sup>6</sup> <https://cpl.thalesgroup.com/encryption/key-management/ciphertrust-cloud-key-manager>

<sup>7</sup> <https://cpl.thalesgroup.com/es/access-management>

<sup>8</sup> <https://cpl.thalesgroup.com/es/access-management/safenet-trusted-access>

<sup>9</sup> <https://cpl.thalesgroup.com/encryption/data-protection-on-demand>

<sup>10</sup> <https://cpl.thalesgroup.com/encryption/hardware-security-modules/network-hsms>

<sup>11</sup> <https://cpl.thalesgroup.com/encryption/security-intelligence-logs>

## About Thales

As pessoas em quem você confia para proteger sua privacidade confiam na Thales para proteger sua nuvem, seus dados e software. Em termos de segurança de dados, as organizações enfrentam cada vez mais momentos decisivos. Seja na criação de uma estratégia de criptografia, licenciamento de software, fornecimento de acesso seguro à nuvem ou cumprimento de normas de conformidade, você pode confiar na Thales para proteger sua transformação digital.

Tecnologia decisiva para momentos decisivos.

# THALES

### Thales – Latin America / Brazil

Rua Surubim, 577, 12º. andar, cjs.  
123/124 Brooklin  
São Paulo - SP - BRASIL 04571-050

Tel: +55 11 3198-5250 | Email: [CPL\\_Sales\\_AMS\\_TG@thalesgroup.com](mailto:CPL_Sales_AMS_TG@thalesgroup.com)

> [cpl.thalesgroup.com](http://cpl.thalesgroup.com) <



# SafeNet Virtual KeySecure



## Overview

SafeNet Virtual KeySecure™ is a hardened virtual security appliance that provides organizations with a more operation and expense friendly friendly alternative to using a hardware appliance for secure key management and meeting security and compliance requirements. By using a virtual key manager instead of a hardware appliance, organizations can scale key management In private or public cloud infrastructures, and eliminate the cost for additional rack space. SafeNet Virtual KeySecure allows organizations to utilize a secure virtual appliance to manage keys as well as data encryption, 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.

## SafeNet Virtual KeySecure, the new generation

Thales's latest SafeNet Virtual KeySecure models are the k170v and k470v, which are built on prevailing cloud-based technologies and provide a cloud friendly, key management solution for your organization.

With a REST interface and microservices based architecture, new generation models have been built to easily deploy and scale within your environment. Using the new REST API Playground, developers can explore the new set of key management and administrative endpoints.

Organizations can lower costs and scale key management with a virtual appliance that is quickly deployed for high-availability in various private and public cloud infrastructures.

## Combat risk and get affordable security

As companies migrate to virtual data centers and cloud environments, sensitive data is likely to reside outside the physical walls of the enterprise, making secure data management a challenge. With regulations such as PCI DSS, HIPAA, HITECH, GDPR, GLBA, and SOX, combined with the high cost of data breaches, the pressure to protect this sensitive data has never been greater. SafeNet Virtual KeySecure provides customers with is a virtual appliance that manages and securely stores encryption keys for diverse encryption solutions,including the SafeNet Data Protection portfolio, traditional storage and hyperconvergence solutions, Big Data, KMIP compliant solutions, and many others. Thales offers a virtual key management appliance that can also be integrated with a hardware root of trust.

# SafeNet Virtual KeySecure benefits

## Host Anywhere:

- The k150v and k450v models can be rented from a service such as AWS marketplace, or hosted on a virtual machine such as VMware.
- New! The new k170v and k470v models provide additional hosting options, and can run as a native virtual machine on VMware, AWS, Microsoft Azure, Oracle VM Virtual Box, OpenStack, Microsoft Hyper-V and Google with more public/private clouds coming soon.

## Compatibility:

- Compatibility with the OASIS Key Management Interoperability Protocol (KMIP) standard provides support for a large and large and growing partner ecosystem.
- Support of the SafeNet Data Protection portfolio provides customers with a broad spectrum of use cases that can be supported.
- Bring Your Own License (BYOL) for AWS Marketplace enables purchase directly from Thales (available for all models).

## Flexibility:

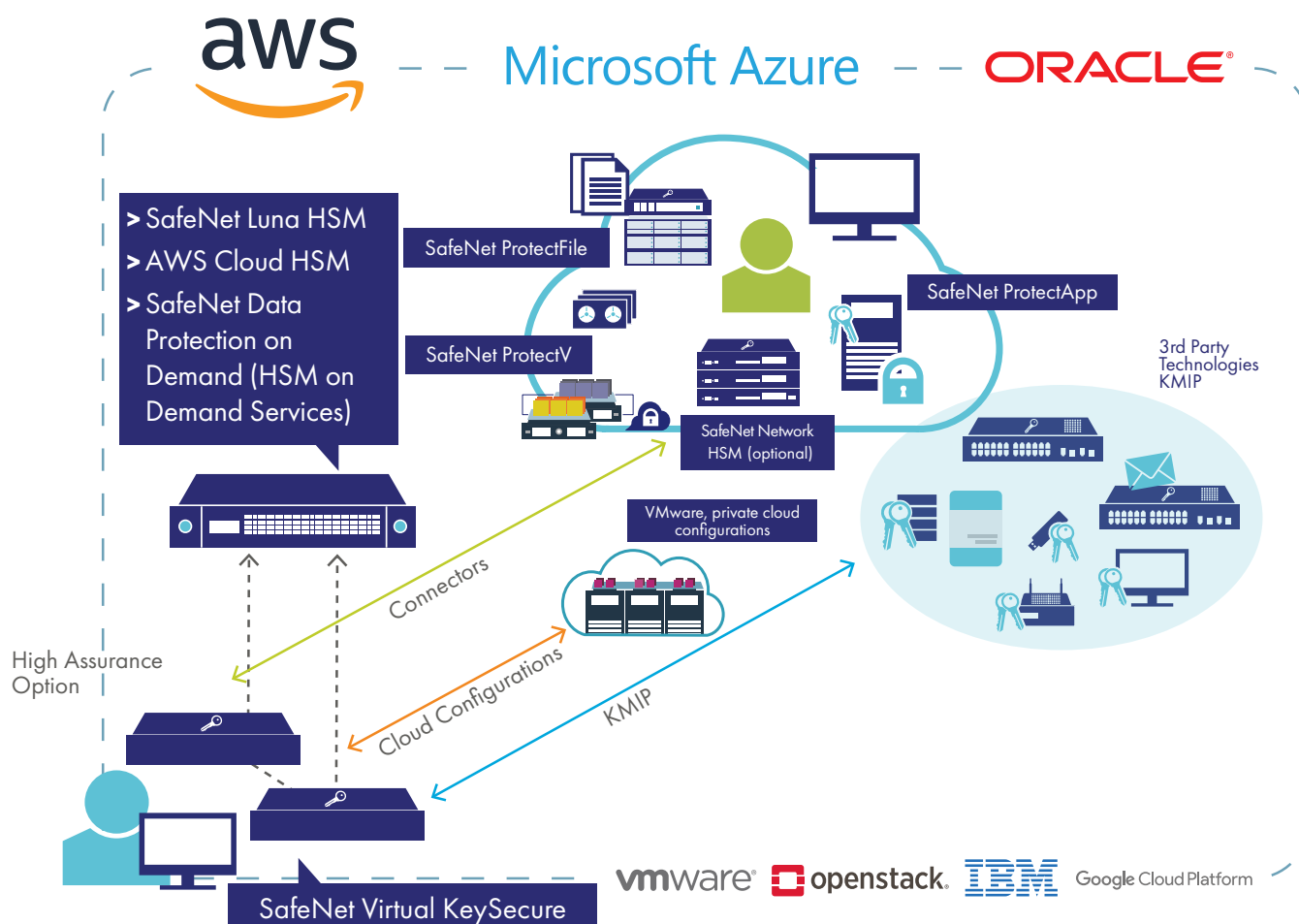
- Subscription-based offerings that are better suited for operating expenditure (op-ex) models, versus capital expenditure models (standard hardware purchases) that require upfront payment.
- Flexible deployment options can easily scale to provide key management at remote facilities or in cloud infrastructures.
- A newly designed GUI for the k170v and k470v models provides a clean, simple, user-friendly experience for commonly used functions like key management, user groups and permissions, and audit log review.

## Centralized Key Management:

- Key security policies can be consolidated across multiple, disparate encryption systems, protecting current investments.
- Centralized, efficient auditing of key management offers simplified compliance for cloud environments and decreases the amount of time spent on compliance mandates.

## SafeNet Virtual KeySecure use cases

SafeNet Virtual KeySecure provides application data protection and centralized key management for Thales and third-party encryption products across stored and archived data, virtual workloads, and application data protection.





## Virtual KeySecure model comparison

| SafeNet Virtual KeySecure                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                               |                                                                                                    |                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
|                                                                                                                                                                    | NEW k170v                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | NEW k470v                                                                                     | k450v                                                                                              | k150v                                                                                              |
| <b>Max concurrent Clients per Cluster</b> ***                                                                                                                      | 100 clients per cluster and 1,000 connections                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 1000 clients per cluster and 7000 connections                                                 | 1,000 clients per cluster and 7,000 connections                                                    | 100 clients per cluster and 1,000 connections                                                      |
| <b>FIPS 140-2 Support</b>                                                                                                                                          | L3 with HSM integration                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | L3 with HSM integration                                                                       | L1, L3 with HSM integration                                                                        | L1, L3 with HSM integration                                                                        |
| <b>HSM Integration*</b>                                                                                                                                            | SafeNet Luna HSM 5, 6, 7, AWS Cloud HSM & SafeNet DPoD (HSM On Demand Services)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | SafeNet Luna HSM 5, 6, 7, AWS Cloud HSM & SafeNet DPoD (HSM On Demand Services)               | SafeNet Luna HSM 5, 6                                                                              | SafeNet Luna HSM 5, 6                                                                              |
| <b>Supports SafeNet Data Protection Portfolio**</b>                                                                                                                | SafeNet ProtectApp, SafeNet ProtectV, SafeNet ProtectDB, SafeNet Tokenization, SafeNet ProtectFile, SafeNet High Speed Encryptors, SafeNet Sure Drop, Oracle TDE and SQL Server EKM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                               |                                                                                                    |                                                                                                    |
| <b>Supports KMIP</b>                                                                                                                                               | Yes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Yes                                                                                           | Yes                                                                                                | Yes                                                                                                |
| <b>CPU (Minimum Requirements)</b>                                                                                                                                  | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 4                                                                                             | 12                                                                                                 | 1                                                                                                  |
| <b>RAM (Minimum Requirements)</b>                                                                                                                                  | 4 GB                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 8 GB                                                                                          | 8192 MB                                                                                            | 1024 MB                                                                                            |
| <b>Network Adapter</b>                                                                                                                                             | 1 Allowed / Available                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Configurable (2 or More)                                                                      | 2 (Configurable)                                                                                   | 1 Allowed / Available                                                                              |
| <b>Interface / Architecture</b>                                                                                                                                    | REST API<br>NAE-XML<br>KMIP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | REST API<br>NAE-XML<br>KMIP                                                                   | NAE-XML<br>KMIP                                                                                    | NAE-XML<br>KMIP                                                                                    |
| <b>Supported Cloud Environments</b>                                                                                                                                | AWS, MS Azure, VMware vSphere, Oracle VM Virtual Box, OpenStack, Microsoft Hyper-V and Google                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | AWS, MS Azure, VMware vSphere, Oracle VM Virtual Box, OpenStack, Microsoft Hyper-V and Google | Can be hosted on a virtual machine running VMware or rented from a service such as AWS Marketplace | Can be hosted on a virtual machine running VMware or rented from a service such as AWS Marketplace |
| <b>Thales Third-party Integration Support</b><br><i>(Support for k170v and k470 underway, please contact your Thales account representative for more details.)</i> | <b>Content Management:</b> Open Text (EMC InfoArchive), Stealth Content Store, ServiceNow, VMcrypt<br><b>Mainframe Encryption:</b> PKWARE<br><b>Big Data:</b> Dataguise, DataStax, Hadoop, MongoDB, MariaDB, SAP HANA, Cassandra, Couchbase, Hortonworks, Cloudera<br><b>Analytics:</b> IBM QRadar, HPE ArcSight, Splunk, RSA Security Analytics, Above Security<br><b>Application Servers:</b> IBM WebSphere, Oracle Weblogic, Microsoft IIS, Apache Tomcat, Red Hat JBoss<br><b>Backup Solutions:</b> Commvault Simpana, Symantec NetBackup (via NetApp)<br><b>Cloud Storage:</b> Amazon Web Services S3, DropBox, Google Cloud Storage, Google Drive, IBM ICDES, Panzura Storage Controller<br><b>Hyperconvergence:</b> Nutanix, NetApp Cloud ONTAP, NetApp AltaVault, Cisco UCS, Cisco HyperFlex, vSAN<br><b>Cloud Access Security Brokers:</b> CipherCloud, SkyHigh Networks, Perspecsys (Blue Coat), Hitachi Sepaton VTL, CSC ServiceMesh, Netskope Active Encryption, Vaultive Cloud Data Protection Platform, Elastica (Symantec CASB), Alfresco Open ECM<br><b>Databases:</b> MS SQL Server (EKM), Oracle (TDE), IBM DB2, Oracle MySQL, Oracle Database, Teradata<br><b>File and Disk Encryption:</b> PKWARE, IBM, Dell, AWS, Microsoft, LUKS, ViaSat<br><b>Identity Management:</b> Centrify Privilege Service, Lieberman Software<br><b>Key Managers:</b> Hadoop KMS, Cloudera Navigator Key Trustee Server, HashiCorp Vault<br><b>Physical Storage:</b> NetApp NSE, Dell Compellent (SC and XC), HPE MSL/ESL Tape Libraries, HPE 3Par StoreServ, HPE XP7, Hitachi, SP, Hitachi HUS, Hitachi RAID700, IBM XIV SED, Quantum Scalar Series[i6000, i500 & i40/80], Viasat, Brocade FS8-18, Huawei Oceanstor, Tintri VMStore, NexentaStor 4.5 |                                                                                               |                                                                                                    |                                                                                                    |

\*Supports master key storage as a root of trust in HSMs: Cloud-based, such as Amazon Web Services CloudHSM, SafeNet Data Protection on Demand (HSM on Demand Services), or SafeNet Luna HSM, a hardware appliance option that is deployed on-premises in a range of models and configurations.

\*\*SafeNet Virtual KeySecure k150v and k450v appliances using SafeNet ProtectApp, SafeNet ProtectDB, and SafeNet Tokenization, require the purchase of SafeNet Crypto Pack. SafeNet ProtectV and SafeNet ProtectFile do not require SafeNet Crypto Pack feature activation. The k170v and k470v using any of the SafeNet Data Protection portfolio do not require the purchase of SafeNet Crypto Pack.

\*\*\*Based on physical/virtual default system environments. Based on an average client test, performance will vary depending on use case and configuration.

## Product highlights:

- Full Lifecycle Key Support and Automated Operations. Simplifies management of encryption keys across the entire lifecycle, including secure key generation, storage and backup, and key distribution, deactivation and deletion. SafeNet Virtual KeySecure makes automated, policy-driven operations easy for tasks such as key expiry and key rotation.
- Centralized Administration of Granular Access, Authorization Controls, and Separation of Duties. Management console unifies key management operations across multiple encryption deployments and products while ensuring that administrator roles are properly defined.
- High-Availability and Intelligent Key Sharing. Deploys in flexible, high-availability configurations across geographically dispersed centers or service provider environments.
- Auditing and Logging. Centralized management includes detailed logging and audit tracking of all key state changes, administrator access and policy changes. Audit trails are securely stored and signed for non-repudiation.

## About Thales Cloud Protection & Licensing

Today's enterprises depend on the cloud, data and software in order to make decisive decisions. That's why the most respected brands and largest organizations in the world rely on Thales to help them protect and secure access to their most sensitive information and software wherever it is created, shared or stored – from the cloud and data centers to devices and across networks. Our solutions enable organizations to move to the cloud securely, achieve compliance with confidence, and create more value from their software in devices and services used by millions of consumers every day.

## Try it before you buy it.

[Click here](#) to enroll in Amazon Web Services 30-day free trial.

## Need an on-premise appliance?

Learn more about [on-premises enterprise key management](#).

White Paper

# Thales Key Management

Simplify data security by  
centrally managing keys  
and policies across  
the enterprise

[cpl.thalesgroup.com](http://cpl.thalesgroup.com)

**THALES**  
Building a future we can all trust

# Contents

|          |                                                           |
|----------|-----------------------------------------------------------|
| <b>3</b> | <b>Executive Summary</b>                                  |
| <b>3</b> | <b>Data Security Challenges</b>                           |
| 4        | Looking Back at Encryption and Key Management Techniques  |
| 4        | Data Encryption Across Disparate Systems → Security Silos |
| 5        | The Business Problem – Siloed Data                        |
| <b>5</b> | <b>Essentials of Enterprise Key Management</b>            |
| 6        | Facilitate Governance and Reporting                       |
| <b>7</b> | <b>Introducing CipherTrust Manager</b>                    |
| 7        | The Foundation of Thales Key Management Solutions         |
| 7        | Key Benefits                                              |
| <b>8</b> | <b>Thales Enterprise Key Management Solutions</b>         |
| <b>9</b> | <b>Thales Hardware Security Module Solutions</b>          |
| <b>9</b> | <b>Summary</b>                                            |
| <b>9</b> | <b>About Thales</b>                                       |

# Executive Summary



Today, every IT organization is striving to protect valuable digital assets from accidental exposure or intentional misuse. To meet various digital privacy regulations and compliance mandates, many organizations have deployed a variety of point encryption solutions as a primary method of protecting sensitive data. Unfortunately, the majority of disparate encryption solutions have fallen short in addressing enterprise key management challenges and the result is a weakened security posture, increases in failed audits and more security breaches.

This white paper looks at the evolution of encryption and key management systems, and examines the major challenges faced by IT teams for encryption systems, including regulation and compliance, complexity, and management tools. The review of major challenges is followed by a review of recent industry initiatives and compliance regulations that are shaping the future of key management.

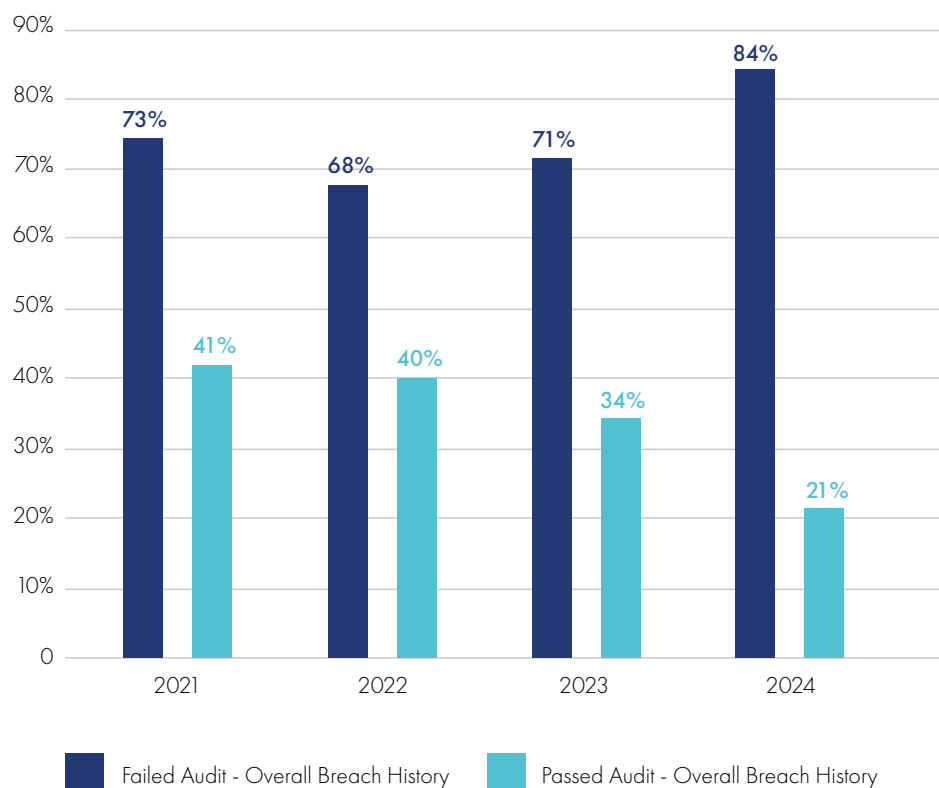
The paper concludes with an introduction to CipherTrust Manager, the next-gen enterprise key management offering from Thales. CipherTrust Manager provides a powerful integrated solution that enables organizations to centrally manage encryption keys and policies for CipherTrust Connectors and third-party KMIP-compliant products.

## Data Security Challenges

To drive operational efficiency into their data-intensive processes, many organizations are adopting digital transformation and migrating their applications and data to the cloud. Data migration to third-party hosted environments and multiple cloud service providers is creating new attack surfaces that cybercriminals can exploit, as a result, data breaches threaten the IT landscape at an increasing rate. All of this adds up to today's data environments becoming even more complex – which is a top barrier to data security.

In the 2024 Thales Data Threat Report (which is based on an S&P Global survey of nearly 3,000 respondents across 18 countries who have responsibility or influence over IT and data security), operational complexity remained a security concern and human error was reported to be the leading cause of data breaches (31%). It wasn't surprising to see a strong correlation between achieving compliance and reducing breaches.





Source S&P Global Market Intelligence's 2021-2024 Data Threat custom surveys

**Figure 1: Correlation between passing audits and experiencing fewer security breaches**

Ransomware attacks are more common, with 28% of respondents experiencing an attack in 2023, up from 22% in 2022. Multi-cloud continues to be a reality regardless of cloud maturity but in 2024 enterprises can better abstract their controls so that they are more applicable to multiple jurisdictions and changing market requirements.

## 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 for authorized users (employees, partners and customers) has taken priority, with data security being implemented as an afterthought. Connecting public and private resources to any company's sensitive data increases the risk of data breaches.

In response to the increasing threat volume, complexity and severity, a variety of perimeter, endpoint security controls and security policies have been adopted and data privacy regulations and compliance requirements have been developed. Potentially exploitable gaps between the controls and policies can be mitigated with simplified encryption and key management. This white paper explains the importance of resolving the difficulties of disparate native key repositories scattered across 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 managing a variety of cryptographic keys has made life much more challenging for the IT and security teams. Nearly all offline data storage devices and many database management systems (DBMS) include the option of embedded native encryption capability. The islands of encryption from disparate providers results in keys and key management software that frequently don't interoperate well with one another.

System administrators and database administrators (DBAs) end up becoming the managers of encryption keys specific to data storage, database management systems and applications. The security silos distract from the primary tasks of IT and database administration, putting an enterprise's overall security posture at risk.

## The Business Problem - Siloed Data

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 data storage, database management systems and applications accumulated over time from separate vendors. A heterogeneous environment means that an enterprise using native TDE to secure databases, such as Oracle and Microsoft SQL Server, has to factor in the increased risk and administrative resources required to manage multiple incompatible encryption solutions. Managing a separate encryption system for each data store is costly, complex and error-prone due to inconsistent security policies and processes, as shown in Figure 2 below.

When each system administrator controls encryption keys separately for each data repository they manage, the keys are generally stored in the same location as the encrypted data, creating an opportunity for security compromise — the electronic equivalent of taping the key onto the front door.

Using manual systems to store and transmit encryption keys, weak password control, or failing to revoke keys when an employee leaves the organization, makes strict adherence to compliance requirements nearly impossible and data breaches are likely to occur.

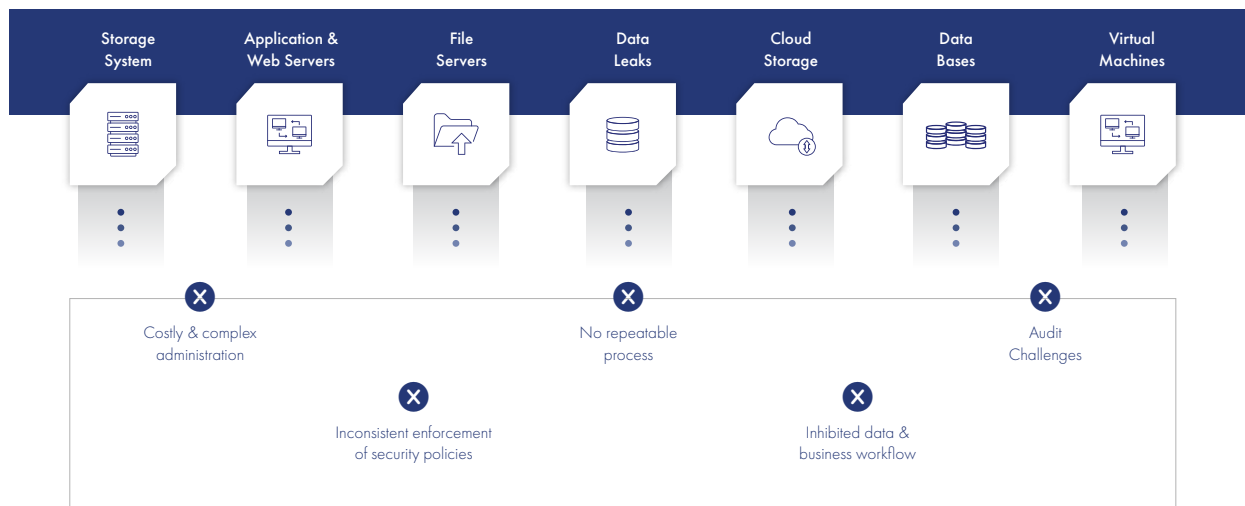


Figure 2: Costly and complex to manage data silos

## Essentials of Enterprise Key Management

As organizations deploy an ever-increasing number of encryption solutions, they can decrease the risk of a breach or non-compliance by using 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 data security challenges.

### 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 HSMs or a cloud-based HSM-as-a-service, based on the level of assurance your company needs — whether it is FIPS 140 L1 or L3.

### Centralized Key Lifecycle Management

The key management system (KMS) should be able to centrally manage keys across their entire lifecycle, including secure key generation, backup/restore, clustering, deactivation and deletion. The KMS 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 onsite datacenter to external hosted environments to multiple cloud service providers, the enterprise key management solution should be flexible enough to adapt to changing requirements for sensitive data.

A flexible key management solution supports on-premises infrastructure and is deployable as a virtual appliance in public cloud environments such as AWS, Azure, Google Cloud, Oracle, Salesforce and SAP, private clouds such as VMware vSphere, Microsoft Hyper-V and Nutanix AHV, and hybrid clouds such as Azure Stack.

## Interoperability with Third-party Systems

Recommended enterprise key management solutions support the following three major interoperability standards that enable you to work with multiple server, storage and device vendors who use the keys for authentication, digital signing or encrypting data.

- **PKCS#11** – Public Key Cryptographic Standard #11 (PKCS#11) specifies an API for devices to interoperate with hardware security modules (HSMs) and smart cards that hold cryptographic tokens. PKCS#11 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, Oracle TDE uses PKCS#11.
- **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 use the keys for embedded encryption. KMIP enables 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 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:

- **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. PCI DSS tests span a wide variety of common security practices along with technologies such as encryption, key management, and other data protection techniques. The latest update to the standard, PCI DSS 4.0, in force since April 2024, includes a shift to a more risk-based and customized approach.

- **General Data Protection Regulations (GDPR)**

Perhaps the most comprehensive data privacy standard to date, GDPR affects every 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 impacts 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.

- **Gramm Leach Bliley Act (GLBA)**

The United States requires that firms acknowledge publicly when a disclosure event occurs. 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 PII and PHI data, with centralized key management as its foundation.

- **Network and Information Security Directive 2 (NIS2)**

The European Union's NIS2 is a legislative act that aims to achieve a high common level of cybersecurity for organizations across the European Union. NIS2 requires operators of critical infrastructure and essential services in the EU to implement appropriate security measures that include encryption, policies and procedures regarding the use of cryptography and access management.



- **Digital Operational Resilience Act (DORA)**

The DORA Act harmonizes the operational resilience rules applying to 20 different types of financial entities. The new regulation requires financial entities, and their IT service providers, such as cloud platforms, to implement measures to improve the digital operational resilience of the sector, including the protection of data at rest, in use, and in motion, and the protection and management of cryptographic keys.

# Introducing CipherTrust Manager

## The Foundation of Thales Key Management Solutions

CipherTrust Manager enables organizations to meet their data privacy and compliance requirements by centrally managing encryption key lifecycles and policies, independent of where the data resides. CipherTrust Manager simplifies key lifecycle management tasks, including secure key generation, backup/restore, clustering, deactivation, and deletion. CipherTrust Manager 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.

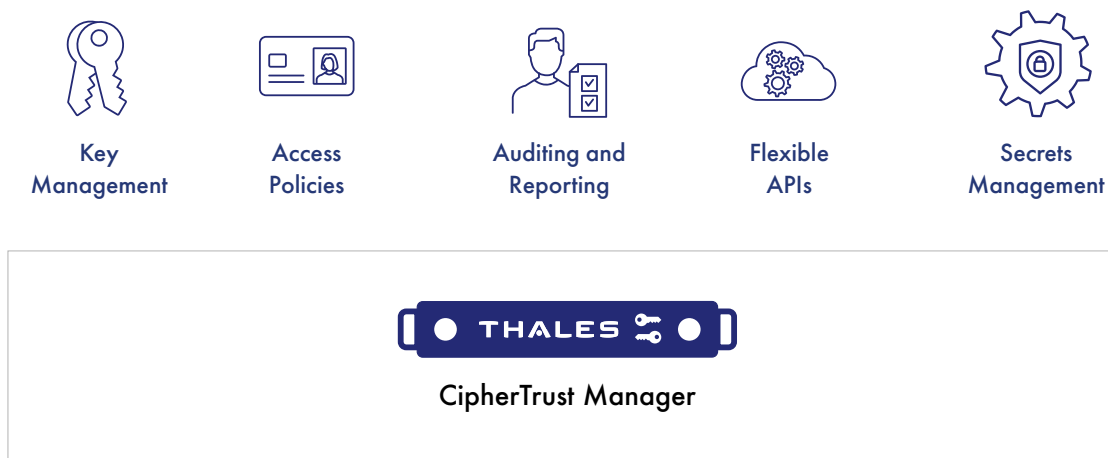


Figure 3: CipherTrust Manager

## 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 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 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 Management 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. Flexible deployment options can easily scale to provide key management at remote facilities or in cloud infrastructures.
- CipherTrust Manager supports managing Key Encryption Keys (KEKs) in the FIPS 140 L3 boundary of Luna Network HSM.

# Thales Enterprise Key Management Solutions

Thales Enterprise Key Management solutions are built on CipherTrust Manager. CipherTrust Manager serves as a robust, standards-based platform for managing encryption keys from disparate sources across the enterprise, simplifying the administrative challenges around encryption key management, ensuring that keys are secure and always provisioned to authorized encryption services.

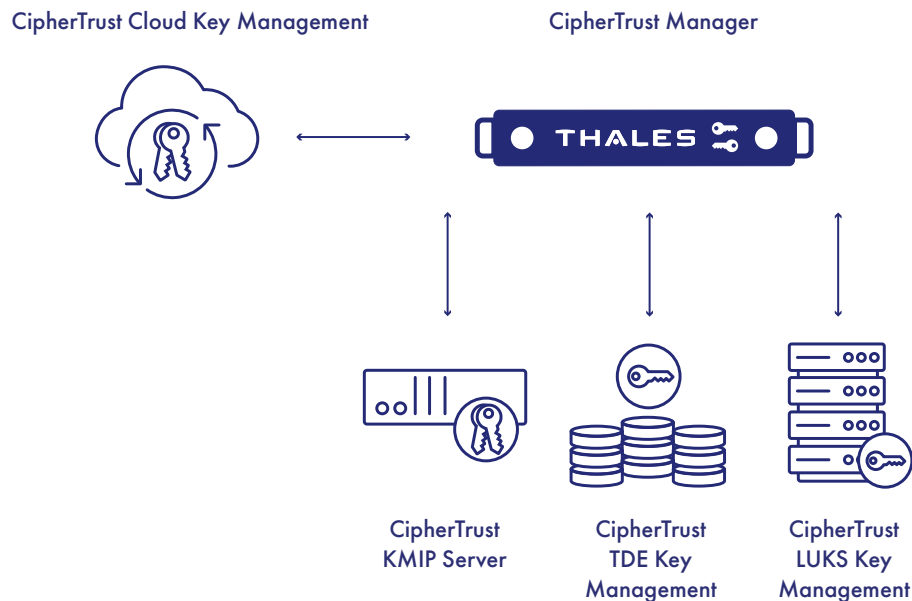


Figure 4: CipherTrust Cloud Key Management and CipherTrust Enterprise Key Management Solutions

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:** Manages encryption keys across multiple KMIP clients (legacy data storage, cloud and virtual environments, SAN and NAS storage arrays, self-encrypting drives, and hyper-converged infrastructure solutions). Provides a simple interface to the key management functions on CipherTrust Manager, delivering powerful, centralized key management capabilities for KMIP-compliant applications/systems.
- **TDE Key Management:** Simplifies transparent data encryption (TDE) across Oracle TDE and Microsoft SQL Server TDE databases using TDE Key Agents, and keeps the TDE keys separate from your database servers.
- **LUKS Key Management:** Linux Unified Key Setup (LUKS) Key Management enables you to centrally manage encryption keys for Linux disk partitions. Provides transparent disk encryption.

## **CipherTrust Manager additionally integrates with a collection of powerful Connectors:**

- **CipherTrust Cloud Key Management (CCKM)** manages keys for multi-cloud environments and services including AWS, Azure, GCP, Oracle, Salesforce and SAP
- **CipherTrust Secrets Management (CSM)** is an enterprise-grade secrets management solution which protects and automates access to secrets
- **CipherTrust Transparent Encryption - Ransomware Protection (CTE-RWP)** provides active behavior monitoring and data analytics, looking for abnormal I/O activity on a per-process basis
- **CipherTrust Data Protection Gateway (DPG)** offers transparent data protection to any RESTful web service or microservice leveraging REST APIs

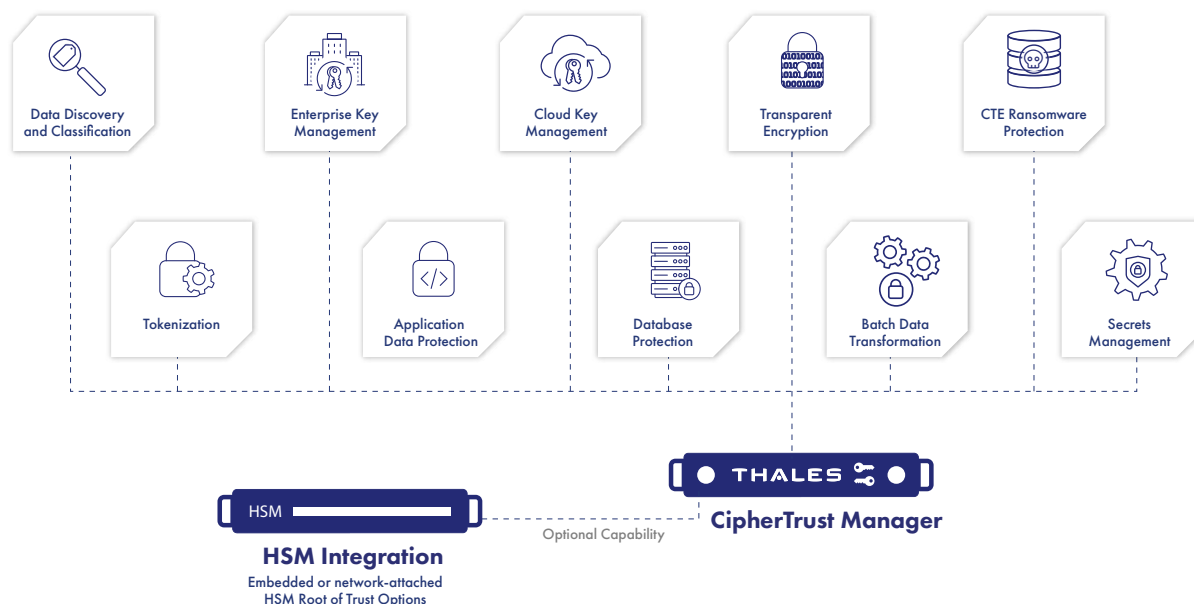


Figure 5: CipherTrust Manager and Thales Connectors comprise the Thales CipherTrust Data Security Platform.

## Thales Hardware Security Module Solutions

Thales offers a variety of hardware security modules (HSMs), which are the highest performing, most secure and easiest HSMs to integrate in the market today. HSMs act as trust anchors to protect the master keys that encrypt your data and digital identities in a high assurance FIPS 140 Level 3-validated, 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. Luna HSMs can easily be integrated with a wide range of applications to accelerate general cryptographic operations. The embedded PCI module is a Luna HSM PCI card available as an option in the CipherTrust Manager model k570.
- Cloud HSMs are available in the Data Protection On Demand (DPoD) online marketplace which provides a wide range of Cloud HSM and key management services.

## Summary

Data is only as secure as the system that manages the encryption keys protecting the data. A centralized enterprise key management solution is critical to ensuring all sensitive enterprise data is secure and available. CipherTrust Manager helps organizations maximize IT efficiency through a centralized, extensible platform-based solution, and supports the burdens of encryption key management across the enterprise and into the cloud – without disrupting existing application or database environments.

## 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.



#### Contact us

For all office locations and contact information,  
please visit [cpl.thalesgroup.com/contact-us](https://cpl.thalesgroup.com/contact-us)

[cpl.thalesgroup.com](https://cpl.thalesgroup.com)



# Top 10 reasons for Migrating to CipherTrust Manager for KeySecure Customers

You have relied on the KeySecure platform over many years to provide centralized key management and data protection throughout your enterprise and cloud environments. Thales has now combined the strengths of two industry leading data protection companies – Gemalto and Vormetric, to create CipherTrust™ Managers. It enables organizations to centrally manage encryption keys for the Thales data protection portfolio as well as our partner ecosystem via KMIP and NAE XML interfaces, and third party products such as Oracle TDE and SQL Server EKM. It simplifies key lifecycle management tasks, including secure key generation, backup/restore, clustering, deactivation, and deletion. The new unified management console

in CipherTrust Manager makes it easy to discover and classify data, and to protect sensitive data wherever it resides using a comprehensive set of data protection connectors from Thales. This brief captures the major reasons for current KeySecure customers to migrate to the new CipherTrust™ Manager platform now.



## Simplified Management

### 1. New Data Discovery and Classification

CipherTrust Manager enables organizations to discover and classify sensitive data wherever they reside across their on-premises, big data and cloud environments, and take remediation actions using a variety of data protection connectors to reduce business risks.

### 2. New Self-service Licensing

A new customer facing licensing portal streamlines provisioning of connector licenses and the new management console in CipherTrust Manager enables organizations to gain better visibility and control of their data protection connector licenses in use.

### 3. Improved Monitoring and Alerting

It includes tracking of all administrator access, encryption key state and policy changes in multiple log formats (RFC-5424, CEF, LEEF) for easy integration with SIEM tools. In addition, customers can generate pre-configured and customizable email alerts (SNMP v1, v2c, v3).

### 4. Developer Friendly REST APIs

CipherTrust Manager offers new REST interfaces, in addition to KMIP and NAE-XML APIs, for developers to simplify deployment of applications integrated with key management capabilities and automate development and testing of administrative functions.

## Physical Appliance Improvements

### 5. High-speed Interfaces and NIC Bonding

The new CipherTrust Manager appliances (k470 and k570) provide optional 2x1GB/2x10GB network interface cards (NIC) as well as NIC bonding to increase available bandwidth.

### 6. Password and PED Authentication

The k570 appliance now offers customers the choice of Password or PIN Entry Device (PED) authentication, allowing customers options for stronger authentication.

## Flexible Deployment Options

### 7. Cloud Friendly Deployments

CipherTrust Manager offers several options to securely migrate applications to multiple cloud environments.

- Offers support for AWS, Azure, Google Cloud, VMware, HyperV, Oracle VM and more.
- Integration with CipherTrust Cloud Key Manager to support bring your own key (BYOK) across multiple cloud infrastructures and SaaS applications

### 8. Hybrid High-Availability Clustering

It offers a choice of clustering a physical with a virtual appliance for high-availability environments to ensure optimum processing regardless of the workload location (data center or cloud).

### 9. Multi-tenancy Support

It provides capabilities required to create multiple domains with separation of duties to support large organizations with distributed locations or multiple companies hosted by Managed Service Providers (MSP)

## Expanded HSM Options

### 10. FIPS 140-2 Validated HSMs

CipherTrust Manager provides several options to integrate with a FIPS 140-2 validated physical or virtual HSMs as a secure root of trust for better key entropy.

- Built-in HSM crypto accelerator card on a CM k570 appliance.
- Network attached Luna HSM with HA clustering
- Cloud HSM (Data Protection on Demand service) for several major cloud service providers.



# Next Generation KeySecure

## DEPLOYMENT GUIDE



## Document Information

|                 |              |
|-----------------|--------------|
| Product Version | 1.10.0       |
| Release Date    | 02 June 2020 |

## Trademarks, Copyrights, and Third-Party Software

2020 Thales. All rights reserved. Thales and the Thales logo are trademarks and service marks of Thales and/or its subsidiaries and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the property of their respective owners.

## Disclaimer

All information herein is either public information or is the property of and owned solely by Thales and/or its subsidiaries 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's information.

This document can be used for informational, non-commercial, internal, and personal use only provided that:

- > The copyright notice, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- > This document shall not be posted on any publicly accessible 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.

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 Group.

# CONTENTS

|                                                                   |    |
|-------------------------------------------------------------------|----|
| Overview .....                                                    | 10 |
| Product references made in this document .....                    | 10 |
| Deployment environments .....                                     | 10 |
| Audience .....                                                    | 11 |
| Document Conventions .....                                        | 11 |
| Notes .....                                                       | 11 |
| Cautions .....                                                    | 11 |
| Warnings .....                                                    | 12 |
| Command Syntax and Typeface Conventions .....                     | 12 |
| Support Contacts .....                                            | 13 |
| Customer Support Portal .....                                     | 13 |
| Telephone Support .....                                           | 13 |
| Email Support .....                                               | 13 |
| Troubleshooting Issues in Conjunction with Customer Support ..... | 13 |
| Hardening Guidelines .....                                        | 15 |
| Network Access - Security Group Rules .....                       | 15 |
| Recommended Security Group Rules .....                            | 15 |
| Encryption .....                                                  | 16 |
| System Administrative Key .....                                   | 16 |
| HSM Configurations .....                                          | 16 |
| TLS Compatibility .....                                           | 16 |
| WARNING - Always use SSL/TLS with NAE .....                       | 17 |
| Installing the CLI toolkit .....                                  | 18 |
| Cloud-init Configuration Options .....                            | 24 |
| Using Cloud-init .....                                            | 24 |

|                                                             |           |
|-------------------------------------------------------------|-----------|
| Passing the cloud-init file in AWS .....                    | 24        |
| Passing the cloud-init file in Azure .....                  | 24        |
| Passing the cloud-init file for a Private Cloud Image ..... | 25        |
| Configuring Initial Password for the Initial User .....     | 33        |
| Configuring Default CA and Server Certificates .....        | 34        |
| Configuring initial default certificates .....              | 34        |
| Configuring IPv6 .....                                      | 36        |
| Configuring the NextGen KeySecure with a Static IP .....    | 36        |
| Configuring Disk Encryption .....                           | 37        |
| Encrypting an instance at first launch .....                | 37        |
| Encrypting an already launched instance .....               | 37        |
| Configuring HSM Integration .....                           | 39        |
| <b>System Configuration Utility (kscfg) .....</b>           | <b>43</b> |
| Logging in as ksadmin user .....                            | 43        |
| Using the kscfg utility .....                               | 44        |
| Commands .....                                              | 44        |
| Network Configuration .....                                 | 45        |
| System Reset .....                                          | 48        |
| Usage .....                                                 | 48        |
| Examples .....                                              | 49        |
| <b>Network Interface Configuration and Bonding .....</b>    | <b>50</b> |
| Typical Network Configuration Tasks .....                   | 50        |
| Configuring Static Routes .....                             | 52        |
| Network Interface Bonding .....                             | 53        |
| Configuring NIC Bonding with Static MAC Address .....       | 55        |
| <b>Amazon Web Services Deployment .....</b>                 | <b>56</b> |
| Minimum Requirements .....                                  | 56        |
| Provisioning the AMI .....                                  | 56        |
| Deploying in AWS .....                                      | 59        |
| Prerequisites .....                                         | 59        |

|                                                       |           |
|-------------------------------------------------------|-----------|
| Tasks .....                                           | 59        |
| Steps .....                                           | 59        |
| Changing the Initial User Password .....              | 66        |
| Using the Hard-coded initial password (default) ..... | 66        |
| Retrieving an Auto-generated initial password .....   | 66        |
| Licensing .....                                       | 67        |
| Lock Codes .....                                      | 67        |
| Connector/Client Licensing .....                      | 69        |
| Enhanced Networking .....                             | 69        |
| <b>Microsoft Azure Deployment .....</b>               | <b>70</b> |
| Minimum Requirements .....                            | 70        |
| Deploying in Azure .....                              | 70        |
| Prerequisites .....                                   | 71        |
| Tasks .....                                           | 71        |
| Steps .....                                           | 71        |
| Changing the Initial User Password .....              | 80        |
| Using the Hard-coded initial password (default) ..... | 80        |
| Retrieving an Auto-generated initial password .....   | 80        |
| Licensing .....                                       | 81        |
| Lock Codes .....                                      | 82        |
| Connector/Client Licensing .....                      | 83        |
| <b>Google Cloud Deployment .....</b>                  | <b>84</b> |
| Minimum Requirements .....                            | 84        |
| Deploying in Google Cloud .....                       | 84        |
| Prerequisites .....                                   | 85        |
| Tasks .....                                           | 85        |
| Steps .....                                           | 85        |
| Changing the Initial User Password .....              | 90        |
| Using the Hard-coded initial password (default) ..... | 90        |
| Retrieving an Auto-generated initial password .....   | 90        |
| Licensing .....                                       | 91        |

|                                                          |            |
|----------------------------------------------------------|------------|
| Lock Codes .....                                         | 91         |
| Connector/Client Licensing .....                         | 93         |
| <b>Private Cloud Deployment .....</b>                    | <b>94</b>  |
| Minimum Requirements .....                               | 94         |
| Deploying on VMware vSphere .....                        | 94         |
| Prerequisites: .....                                     | 95         |
| Tasks .....                                              | 95         |
| Steps .....                                              | 95         |
| Decompressing an OVA file .....                          | 97         |
| Using Cloud-init with VMware vSphere .....               | 98         |
| Deploying in Hyper-V .....                               | 99         |
| Prerequisites: .....                                     | 99         |
| Tasks .....                                              | 99         |
| Steps .....                                              | 99         |
| Using Cloud-init with Hyper-V .....                      | 106        |
| Replacing SSH key before system can boot .....           | 107        |
| Changing the password for the initial user (ksctl) ..... | 108        |
| Using the Hard-coded password .....                      | 109        |
| Using an Auto-generated password .....                   | 109        |
| Licensing .....                                          | 110        |
| Lock Codes .....                                         | 110        |
| Connector/Client Licensing .....                         | 111        |
| <b>Physical Appliance Deployment .....</b>               | <b>112</b> |
| Minimum Requirements .....                               | 112        |
| Deploying on a Physical Appliance .....                  | 112        |
| Prerequisites .....                                      | 113        |
| Tasks .....                                              | 113        |
| Software Installation .....                              | 114        |
| Appliance Initialization .....                           | 114        |
| SafeNet Luna PCIe HSM Card Initialization .....          | 116        |
| HSM Root of Trust Configuration .....                    | 117        |

|                                                                    |            |
|--------------------------------------------------------------------|------------|
| Appliance Activation .....                                         | 118        |
| HSM Partition Creation .....                                       | 119        |
| Troubleshooting .....                                              | 119        |
| Replacing SSH key before system can boot .....                     | 119        |
| Changing the Password for the Initial User .....                   | 120        |
| System Administrator's Initial Private Key .....                   | 121        |
| Licensing .....                                                    | 123        |
| Lock Codes .....                                                   | 123        |
| Connector/Client Licensing .....                                   | 124        |
| <b>Appendix A: k470/k570 Physical Appliance Hardware .....</b>     | <b>125</b> |
| Physical Features .....                                            | 125        |
| Next Generation KeySecure Network Interface .....                  | 127        |
| System Behavior with Hardware Tamper Events .....                  | 128        |
| Tampering with the Appliance .....                                 | 128        |
| Decommission .....                                                 | 128        |
| What Happens When You Tamper - Including Opening the Fan Bay ..... | 128        |
| Power-on, Power-off, or Reboot the Appliance .....                 | 130        |
| Power On .....                                                     | 130        |
| Power Off .....                                                    | 130        |
| Reboot .....                                                       | 131        |
| Automatic Restart Following a Power Interruption .....             | 131        |
| Power Supply and Fan Maintenance .....                             | 131        |
| Replacing a Power Supply .....                                     | 132        |
| The Fans .....                                                     | 133        |
| Summary .....                                                      | 137        |
| HSM Emergency Decommission Button .....                            | 137        |
| What the Emergency Decommission Button Does .....                  | 138        |
| Serial Connections .....                                           | 138        |
| Front Locking Bezel .....                                          | 140        |
| Replacement Keys .....                                             | 141        |
| Power Consumption .....                                            | 141        |

|                                                            |     |
|------------------------------------------------------------|-----|
| Appendix B: Migrating objects from KeySecure Classic ..... | 143 |
| Overview .....                                             | 143 |
| User and Group Migration .....                             | 143 |
| Identifying objects to be Backed-up .....                  | 144 |
| Creating a Backup File .....                               | 145 |
| Migrating the Backup File .....                            | 149 |

# Overview

The Next Generation KeySecure server can be deployed in a number of cloud based environments as well as in a physical appliance.

## Product references made in this document

- > SafeNet Next Generation KeySecure (or NextGen KeySecure) - refers to all platforms running the Next Generation KeySecure software.
- > SafeNet Virtual KeySecure k170v (or simply k170v) - refers specifically to virtual and private cloud instances running NextGen KeySecure Software.
- > SafeNet KeySecure k470 and k570 Appliances (or simply k470 and k570) - refers specifically to the k470 and k570 physical appliances running the NextGen KeySecure Software.
- > SafeNet KeySecure Classic (or simply KeySecure Classic) - refers to the current generation of SafeNet KeySecure Appliances (models k450 and k460). These appliances can be upgraded to the NextGen KeySecure Software.

## Deployment environments

How the NextGen KeySecure is distributed and deployed is based on the environment. The supported environments are listed below along with the supported distribution method:

### Public Cloud Images:

- > Amazon AWS: SafeNet Cloud Provisioning System  
Refer to: ["Amazon Web Services Deployment" on page 56](#)
- > Microsoft Azure: Available as a BYOL image in the Microsoft Azure Marketplace  
Refer to: ["Microsoft Azure Deployment" on page 70](#)
- > Google Cloud:  
Refer to: ["Google Cloud Deployment" on page 84](#)

### Private Cloud Images:

- > Oracle VMware vSphere: OVA image file  
Refer to: ["Deploying on VMware vSphere" on page 94](#)
- > Microsoft Hyper-V: VHDX image file  
Refer to: ["Deploying in Hyper-V" on page 99](#)
- > OpenStack: qcow2 image file  
Please contact Gemalto Technical Support.



## Physical Appliances:

SafeNet KeySecure (Classic) models:

- > SafeNet KeySecure k450
- > SafeNet KeySecure k460

Refer to: ["Physical Appliance Deployment" on page 112](#)

SafeNet Next Generation KeySecure model:

- > SafeNet KeySecure k570

Refer to: ["Physical Appliance Deployment" on page 112](#)

## Audience

---

This document is intended for personnel responsible for deploying your SafeNet KeySecure server infrastructure.

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 security concepts.

## Document Conventions

---

This document uses standard conventions for describing the user interface and for alerting you to important information.

### Notes

Notes are used to alert you to important or helpful information. They use the following format:



**NOTE** Take note. Contains important or helpful information.

### Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss. They use the following format:



**CAUTION!** Exercise caution. Contains 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. They 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

| Format                     | Convention                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>bold</b>                | The bold attribute is used to indicate the following: <ul style="list-style-type: none"><li>&gt; Button names (Click Save As.)</li><li>&gt; Check box and radio button names (Select the Print Duplex check box.)</li><li>&gt; Dialog box titles (On the Protect Document dialog box, click Yes.)</li><li>&gt; Field names (User Name: Enter the name of the user.)</li><li>&gt; Menu names (On the File menu, click Save.) (Click Menu &gt; Go To &gt; Folders.)</li><li>&gt; User input (In the Date box, type April 1.)</li></ul> |
| mono-spaced                | Command-line commands and options (Type dir /p.)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <i>italics</i>             | In type, the italic attribute is used for emphasis or to indicate a related document. (See the <i>Installation Guide</i> for more information.)                                                                                                                                                                                                                                                                                                                                                                                      |
| <variable>                 | In command descriptions, angle brackets represent variables. You must substitute a value for command line arguments that are enclosed in angle brackets.                                                                                                                                                                                                                                                                                                                                                                             |
| [optional]<br>[<optional>] | Represent optional <b>keywords</b> 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.                                                                                                                                                                                                                                                                                                               |
| {a b c}<br><a> <b> <c>     | Represent required alternate <b>keywords</b> 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.                                                                                                                                                                                                                                                                                                                        |
| [a b c]<br>[<a> <b> <c>]   | Represent 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.                                                                                                                                                                                                                                                                                                                              |

# 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 Group Customer Support.

Thales Group 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 Group 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 Group Customer Support by telephone at **+1 410-931-7520**. 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](mailto:technical.support.dis@thalesgroup.com).

## Troubleshooting Issues in Conjunction with Customer Support

When making a support call, the KeySecure System Administrator (ksadmin) may be asked to log in using ssh to retrieve KeySecure Logs. KeySecure 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.

# Hardening Guidelines

The Next Generation KeySecure should be deployed into as secure an environment as possible. Every effort has been made to make the NextGen KeySecure as secure as possible, however, additional precautions should be taken especially when the NextGen KeySecure is deployed into an untrusted environment.

## Network Access - Security Group Rules

Security Group rules must be set up in the NextGen KeySecure environment to only permit access to required ports. For the list of available NextGen KeySecure ports, refer to the following Recommended Security Group Rules.

### Recommended Security Group Rules

The recommended Security Group Rules for the NextGen KeySecure are:

| Type                                                                | Protocol | Port Number           |
|---------------------------------------------------------------------|----------|-----------------------|
| SSH                                                                 | TCP      | 22 Inbound            |
| SSH (if using Luna Network HSM, DPoD Key Vault HSM or AWS CloudHSM) | TCP      | 22 Outbound           |
| HTTP                                                                | TCP      | 80 Inbound            |
| HTTPS                                                               | TCP      | 443 Inbound           |
| HTTPS (if using DPoD Key Vault HSM)                                 | TCP      | 443 Outbound          |
| PostgreSQL*                                                         | TCP      | 5432 Inbound/Outbound |
| NAE                                                                 | TCP      | 9000 Inbound          |
| KMIP                                                                | TCP      | 5696 Inbound          |
| If using Luna Network HSM, DPoD Key Vault HSM or AWS CloudHSM       | TCP      | 1792 Outbound         |
| NTP                                                                 | UDP      | 123 Outbound          |
| Syslog                                                              | UDP      | 514 Outbound          |
|                                                                     | TCP      | 6514 Outbound         |

| Type | Protocol | Port Number  |
|------|----------|--------------|
| SNMP | UDP      | 161 Inbound  |
|      | UDP      | 162 Outbound |

\* Applies only to Clustering. See ["Clusters and Nodes" on page 1](#)



**CAUTION!** Rules with source IP of 0.0.0.0/0 (IPv4) and ::/0 (IPv6) allow all IP addresses to access the instance. It is recommended to set security group rules to allow access from known IP addresses only.



**CAUTION!** Whenever a new interface is added, the respective port should be added to the security group also.

## Encryption

It is best practice to encrypt any NextGen KeySecure used in production. This is especially true if the NextGen KeySecure is deployed into an untrusted environment. When a NextGen KeySecure instance first boots, there are a number of secrets generated specific to that instance. To ensure that these secrets are never exposed, the NextGen KeySecure should be encrypted on first boot before it generates these. Please refer to ["Disk Encryption" on page 1](#) for details.

## System Administrative Key

The SSH key, used to access the System Administrative account "ksadmin", is extremely sensitive and should be kept in a secure environment.

## HSM Configurations

If configured to use an HSM (SafeNet Luna Network HSM, SafeNet DPoD Key Vault HSM or AWS CloudHSM), the NextGen KeySecure will protect all of its secrets with a non exportable HSM key. To protect all secrets, the NextGen KeySecure must be connected to the HSM on first boot. This is the most secure configuration. For clustering information, refer to the Administrator Guide > Concepts > Clusters and Nodes.

## TLS Compatibility

This table identifies the supported TLS versions for each of the NextGen KeySecure interfaces.

| Interface | Minimum TLS version | Maximum TLS version |
|-----------|---------------------|---------------------|
| Web UI    | TLS 1.2             | TLS 1.2             |
| NAE       | TLS 1.1             | TLS 1.2             |
| KMIP      | TLS 1.0             | TLS 1.2             |

## WARNING - Always use SSL/TLS with NAE

---

In a production environment, always enable SSL/TLS with the NAE interface. You should only disable SSL/TLS with NAE for troubleshooting purposes.

# Installing the CLI toolkit

Next Generation KeySecure includes a CLI toolkit, named **ksctl**, that can be downloaded and run locally to control a remote NextGen KeySecure.



**NOTE** **ksctl** is designed to be run from a remote system, not on the NextGen KeySecure itself.

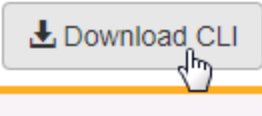
**ksctl** exclusively uses the REST API to communicate with NextGen KeySecure, so anything you can do with the CLI tool, you can also do directly with the REST API. Conversely, **ksctl** exposes most of the functionality of the REST API. It can perform management functions, such as adding users and groups, and end-user functions, such as creating keys.

To get started with **ksctl** you must do the following:

1. Download and unzip the **ksctl\_images.zip** file.
2. Setup the *ksctl-os* file for your system.
3. Run **ksctl**.

## To download and unzip the **ksctl\_images.zip** file:

1. Enter the IP address of your Next Generation KeySecure system in your browser.
2. On the Next Generation KeySecure start page, select **API & CLI Documentation** link.
3. At the top of the **API & CLI Documentation** page, click on **Download CLI** button:



Download CLI

4. Unzip the **ksctl\_images.zip** file.

Example set of available images in **ksctl\_images.zip** file:

```
ksctl-darwin-amd64
ksctl-linux-amd64
ksctl-win-amd64.exe
```

## To setup the *ksctl-os* for your system:

### Windows

1. After unzipping, rename either `ksctl-win-386.exe` or `ksctl-win-amd64.exe` to `ksctl.exe`.



**NOTE** **ksctl.exe** is a single executable with no dependencies, so it can be run from anywhere.



2. Move the **ksctl.exe** file to a folder easily accessed by the local command prompt.



**NOTE** You can also add the location of the **ksctl.exe** file to your PATH variable or copy **ksctl.exe** to a location already in your PATH.

3. Create a **ksctl** configuration file using the local command prompt **cmd.exe**:

- a. To find or confirm your **%HOMEPATH%** directory, enter:

```
$ set homedir
```

- b. Create a **.ksctl** directory in your **%HOMEPATH%** directory:

```
$ mkdir %HOMEPATH%\ksctl
```

```
$ cd %HOMEPATH%\ksctl
```

- c. In the **%HOMEPATH%\ksctl** directory, create a **config.yaml** file with the following items:

```
KSCTL_USERNAME: admin
```

```
KSCTL_PASSWORD: <your password>
```

```
KSCTL_URL: https://<keysecure ipaddress>
```

```
KSCTL_NOSSLVERIFY: true
```

## Linux

1. After unzipping, rename the file of choice to **ksctl** and move it to a directory within your PATH.



**NOTE** **ksctl** is a single executable with no dependencies, so it can be run from anywhere.

2. Create a **ksctl** configuration file:

- a. Create a **.ksctl** directory in your **\$HOME** directory:

```
$ mkdir $HOME/.ksctl
```

```
$ cd $HOME/.ksctl
```

- b. In the **\$HOME/.ksctl** directory create a **config.yaml** file with the following items:

```
KSCTL_USERNAME: admin
```

```
KSCTL_PASSWORD: <your password>
```

```
KSCTL_URL: https://<keysecure ipaddress>
```

```
KSCTL_NOSSLVERIFY: true
```

3. Bash completions can be generated by typing **ksctl bashcomp** at the bash prompt and following the instructions.

## To override ksctl configuration file

Summary of **ksctl** configuration variables for the config.yaml configuration file:

```
KSCTL_USERNAME: <your username>
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<keysecure ipaddress>
KSCTL_NOSSLVERIFY: <true or false>
KSCTL_CONNECTION: <LDAP connection name>
KSCTL_VERBOSITY: <true or false for verbosity>
KSCTL_JWT: <JWT>
KSCTL_RESPFMT: <response output format, the default "json" is the only supported value at present>
KSCTL_TIMEOUT: <timeout in seconds for TCP connection attempts>
```

**ksctl** is easiest to use with the configuration file (config.yaml). However, those parameters can be overridden by using these flags on the command line:

```
--user (replaces KSCTL_USERNAME)
--password (replaces KSCTL_PASSWORD)
--url (replaces KSCTL_URL)
--nosslverify (replaces KSCTL_NOSSLVERIFY)
```



**NOTE** If you wish to override KSCTL\_NOSSLVERIFY it must be set to false in the configuration file (or removed).

```
--connection (replaces KSCTL_CONNECTION)
--verbose (replaces KSCTL_VERBOSITY)
--jwt (replaces KSCTL_JWT)
--respfmt (replaces KSCTL_RESPFMT)
--timeout int (replaces KSCTL_TIMEOUT)
```

## To run ksctl:



**NOTE** This is an example only; not all supported CLI commands and flags are shown.

1. At the command line prompt, enter **ksctl**. You will get the following example output:

```
$ ksctl
This command line utility can be used to interface with a KeySecure device.
```

```
Configuration parameters can be used to pass in the username, URL, etc.
These parameters can be passed to the application in several ways.
```

```
When the configuration parameters are available from multiple sources,
the order of precedence (highest to lowest) is:
(a) command line arguments,
```

- (b) configuration file passed via the command line,
- (c) environmental variables, and
- (d) configuration file in the default location.

The parameters passed in as command line arguments override everything else.

The structure of the configuration file must be based on 'config\_example.yaml'. The name of the default configuration file is 'config.yaml'. It should be located in the '.ksctl' subdirectory of the user's home directory. On Linux, this file should be located here: '~/ksctl/config.yaml'.

If defined, proxy environment variables (i.e. http\_proxy) will be honored for all functions except 'init getpw'. 'init getpw' internally uses ssh which means that ssh must be configured to be proxy aware for the users local operating system.

#### Usage:

```
ksctl [command]
```

#### Available Commands:

|             |                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------|
| alarms      | Execute KeySecure alarms commands.                                                              |
| backup      | Execute KeySecure backup commands.                                                              |
| backupkeys  | Execute KeySecure backup keys commands.                                                         |
| bashcomp    | Generate bash completion script file for all commands and flags.                                |
| ca          | Execute KeySecure certificate authority commands.                                               |
| changepw    | User ability to change their own password.                                                      |
| clientmgmt  | Execute KeySecure client management commands.                                                   |
| cluster     | Execute KeySecure cluster management commands.                                                  |
| connections | LDAP connections management.                                                                    |
| crypto      | Execute KeySecure cryptographic data processing commands that use keys stored in the KeySecure. |
| diskenc     | Execute KeySecure disk encryption commands.                                                     |
| gendocs     | Generate man pages and documentation for all commands and flags.                                |
| .           |                                                                                                 |
| .           |                                                                                                 |
| .           |                                                                                                 |

#### Flags:

```
--configfile string 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>-h, --help</code>          | help for ksctl                                                                                                   |
| <code>--jwt string</code>        | The JSON web token (JWT) can be passed instead of user/password (optional). 'ksctl tokens create' creates a JWT. |
| <code>--nossllverify</code>      | Do not verify the certificate for SSL/HTTPS authentication (not recommended)                                     |
| <code>--password string</code>   | KeySecure 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>--url string</code>        | KeySecure Server URL                                                                                             |
| <code>--user string</code>       | KeySecure Server User Name                                                                                       |
| <code>-v, --verbose</code>       | Provide verbose output while executing command (optional)                                                        |

Use "ksctl [command] --help" for more information about a command.

## 2. Try creating a key by entering the command `ksctl keys create --autoname`:

```
$ ksctl keys create --autoname
```

This will create output similar to the following:

```
{
 "id": "b979d9c7-1374-4b97-a274-dfaa20f8139b",
 "uri": "kylo:kylo:vault:keys:keys-f0f3ellhqh5r7xavk3l0-rx-ktkmwgzf3-rudppxrw4-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2016-12-15T15:35:51.322996318Z",
 "name": "Keys-f0F3eLlhQh5R7XAVK3l0_rx-KtkmWGzF3-rUDPpXRW4",
 "updatedAt": "2016-12-15T15:35:51.322996318Z",
 "material": "",
 "usage": "blob",
 "meta": null,
 "version": 0,
 "algorithm": "AES",
 "size": 256,
 "format": "raw",
 "unexportable": false,
```

```
"undeletable": false,
"publickey": ""
}
```

# Cloud-init Configuration Options

This section identifies cloud-init configuration options to be considered in advance of deployment.

- > ["Using Cloud-init" below](#)
- > ["Configuring Initial Password for the Initial User" on page 33](#)
- > ["Configuring Default CA and Server Certificates" on page 34](#)
- > ["Configuring IPv6" on page 36](#)
- > ["Configuring the NextGen KeySecure with a Static IP" on page 36](#)
- > ["Configuring Disk Encryption" on page 37](#)
- > ["Configuring HSM Integration" on page 39](#)

## Using Cloud-init

The Next Generation KeySecure supports initial boot time configuration using cloud-init. On first boot, the appliance can be passed a configuration text file, known as a **user-data** file. The NextGen KeySecure reads this file to retrieve various configuration settings such as disk encryption, IP configuration and HSM configuration.

How a cloud-init configuration file is passed to the NextGen KeySecure appliance depends on the environment in which it runs. Below are examples for each of the supported environments.



**NOTE** Cloud-init configuration (user-data) files uses YAML syntax; indentation is important and tabs can not be used.

## Passing the cloud-init file in AWS

When launching the instance from AWS web interface, at Step 3 - Configure Instance Details, select 'As file' in the 'User data' section and browse to 'config.dat' file. Then continue with the rest of the launch steps.

If launching the instance using AWS CLI, then use the '--user-data' parameter with the run-instances command. Use the 'file://' prefix to pass in the user data from 'config.dat' file.



**NOTE** Verify the user data is 16KB or less, otherwise it will be ignored by AWS. If it is larger then remove unnecessary whitespace (e.g. ident lines with a single space) and comments.

## Passing the cloud-init file in Azure

When launching the Azure instance using Azure CLI, use the '-d' or '--custom-data' with the `vm create` command to pass in the user data from 'config.dat' file.

## Passing the cloud-init file for a Private Cloud Image

When using the disk image, how the cloud-init data is passed to Next Generation KeySecure depends on the virtualization platform - please refer to documentation or notes for your specific cloud environment.

The following are two examples of passing cloud-init data when using a disk image, one using 'libvirt' and one using VMware/vSphere.

### Example using 'libvirt'

When launching a virtual machine with the Qcow2 image using 'libvirt', the cloud-init data has to be passed in as an ISO file. The ISO can be generated as follows:

1. Prepare the **user-data** file as follows:

a. Rename the file **config.data** to **user-data**.

b. Since, user's ssh key is used for wrapping a layer of encryption keys, it must be added to the cloud-init config. So, the 'user-data' file should look like:

```
#cloud-config
hostname: <host name for the instance>
diskenc:
 encrypt: true
ssh_authorized_keys:
 - <replace with user's ssh public key>
```



**NOTE** 'ssh\_authorized\_keys' can be configured with multiple ssh public keys.

2. Create a **meta-data** file and provide instance parameters, for example:

```
instance-id: <some instance id>
```

3. Create an ISO image file:

a. Make sure **genisoimage** utility is installed.

b. Create the ISO file.

```
$ genisoimage -o config.iso -volid cidata -joliet -rock user-data meta-data
```

4. Launch instance using **virt-install**, for example:

```
$ virt-install --virt-type kvm --name <virtual image name> --ram 2048 --disk path=<path to keysecure qcow2 image>,size=16,format=qcow2 --disk path=<path to config.iso> --network network=default --graphics vnc,listen=0.0.0.0 --noautoconsole --os-type=linux --os-variant=ubuntu16.04 --import
```

### Example using VMware/vSphere

This example describes how to deploy Next Generation KeySecure on VMware with a static IP configuration. In general, if you have virtual machines you intend to use frequently or for extended periods of time, it can be convenient to assign a static IP address, or configure DHCP server to always assign the same IP address, to

each of these virtual machines.

For virtual machines that you do not expect to keep for extended periods of time, use DHCP and let it allocate IP addresses for these machines.

Use the following procedure to deploy NextGen KeySecure on VMware with a static IP configuration.



**NOTE** This procedure includes preparation of a cloud-init configuration file used to set up a static IP address during launch of the NextGen KeySecure.

## Step 1 - Get the Installer

Get NextGen KeySecure installation file for VMware from Gemalto Support Portal.

## Step 2 - Deploy OVA in Esxi Server

- a. Select “Deploy OVF Template”.
- b. On the **Select an OVF template** page, choose OVA file and select **NEXT**.

Deploy OVF Template

**1 Select an OVF template**

2 Select a name and folder

3 Select a compute resource

4 Review details

5 Select storage

6 Ready to complete

Select an OVF template

Select an OVF template from remote URL or local file system

Enter a URL to download and install the OVF package from the Internet, or browse to a location accessible from your computer, such as a local hard drive, a network share, or a CD/DVD drive.

☐ URL

[http | https://remoteserver-address/filetodeploy.ov](http://https://remoteserver-address/filetodeploy.ov)

☒ Local file

Choose Files 610-000121-001...1.4.0\_RevA.ova

CANCEL BACK NEXT

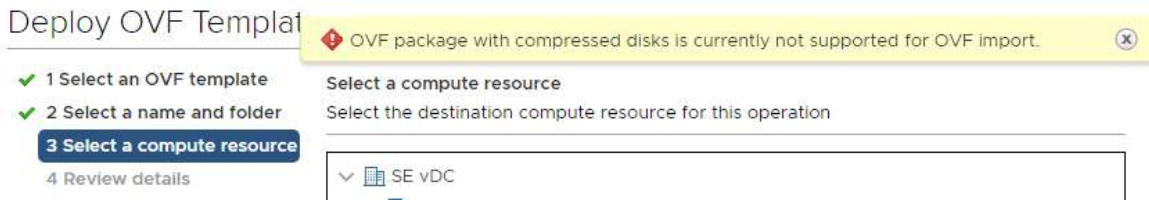
## Step 3 - Select VM

On the **Select a name and folder** page, select the name of the virtual machine and its location.



## Step 4 - Validate the NextGen KeySecure Virtual Machine Configuration

On the **Select a compute resource** page, select the destination compute resource (if applicable).



If you see the above error, please perform the following steps:

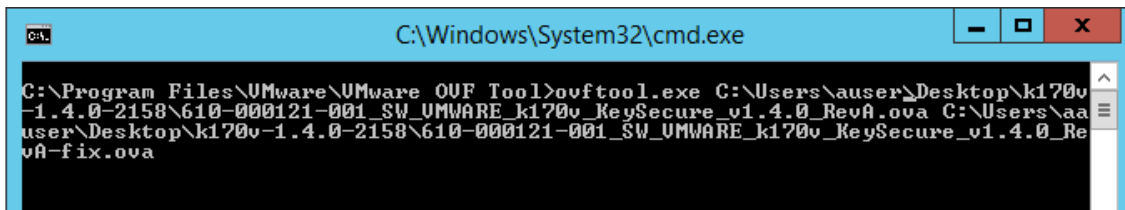


**NOTE** OVF with compressed disk is not supported on newer version of Vsphere client. It may work on older versions.

- a. Use `ovftool.exe` to convert ova file into uncompressed file(s).

Execute the following command:

```
ovftool.exe --lax <source_OVA_file> <destination_OVF_file>
```



- b. Repeat process from Step 2 with new installer file.

## Step 5 - Review details

On the **Review details** page, verify the template details of KeySecure and if correct, select **NEXT**.

## Deploy OVF Template

- ✓ 1 Select an OVF template
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- 4 Review details**
- 5 Select storage
- 6 Select networks
- 7 Ready to complete

### Review details

Verify the template details.

|               |                             |
|---------------|-----------------------------|
| Publisher     | No certificate present      |
| Product       | KeySecure k170v             |
| Version       | 1.4.0.2158                  |
| Vendor        | Gemalto NV                  |
| Download size | 1.8 GB                      |
| Size on disk  | 4.9 GB (thin provisioned)   |
|               | 16.0 GB (thick provisioned) |

CANCEL

BACK

NEXT

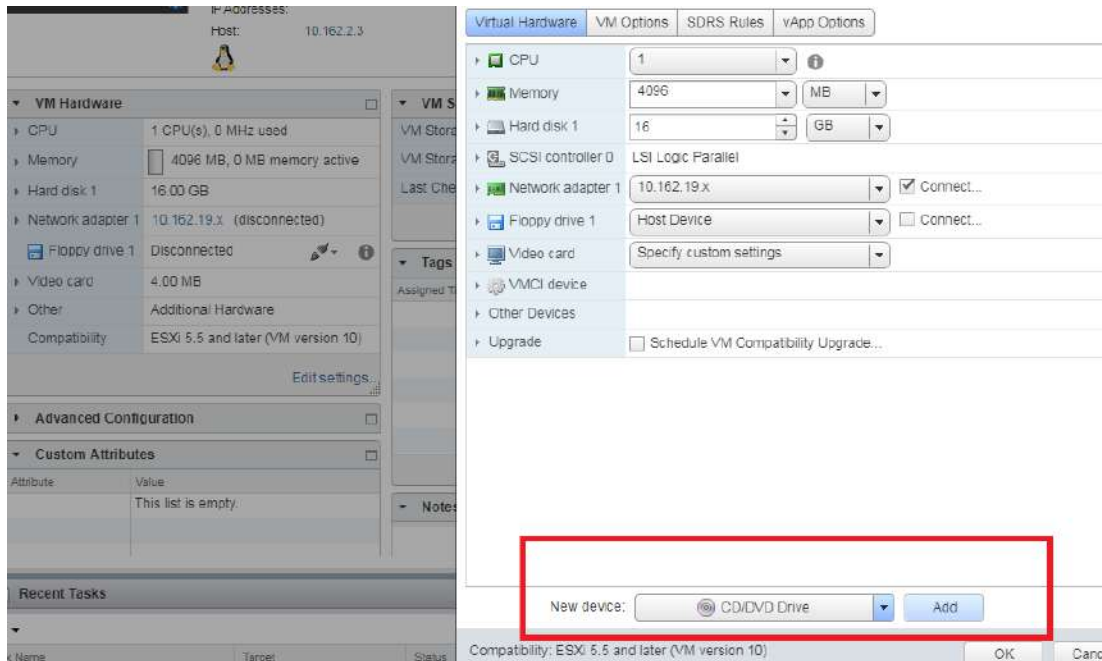
On the **Select storage** page, to the storage on which install the NextGen KeySecure and the select **NEXT**.  
On the **Select network** page, select the network and then select **FINISH**.



**NOTE** Do not launch/start the machine at this time. Continue with the following steps to configure cloud-init.

## Step 6 - Prepare cloud-init configuration

- a. Add a CD drive to the VM.



- b. Before booting up the VM, we prepare the cloud-init configuration. The following cloud-init example configures the VM's eth0 port with a static IP address.

```
keysecure:
```

```
 netcfg:
```

```
 iface:
```

```
 name: eth0
```

```
 type: static
```

```
 address: 192.168.1.150
```

```
 netmask: 255.255.255.0
```

```
 gateway: 192.168.1.1
```

```
 dns1: 192.168.1.100
```



**NOTE** Update the above configuration to the desired network settings.



**NOTE** Cloud-init configuration files use YAML syntax; indentation is important and tabs cannot be used.

- c. Copy the above configuration, update it as needed and convert the string to base64. To convert to base64, use the `openssl` command:

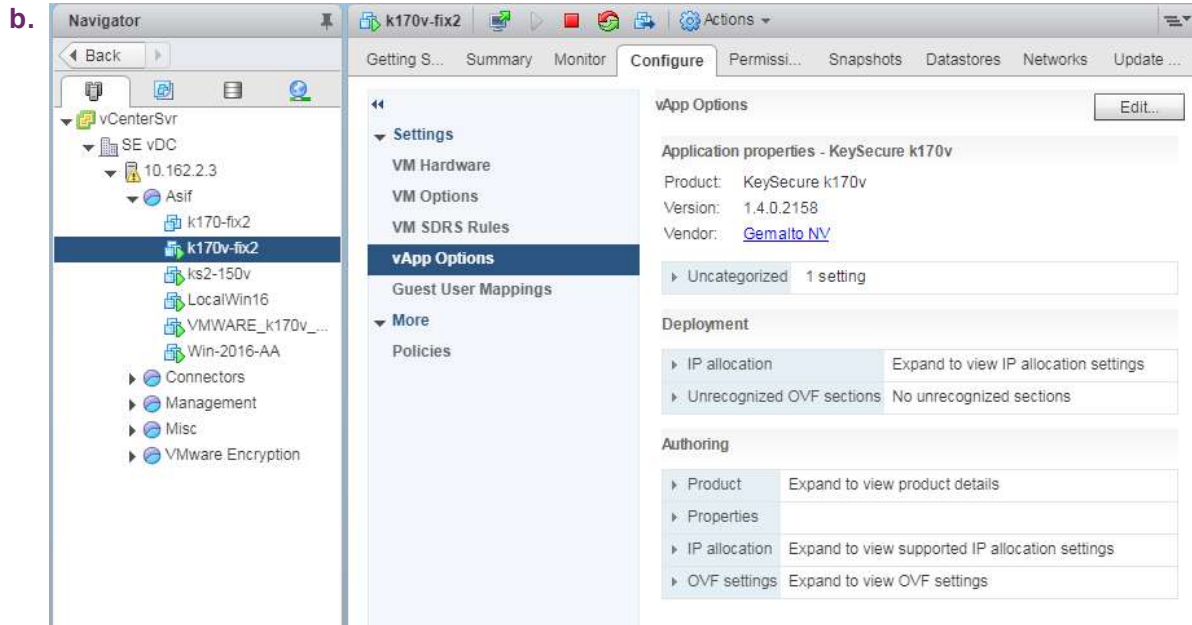
```
$ openssl base64 -in <infile> -out <outfile>
```

- d. Save this base64 string to use in next steps.

## Step 7 - Add configuration

In this step, we provide base64 configuration to the VM to boot up. In this guide, vSphere web client (Flash) version is being used. You may find similar options in other clients.

a. Select: *virtual machine* > Configure > Settings > vApp Options



Press **Edit** button on top right on this page.

c. Under OVF Settings, select the check box “ISO Image”, which is next to OVF environment transport.

**kesecure 170v-2 - Edit Settings**

Virtual Hardware | VM Options | SDRS Rules | **vApp Options**

Product: KeySecure k170v  
 Version: 1.4.0.2158  
 Vendor: [Gemalto NV](#)

*There are no properties to edit*

**Deployment**

|                             |                                          |
|-----------------------------|------------------------------------------|
| ▶ IP allocation             | <i>Expand for IP allocation settings</i> |
| ▶ Unrecognized OVF sections | <i>No unrecognized sections</i>          |

**Authoring**

|                           |                                                                    |                                                                                                        |
|---------------------------|--------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| ▶ Product                 | Name                                                               | KeySecure k170v                                                                                        |
| ▶ Properties              |                                                                    |                                                                                                        |
| ▶ IP allocation           |                                                                    | <i>Expand to edit supported IP allocation settings</i>                                                 |
| ▶ OVF settings            |                                                                    |                                                                                                        |
| OVF environment           | <input data-bbox="566 951 699 989" type="button" value="View..."/> | The OVF environment is only available when the VM is powered on.                                       |
| OVF environment transport | <input checked="" type="checkbox"/> ISO image                      | An ISO image, containing the OVF environment document, is mounted on the first available CD-ROM drive. |
|                           | <input type="checkbox"/> VMware Tools                              | The VMware tools guestInfo.ovfEnv variable is initialized with the OVF environment document.           |
| Installation boot         | <input type="checkbox"/> Enable                                    | The installation boot automatically gets                                                               |

Compatibility: ESXi 5.5 and later (VM version 10)

d. On the same page, expand "Properties" to add configuration.

**kesecure 170v-2 - Edit Settings**

Virtual Hardware | VM Options | SDRS Rules | **vApp Options**

Product: KeySecure k170v  
 Version: 1.4.0.2158  
 Vendor: [Gemalto NV](#)

*There are no properties to edit*

**Deployment**

- IP allocation *Expand for IP allocation settings*
- Unrecognized OVF sections *No unrecognized sections*

**Authoring**

- Product Name: KeySecure k170v

**Properties**

| Key | Label | Category | Type |
|-----|-------|----------|------|
|     |       |          |      |
|     |       |          |      |
|     |       |          |      |
|     |       |          |      |
|     |       |          |      |

New Edit Delete

Compatibility: ESXi 5.5 and later (VM version 10)

OK Cancel

- e. Press the **New** button to add a property for the configuration. On following screen, there are two fields which need to be changed.
- **Label:** user-data
  - **Default value:** <base64 string of configuration, generated in step6-b >



**NOTE** Key ID will change automatically when you change Label.

**Edit Property Settings**

Category: [Dropdown]

Label: **user-data**

Key class ID: [Empty]

Key ID: user-data

Key instance ID: [Empty]

Description: [Empty]

Type: ☒ Static property

Type: String

Length: 0 - 65535

Default value: **3kbnMxOiAxMC4xNjluMTkuMTAwCg==**

User configurable: ☒

☐ Dynamic property

Macro: IP address

Network: 10.162.19.x

OK Cancel

- f. Press **OK** to save the Property Settings.
- g. Then press **OK** again to save the vApp Options page.

## Step 8 - Launch

Launch the instance. The VM should boot up configured with a static IP.

## Configuring Initial Password for the Initial User

The initial password for the initial Application Administrator can be auto-generated using a cloud-init configuration file containing the following setting:

```
#cloud-config
keysecure:
 initial-user:
 generate-password: true
```



**NOTE** Cloud-init configuration (user-data) files uses YAML syntax; indentation is important and tabs can not be used.

## Configuring Default CA and Server Certificates

Two certificates are created by default during initial boot-strapping of the Next Generation KeySecure, one Self-signed Root CA Certificate and one Server Certificate that is signed by the Root CA.



**NOTE** NextGen KeySecure does not support `dns/ip atm`; only the `emails` field is supported, regardless of the `oid` setting.

### Configuring initial default certificates

Before the first boot of the system, the content of these certificates can be overwritten with help of a cloud-init configuration as follows:



**NOTE** Cloud-init configuration (user-data) files uses YAML syntax; indentation is important and tabs can not be used.

```
#cloud-config
keysecure:
 default-ca:
 CN: <Common Name of Root CA Certificate, for example "Cloud-init test CA">
 emails:
 - <Subject Alternate Name E-Mail, for example "foo@example.com">
 - ...
 names:
 - C: <Country, for example "US">
 ST: <State/province, for example "TX">
 L: <Location, for example "Austin">
 O: <Organization, for example "Gemalto Inc.">
 OU: <Organizational Unit, for example "Gemalto">
 - ...
 default-server-cert:
 CN: <Common Name of Server Certificate, for example "Cloud-init test server cert">
```



```

emails:
 - <Subject Alternate Name E-Mail, for example "foo@example.com">
 - ...
names:
 - C: <Country, for example "US">
 ST: <State/province, for example "TX">
 L: <Location, for example "Austin">
 O: <Organization, for example "Gemalto Inc.">
 OU: <Organizational Unit, for example "Gemalto">
 - ...

```

A working cloud-init configuration could look as follows:

```

#cloud-config
keysecure:
 default-ca:
 CN: Cloud-init test CA
 emails:
 - certadmin@example.com
 - support@example.com
 names:
 - C: US
 ST: TX
 L: Austin
 O: Gemalto Inc.
 OU: Gemalto
 - C: Sweden
 L: Gothenburg
 O: Gemalto AB
 default-server-cert:
 CN: Cloud-init test server cert
 emails:
 - certadmin@example.com
 - support@example.com
 names:
 - C: US
 ST: MD
 L: Belcamp
 O: Gemalto Inc.
 OU: Gemalto
 - C: Sweden
 L: Stockholm
 O: Gemalto AB

```

The only required field is 'CN'. If it is missing the rest are ignored. If any of the remaining are missing, they will be empty when the certificate is created. If there is no default-ca or default-server-cert section, the defaults will be used for the missing section.

## Configuring IPv6

---

There are two formats for IPv6 configuration, 'auto' and 'dhcp'. All systems come configured to use 'auto' as the default.

To set 'dhcp' as the default, add the following cloud-init directive to the user-data:

```
#cloud-config:
keysecure:
 netcfg:
 config_ipv6: dhcp
```



**NOTE** Cloud-init configuration (user-data) files uses YAML syntax; indentation is important and tabs can not be used.



**NOTE** This parameter must be set before the first boot of the system since it is ignored after the first boot.



**NOTE** Do not use 'dhcp' on AWS if there are no IPv6 interfaces defined. This will result in a five-minute delay during each boot of your software.

## Configuring the NextGen KeySecure with a Static IP

---

The NextGen KeySecure OVA and Qcow2 disk images can be configured with a Static IP using cloud-init. The configuration format is as follows:



**NOTE** Static IP is only supported via cloud-init, and must be done on initial boot of a service.

```
#cloud-config
keysecure:
 netcfg:
 iface:
 name: <interface name for e.g. eth0>
 type: <static>
 address: <ip address to be assigned>
 netmask: <net mask in x.x.x.x format>
```

```
gateway: <gateway ip address>
dns1: <dns server's ip address, optional>
dns2: <alternate dns server's ip address, optional>
```



**NOTE** Cloud-init configuration (user-data) files uses YAML syntax; indentation is important and tabs can not be used.

## Configuring Disk Encryption

For added security, the disk of the Next Generation KeySecure appliance can be fully encrypted. Encryption can either be initiated when an instance is first launched or on an already launched instance. Because installation specific secrets are generated the first time a NextGen KeySecure appliance is run, it is recommended that the instance be encrypted at launch time to ensure these secrets are never exposed.



**\*\*WARNING\*\*** After encrypting the disk, you will need to unlock the encrypted instance on every boot using the 'ksctl diskenc secureboot' command. See ["To unlock an encrypted instance" on page 39](#).

### Encrypting an instance at first launch

To configure the instance to encrypt at launch, use the cloud-init settings below.

```
#cloud-config
diskenc:
 encrypt: true
```



**NOTE** Cloud-init configuration (user-data) files uses YAML syntax; indentation is important and tabs can not be used.

This file must be passed to the NextGen KeySecure instances, which uses cloud-init to enable the encryption. How this file is passed to the NextGen KeySecure appliance depends on the environment in which it is run. For other examples of passing cloud-init config.dat file in supported environments, go to "Deployment Guide > Cloud-init Configuration Options > Using Cloud-init.

### Encrypting an already launched instance

The following are examples for encrypting an already launched instance and for checking on it's encryption status. Also provided is a CLI example for unlocking the instance at boot time.

#### To encrypt a disk image

When using the disk image, how the cloud-init data is passed to Next Generation KeySecure depends on the virtualization platform.

For example, when launching a virtual machine with the Qcow2 image using 'libvirt', the cloud-init data has to be passed in as an ISO file. The ISO can be generated as follows:

1. Prepare the **user-data** file as follows:

- a. Rename the file **config.data** to **user-data**.
- b. Since, user's ssh key is used for wrapping a layer of encryption keys, it must be added to the cloud-init config. So, the 'user-data' file should look like:

```
#cloud-config
diskenc:
 encrypt: true
ssh_authorized_keys:
 - <replace with user's ssh public key>
```



**NOTE** 'ssh\_authorized\_keys' can be configured with multiple ssh public keys.

2. Create a **meta-data** file and provide instance parameters, for example:

```
$ instance-id: <some instance id>
$ local-hostname: <host name for the instance>
```

3. Create an ISO image file:

- a. Make sure **genisoimage** utility is installed.
- b. Create the ISO file.

```
$ genisoimage -o config.iso -volid cidata -joliet -rock user-data meta-data
```

4. Launch the NextGen KeySecure instance and attach the ISO created in the previous step before powering on the NextGen KeySecure appliance.



**NOTE** To encrypt the disk via cloud-init, the ISO must be mapped to the NextGen KeySecure appliance before the first power-on or first boot-up operation.

**Example launch in OpenStack using virt-install:**

```
$ virt-install --virt-type kvm --name <virtual image name> --ram 2048 --disk path=<path to
keysecure qcow2 image>,size=16,format=qcow2 --disk path=<path to config.iso> --network network=
default --graphics vnc,listen=0.0.0.0 --noautoconsole --os-type=linux --os-variant=ubuntu16.04
--import
```

## To check encryption status

To check the encryption progress, you can run following CLI command:

```
$ ksctl diskenc status -p
```

### Example:

```
$ ksctl diskenc status -p
```

This returns the following response:

```
Encrypting...
```

```
14.81 GiB / 15.52 GiB [=====>-----] 95.44% 11s
```

### To unlock an encrypted instance

The instance will start up after the encryption has finished. To unlock the instance, you must run following CLI command:

```
$ ksctl diskenc secureboot -i <ssh key used to launch the instance> -u https://<instance dns name>
```

Every time an encrypted instance boots, the above CLI command must be executed to unlock the instance.

## Configuring HSM Integration

The Next Generation KeySecure can be configured to use an HSM as a root of trust. This can be done using cloud-init.



**NOTE** HSM configuration is now performed through the API. Using the cloud-init method is not recommended because it exposes sensitive information, such as the partition password, that can be viewed by anyone with access to the associated AWS console. For more information on HSM support, refer to the NextGen KeySecure Administrator Guide > Concepts > Hardware security Module.



**NOTE** Before connecting an HSM to the NextGen KeySecure, the HSM must be set up and configured by an HSM Administrator. This includes creating a partition and partition password, and generating a client certificate for the NextGen KeySecure.

To enable HSM integration using cloud-init, user-data is supplied to cloud-init. This file must be passed to the Next Generation KeySecure instances that uses cloud-init to enable the encryption. How this file is passed to the Next Generation KeySecure appliance depends on the environment in which it is run. For a list of supported environments, see ["Overview " on page 10"Using Cloud-init" on page 24](#)



**NOTE** Cloud-init configuration (user-data) files uses YAML syntax; indentation is important and tabs can not be used.

### Example - Configuring NextGen KeySecure to use SafeNet Luna Network HSM:

```
#cloud-config
keysecure:
 hsm:
```

```

type: luna

Replace dummy partition name and password values below with your existing Luna HSM
settings.
conninfo: '{"partition_name": "kylo-partition","partition_password": "sOmeP@ssword"}'

config:
 # luna host name or ip address
 host: 172.20.32.11

 # luna server PEM encoded cert
 server-cert: |
 -----BEGIN CERTIFICATE-----
 ...
 -----END CERTIFICATE-----

 # luna client PEM encoded cert
 client-cert: |
 -----BEGIN CERTIFICATE-----
 ...
 -----END CERTIFICATE-----

 # luna client PEM encoded cert private key
 client-cert-key: |
 -----BEGIN RSA PRIVATE KEY-----
 Proc-Type: 4, ENCRYPTED
 DEK-Info: DES-EDE3-CBC, 92852AD5AFE768B8
 ...
 -----END RSA PRIVATE KEY-----

```

### Example - Configuring NextGen KeySecure to use AWS CloudHSM (cavium):

```

#cloud-config
keysecure:
 hsm:
 type: aws

 # Replace dummy crytpo user credentials below with your actual CloudHSM CU credentials.
 conninfo: '{"partition_name": "cavium","partition_password": "hsmuser:password"}'

 config:

```

```
CloudHSM Cluster ENI IP Address
host: 172.20.32.11

CloudHSM Cluster certificate
server-cert: |
 ----BEGIN CERTIFICATE----
 ...
 ----END CERTIFICATE----
```

## Example - Configuring NextGen KeySecure to DPoD Key Vault HSM:



**NOTE** DPoD cloud configurations are sometimes above the AWS size limit for user data. Verify the user data is 16KB or less, otherwise it will be ignored by AWS. If it is larger then remove unnecessary whitespace (e.g. ident lines with a single space) and comments.

```
#cloud-config
keysecure:
 hsm:
 # type: Specifies HSM type as dpod.
 type: dpod

 # conninfo: "partition_name" and "partition_password"
 # {
 # "partition_name": '', # HSM partition name provided during partition provisioning
 # 'partition_password': '', # HSM Crypto Officer password
 # }
 conninfo: '{"partition_name": "kylo-partition","partition_password": "sOmeP@ssword"}'

 # config: include the following attributes:
 # {
 # 'server_cert': '', # Server certificate in PEM format from server-certificate.pem
 # 'partition_cert': '', # Partition certificate in PEM format from partition-
 # 'partition_ca_cert': '', # Partition CA certificate in PEM format from partition-
 # 'cv_app_specific_data': '', # CVAppSpecificData value from Chrystoki.conf file
 # 'cv_partition_data': '', # PartitionData00 value from Chrystoki.conf file
 # }
 #
```

```
config:
 server_cert: |
 -----BEGIN CERTIFICATE-----
 ...
 -----END CERTIFICATE-----
 partition_cert: |
 -----BEGIN CERTIFICATE-----
 ...
 -----END CERTIFICATE-----
 partition_ca_cert: |
 -----BEGIN CERTIFICATE-----
 ...
 -----END CERTIFICATE-----
 cv_app_specific_data: <CVAppSpecificData>
 cv_partition_data: <PartitionData00>
```



# System Configuration Utility (kscfg)

The Next Generation KeySecure has a 'CLI based' System Configuration Utility (**kscfg**). The "ksadmin" user can remotely access the NextGen KeySecure **kscfg** utility in a private cloud deployment by using SSH, or in a physical appliance deployment, by directly connecting to the appliance's console port and using your `ksadmin` password.

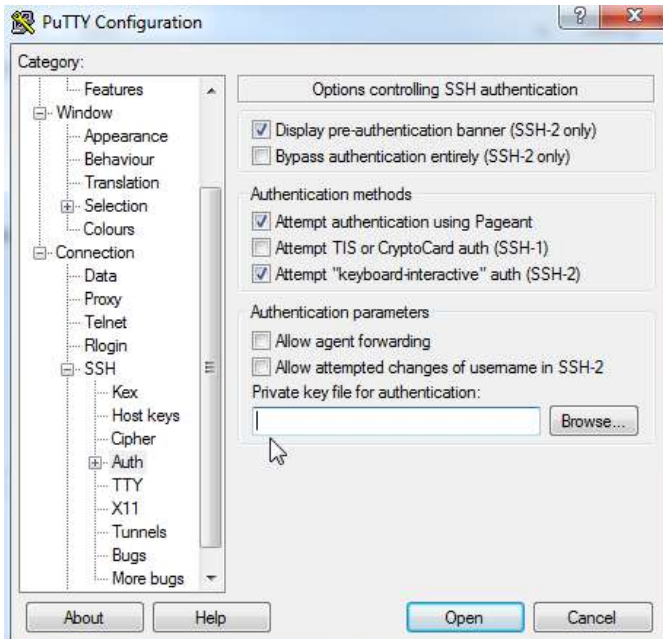
## Logging in as ksadmin user

To log in as `ksadmin`, you must first connect to the KeySecure console.

### To connect and log in to the console (for private cloud deployments):

Using SSH, you can remotely connect to the console port of a NextGen KeySecure instance deployed in a private cloud (e.g. VMware vSphere and HyperV).

1. Using an SSH utility (e.g. PuTTY) select an SSH session and enter the IP address assigned to the NextGen KeySecure instance during deployment. This is the same IP address used to browse to the GUI.
2. If using PuTTY, make sure your SSH keys are in ppk format. If they are in PEM format, you can convert them to ppk (e.g., using PuTTYgen utility).
3. Using the SSH utility, select the path to your SSH Private Key you will use to authenticate the session.



4. Select **Open** to start the SSH session.

---

## To connect and log in to the console (for physical appliance deployments):

Using a serial cable, you can directly connect your console device (e.g., laptop) to the console port of a physical appliance (k450, k460, k470 and k570).

1. Connect the serial cable from your console device to the physical appliance console port.
2. Log in to the physical appliance as `ksadmin` user using the password you created during Appliance Initialization; refer to ["Appliance Initialization" on page 114](#)

---

## Using the `kscfg` utility

The NextGen KeySecure **`kscfg`** utility can be used to retrieve network interfaces (NICs) configurations and values, and to perform a hard system reset.

All configurable network interfaces are always listed and are the same as those available from standard Linux network interfaces utilities such as `ip` and `ifconfigmcli`. The network interface names from **`kscfg`** match the network interface names from the operating system.

A network interface contains two configurable families: `inet` (IPv4 and `inet6` (IPv6)). Their methods are:

- > "none"  
For a disabled network family.
- > "dhcp"  
Use DHCP to automatically acquire a network address. ("auto" might be preferred for IPv6.)
- > "static"  
Statically set a network address.
- > "auto"  
Automatically setup IPv6 from the network environment. (IPv6 only)

---

## Commands

### To view the available `kscfg` commands:

#### Example:

```
$ kscfg --help
```

#### Response:

```
Command-line interface application for configuring the KeySecure.
```

#### Usage:

```
kscfg [command]
```

#### Available Commands:

|        |                                 |
|--------|---------------------------------|
| help   | Help about any command          |
| net    | KeySecure network configuration |
| system | KeySecure system commands       |

Flags:

|               |                                                           |
|---------------|-----------------------------------------------------------|
| -h, --help    | help for kscfg                                            |
| -v, --verbose | Provide verbose output while executing command (optional) |

## Network Configuration

---

### To list the available network interfaces:

**Usage:**

```
kscfg net interfaces list
```

**Example:**

```
$ kscfg net interfaces list
```

**Response:**

```
{
 "skip": 0,
 "limit": 0,
 "total": 3,
 "resources": [
 {
 "name": "eth0",
 "inet": {
 "method": "static",
 "ip": "10.121.105.137",
 "netmask": "255.255.252.0",
 "gateway": "10.121.104.1",
 "dns": [
 "172.16.2.12"
]
 },
 "inet6": {
 "method": "none"
 }
 },
 {
 "name": "eth1",
 "inet": {
```

```

 "method": "dhcp",
 "ip": "10.121.105.81",
 "netmask": "255.255.252.0",
 "gateway": "10.121.104.1",
 "dns": [
 "172.16.2.12",
 "10.121.8.7",
 "172.16.2.13"
]
 },
 "inet6": {
 "method": "none"
 }
},
{
 "name": "eth2",
 "inet": {
 "method": "none"
 },
 "inet6": {
 "method": "none"
 }
}
]
}

```

### To view information on a specific network interface:

#### Usage:

```
kscfg net interfaces get [flags]
```

#### Flags:

```

-h, --help help for get
-n, --name string A network interface name such as 'enp0s25'.

```

#### Example:

```
$ kscfg net interfaces get -n eth0
```

#### Response:

```

{
 "name": "eth0",
 "inet": {
 "method": "static",
 "ip": "10.121.105.137",

```

```

 "netmask": "255.255.252.0",
 "gateway": "10.121.104.1",
 "dns": [
 "172.16.2.12"
],
 },
 "inet6": {
 "method": "none"
 }
}

```

## To modify the configuration of a specific network interface:



**CAUTION!** This operation has been deprecated. Please use NetworkManager's `nmcli` tool to modify a network interface's configuration; refer to: ["Network Interface Configuration and Bonding" on page 50](#)

### Usage:

```
kscfg net interfaces modify [flags]
```

### Flags:

|                                     |                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-d, --dhcp</code>             | Use DHCP for the network interface. Deprecated - use "method" instead.                                                                                                                                                                                                                                                                                                                  |
| <code>-r, --dns string</code>       | IP addresses of the DNS servers (comma separated), or "" to unset and use entries in /etc/resolv.conf.                                                                                                                                                                                                                                                                                  |
| <code>--force-gateway string</code> | Force system default gateway update, i.e. overwrite system default gateway when this device is brought up. By default a network interface will only set the system default gateway if is not already set. This feature can be used to force a specific network interface to be used for outgoing traffic initiated from the machine itself. Set to "yes" to enable and "no" to disable. |
| <code>-g, --gateway string</code>   | Default gateway, or "" to unset.                                                                                                                                                                                                                                                                                                                                                        |
| <code>-h, --help</code>             | help for modify                                                                                                                                                                                                                                                                                                                                                                         |
| <code>-4, --inet</code>             | Use IPv4 for the network interface. (default true)                                                                                                                                                                                                                                                                                                                                      |
| <code>-6, --inet6</code>            | Use IPv6 for the network interface.                                                                                                                                                                                                                                                                                                                                                     |
| <code>-i, --ipaddress string</code> | Static IP Address.                                                                                                                                                                                                                                                                                                                                                                      |
| <code>-e, --method string</code>    | Method for obtaining an IP. Accepted inet values are dhcp, none, or static; inet6 values are auto, dhcp, none, or static. (default "static")                                                                                                                                                                                                                                            |
| <code>-n, --name string</code>      | A network interface name such as 'enp0s25'.                                                                                                                                                                                                                                                                                                                                             |

`-m, --netmask string` Subnet mask. IPv4 must be an IP (e.g. 255.255.255.0). IPv6 must be the number of bits (e.g. 64).

### Example:

```
$ kscfg net interfaces modify --name eth0 --ipaddress 10.121.105.27 --netmask 255.255.252.0 --gateway 10.121.104.1 --dns 172.16.2.12
```

### Response:

```
{
 "name": "eth0",
 "inet": {
 "method": "static",
 "ip": "10.121.105.27",
 "netmask": "255.255.252.0",
 "gateway": "10.121.104.1",
 "dns": [
 "172.16.2.12"
]
 },
 "inet6": {
 "method": "none"
 }
}
```

## System Reset

The 'kscfg system reset' command can be used to perform a hard reset of the NextGen KeySecure.



**\*\*WARNING\*\*** This destructive operation wipes all data on the NextGen KeySecure and should be used with care.

Normally, the REST API or the CLI should be used for performing the reset. This method of performing the reset should be used as a last resort. This operation deletes all backup keys and the HSM configuration. It is good practice to do the following before running this command:

1. Create and download a backup of the database.
2. Download all the backup keys. Any backups downloaded from this device will not be useful without the backup keys.

## Usage

```
$ kscfg system reset [flags]
```

Flags:

`-f, --force` When this flag is set, any errors encountered during reset are ignored, and the reset procedure continues to the end. This flag must be used with care as it could place the system in an unuseable state. It should be used when all else fails.

`-h, --help` help for reset

`-y, --yes` When this flag is set, all user prompts during the reset process are skipped. A default value of 'yes' is used as the automatic response to all prompts.

## Examples

```
$ kscfg system reset [-f] [-y]
```

Response:

This will perform a full reset of the KeySecure services.

WARNING - This is a destructive operation and will wipe all data in the KeySecure.

It will delete all backupkeys and the HSM configuration.

Normally, the REST API or the CLI should be used for performing the reset.

THIS METHOD OF PERFORMING THE RESET SHOULD BE USED AS A LAST RESORT.

It is good practice to perform the following steps prior to running this command:

1. Create and download a backup of the database.

2. Download all the backupkeys; any backups downloaded from this device will not be useful without the backupkeys.

Do you want to continue? [y/N] y

This will take some time, please wait

Device reset has started. It will take a few minutes to complete.

\$

# Network Interface Configuration and Bonding

This chapter provides a primer on configuring multiple network interfaces on the Next Generation KeySecure and bonding these interfaces to achieve redundancy and performance improvements.

The NextGen KeySecure uses GNOME NetworkManager and its CLI tool `nmcli` to configure network interfaces. For in-depth details regarding these tools, refer to GNOME official documentation:

- > GNOME NetworkManager
- > GNOME nmcli tool



**NOTE** Modifying a network interface's configuration should only be done on a Physical Appliance or in a Private Cloud, VMware vSphere, HyperV, etc. Network interface configuration on public clouds should be done using the cloud provider's tools.



**CAUTION!** Modifying a remote network interface over SSH is risky. The remote connection will stop responding if the IP address settings are incorrectly configured, resulting in the remote machine being unreachable.

## Typical Network Configuration Tasks

This section provides an example of typical network configuration steps using `nmcli`.

### To get started with `nmcli`

1. Start by using `nmcli` to list the available network devices (aka network interfaces, NICs, etc.). The devices listed in this example are truncated for brevity.

```
$ nmcli device
DEVICE TYPE STATE CONNECTION
kylo0 bridge connected kylo0
eth0 ethernet connected eth0
eth1 ethernet disconnected --
veth05d02c4 ethernet unmanaged --
...
```

The output of this command may show over 30 devices. However, only a few require explanation:

- `kylo0` is used for internal communication by NextGen KeySecure services. It should never be altered in any way.
- `eth0` is an active device with a NetworkManager connection with the same name.
- `eth1` is an inactive device with no NetworkManager connection defined.



- `veth05d02c4` is an unmanaged device and should be ignored because NetworkManager cannot be used to control or configure the device.

As a general rule, device names starting with `eth` or `en` are devices an administrator may configure.

2. Next, use `nmcli` to see an active device's live values. In this example the only active device so far is `eth0`.

```
$ nmcli device show eth0
GENERAL.DEVICE: eth0
GENERAL.TYPE: ethernet
GENERAL.HWADDR: 00:50:56:99:3F:54
GENERAL.MTU: 1500
GENERAL.STATE: 100 (connected)
GENERAL.CONNECTION: eth0
GENERAL.CON-PATH: /org/freedesktop/NetworkManager/ActiveConnection/8
WIRED-PROPERTIES.CARRIER: on
IP4.ADDRESS[1]: 10.121.105.97/22
IP4.GATEWAY: 10.121.104.1
IP4.DNS[1]: 10.121.8.7
IP4.DNS[2]: 172.16.2.13
IP4.DNS[3]: 172.16.2.12
IP6.ADDRESS[1]: fe80::bd7e:b93f:7e66:4b92/64
IP6.GATEWAY:
```

The output shows that IPv4 has an address and accompanying values and that IPv6 only has a link-local address.

3. Next, use `nmcli` to see the configured network devices that NetworkManager calls *connections*. Notice that the connections listed in this example do not include device `eth1` because it is not yet configured.

```
$ nmcli conn
NAME UUID TYPE DEVICE
eth0 9d86421b-7032-48eb-ac5a-3c84d285d01e 802-3-ethernet eth0
kylo0 78765519-b051-4d85-a9bd-3a681ce3c9cf bridge kylo0
```

A connection `NAME` and a `DEVICE` name are the same in this example. However, the connection name does not need to match the device name and may be any string, such as "Wired connection 1".

4. Next, use `nmcli` to create a connection configuration for device `eth1` using DHCP for IPv4 and no IPv6 configuration. When `ipv4.method` and/or `ipv6.method` are not explicitly provided, then NetworkManager uses `auto` by default.

```
$ nmcli conn add type ethernet con-name eth1 ifname eth1 -- ipv4.method auto ipv6.method ignore
```

5. Next, use `nmcli` to confirm the connection is created with the correct configuration for device `eth1`. The configuration and active values in this example are truncated for brevity.

```
$ nmcli conn show eth1
connection.id: eth1
connection.uuid: d797d28c-fe8a-49ab-8181-271870d6cfc6
connection.interface-name: eth1
```

```

connection.type: 802-3-ethernet
...
ipv4.method: auto
...
ipv6.method: ignore
...
IP4.ADDRESS[1]: 10.121.105.113/22
IP4.GATEWAY: 10.121.104.1
IP4.DNS[1]: 10.121.8.7
IP4.DNS[2]: 172.16.2.13
IP4.DNS[3]: 172.16.2.12
...

```

The output format uses lowercase key names to indicate configuration values (e.g. `ipv4.method`) and uppercase key names to indicate live values (e.g. `IP4.ADDRESS[1]`). From this output we can see the newly configured device's IP address obtained via DHCP is 10.121.105.113.

6. Next, try connecting to `eth1`'s address from a browser and confirm that NextGen KeySecure UI loads. If the UI does not load, try pinging the address to confirm the IP address can be reached. If neither works, then double check the connection values are as expected.
7. Now that a connection is created for `eth1`, if it needs to be altered, then the `modify` sub-command may be used. Use `nmcli` to modify device `eth1`'s connection to use a static IP address instead of DHCP.

```

$ nmcli conn modify eth1 ipv4.method manual ipv4.addresses 10.121.105.18/22 ipv4.gateway
10.121.104.1 ipv4.dns 8.8.8.8,8.8.4.4
$ nmcli conn show eth1 | grep IP4.ADDRESS
IP4.ADDRESS[1]: 10.121.105.113/22

```

8. Notice that the `IP4.ADDRESS[1]` field listed in the second command still has the original IP address obtained via DHCP. In order to activate the modification, the connection must be restarted using the `up` sub-command:

```

$ nmcli conn up eth1
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/15)
$ nmcli conn show eth1 | grep IP4.ADDRESS
IP4.ADDRESS[1]: 10.121.105.18/22

```

## Configuring Static Routes

To configure a static route:

1. Modify the NetworkManager connection to use a static IP address.

Run the command: `nmcli conn modify <network device> +ipv4.routes "<static IP address> <gateway>"`

**Example:**

```
$ nmcli conn modify eth0 +ipv4.routes "192.168.122.0/24 172.30.1.1"
```

## 2. Apply the configuration change.

Run the command: `$ nmcli conn up <network device>`

### Example:

```
$ nmcli conn up <network device>
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/10)
```

## 3. Verify that the route is active.

```
$ ip route
default via 172.30.1.1 dev eth0 proto static
172.30.0.128/26 dev kylo0 proto kernel scope link src 172.30.0.129
172.30.1.0/24 dev eth0 proto kernel scope link src 172.30.1.71
192.168.122.0/24 via 172.30.1.1 dev eth0 proto static
```

The output shows that the route is now active.

# Network Interface Bonding

A bonded network interface provides redundancy and performance improvements by aggregating two or more network interfaces to form a single logical interface.

The official [Linux kernel HOWTO](#) documents this feature in detail.

The [nmcli's reference documentation](#) covers details specific to NetworkManager configuration.



**NOTE** Bonding network interfaces is intended for physical appliances, for example, SafeNet KeySecure k570. It is generally not relevant to k170v instances, as bonding is not useful in a virtual environment.

Start by using `nmcli` to create a bonded connection for fail-over only (the most basic type of bond).

```
$ nmcli conn add type bond con-name bond0 ifname bond0 -- bond.options mode=active-backup
Connection 'bond0' (c6661f6b-5d68-4ffa-9ae8-63f0e4224cb4) successfully added.
$ nmcli conn show bond0
...
bond.options: mode=active-backup
...
```

Alternatively, for certain networking environments such as in a virtual environment, the default setting of `fail_over_mac=0` might not be optimal and setting `fail_over_mac=1` can be an option. Please refer to the [Linux Kernel HOWTO](#) for more information. The command to create the bonded connection would then be:

```
$ nmcli conn add type bond con-name bond0 ifname bond0 -- bond.options mode=active-backup,fail_
over_mac=1
Connection 'bond0' (c6661f6b-5d68-4ffa-9ae8-63f0e4224cb4) successfully added.
```

```
$ nmcli conn show bond0
```

```
...
```

```
bond.options: mode=active-backup,fail_over_mac=1
```

```
...
```

Next, use `nmcli` to list devices and add some of them to the bond.

```
$ nmcli device
```

| DEVICE | TYPE | STATE | CONNECTION |
|--------|------|-------|------------|
|--------|------|-------|------------|

```
...
```

|      |          |           |      |
|------|----------|-----------|------|
| eth0 | ethernet | connected | eth0 |
|------|----------|-----------|------|

|      |          |              |    |
|------|----------|--------------|----|
| eth1 | ethernet | disconnected | -- |
|------|----------|--------------|----|

```
...
```

Create the bond connections for `eth0` and `eth1` by specifying `bond0` as their master and bring them up. Notice that `eth0` has an existing connection profile, so delete it first to ensure only the bond connection profile is used.

```
$ nmcli conn delete eth0
```

```
Connection 'eth0' (f4e38090-6d3c-4612-b5b5-2d4c6a0c361d) successfully deleted.
```

```
$ nmcli conn add type ethernet con-name eth0-bond0 ifname eth0 master bond0
```

```
Connection 'eth0-bond0' (3d87c66e-a128-4e4f-897a-b31cf1b5fdbf) successfully added.
```

```
$ nmcli conn add type ethernet con-name eth1-bond0 ifname eth1 master bond0
```

```
Connection 'eth1-bond0' (b488dlf4-4a73-43ee-8a7f-9e992a850903) successfully added.
```

```
$ nmcli conn up eth0-bond0
```

```
Connection successfully activated (D-Bus active path:
```

```
/org/freedesktop/NetworkManager/ActiveConnection/7)
```

```
$ nmcli conn up eth1-bond0
```

```
Connection successfully activated (D-Bus active path:
```

```
/org/freedesktop/NetworkManager/ActiveConnection/8)
```

Next, use `nmcli` to bring up the bonded interface and confirm that it received an IP address.

```
$ nmcli conn show bond0
```

```
...
```

```
connection.type: bond
```

```
...
```

```
IP4.ADDRESS[1]: 10.121.105.113/22
```

```
...
```

Verify the bonded interface continues to receive and transmit data when either of the enslaved network interfaces are down (e.g. by removing the cable, disabling the port in the switch, etc.).

It is recommended to restart the OS to verify that the bond persist across reboots. If it does not then check if a connection profile exist for any of the interfaces that compose this bond and have an equal or higher connection priority. Either delete the duplicate connection profile or increase the priority of the bonded connections so it has precedence.

## Configuring NIC Bonding with Static MAC Address

---

For NextGen KeySecure appliances, NIC bonding can result in alternating MAC address after reboot. If a static MAC address is preferred, it can be set on the virtual bonding network interface to ensure it never changes after reboot. The MAC address can, for example, be taken from the first NIC such as `eth0`.

After the NIC bonding is setup, a static MAC address can be set with the following command:

```
$ nmcli connection modify bond0 802-3-ethernet.cloned-mac-address "00:0C:29:56:F8:A2"
```



**NOTE** This feature should be used together with `fail_over_mac=0` on the bonded network interface, the default setting.

# Amazon Web Services Deployment

This chapter describes how to deploy a Next Generation KeySecure within Amazon Web Services (AWS).

This chapter contains the following topics:

- > ["Minimum Requirements" below](#)
- > ["Provisioning the AMI" below](#)
- > ["Deploying in AWS" on page 59](#)
- > ["Changing the Initial User Password" on page 66](#)
- > ["Licensing" on page 67](#)
- > ["Enhanced Networking" on page 69](#)

## Minimum Requirements

To deploy a SafeNet Virtual KeySecure k170v instance, the system volume requirements must be at least 100 GB in a production environment and 30 GB when evaluating the product.



**NOTE** The system volume holds all data as well as backups.

### Here are the minimal system requirements for AWS:



**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.

System volume: 1 volume (auto created 100 GB)

Memory: 16 GB

CPUs: 2

NICs: 1

## Provisioning the AMI

An AWS region-specific, and confidential SafeNet Virtual KeySecure k170v image (AMI) is obtained by provisioning. Provisioning places a k170v AMI in your target AWS account. Separate provisioning requests must be made if you need multiple k170v instances in one or multiple regions.

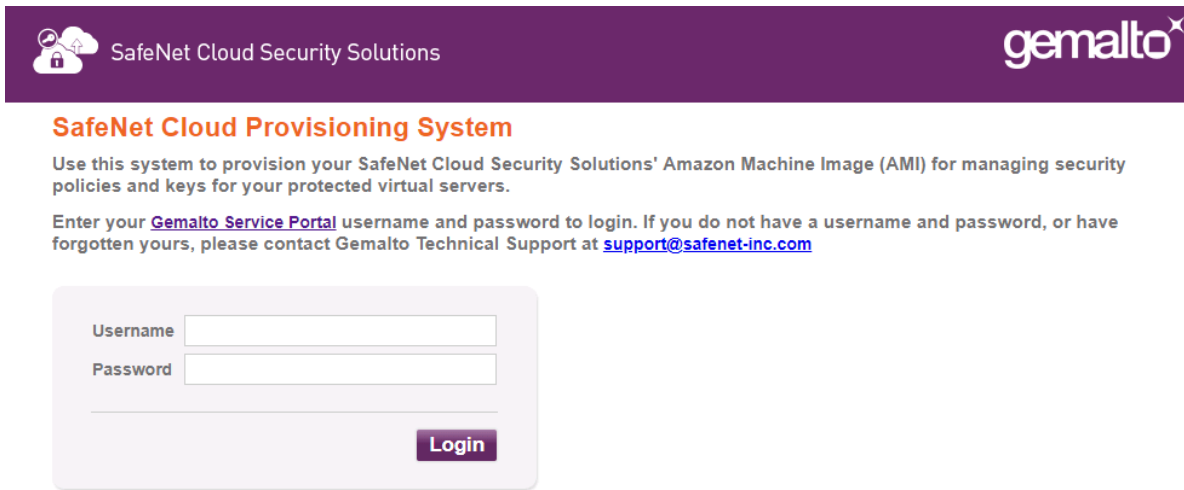
Use the **SafeNet Cloud Provisioning System**, to customize a k170v AMI for a specific region.



**NOTE** Only SafeNet-authorized Technical Support Customer Portal users can access the **SafeNet Cloud Provisioning System**.

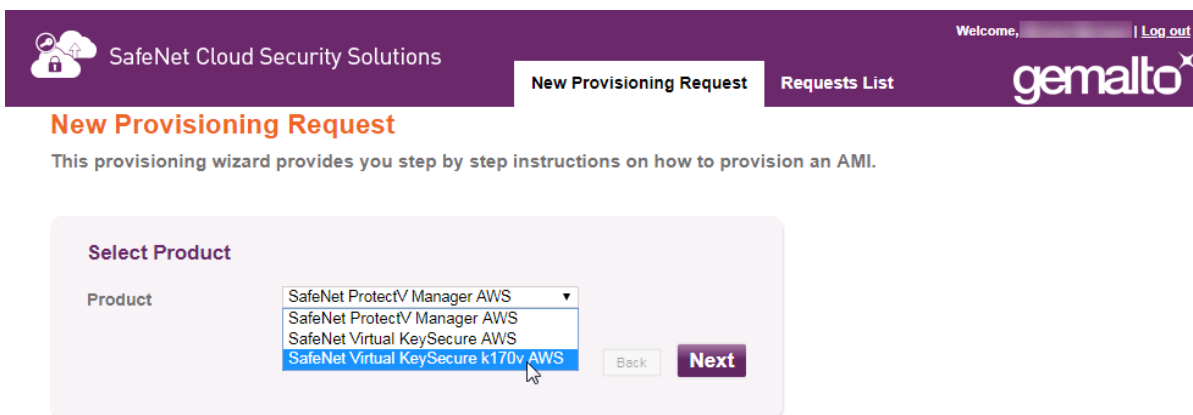
## Steps to provision a SafeNet Virtual KeySecure k170v

1. Enter URL: `https://cps.safenet-inc.com/app` to access the **SafeNet Cloud Provisioning System** Login page:



2. Enter your Gemalto Service Portal **Username** and **Password**, and click **Login**.

This will bring you to the following **New Provisioning Request** page:



**NOTE** Notice the two tabs at the top of the page, **New Provisioning Request** tab, currently open, and **Request List** tab for checking on the status of your submitted requests.

3. Select the product **SafeNet Virtual KeySecure k170v AWS** and click **Next**.

You are then asked to select the product version:

**Select Product Version**

Version

**Back****Next**

**NOTE** Generally, it is recommended that you select the latest version at the top of the list.

4. Select **Select Product Version** and click **Next**.

The following Provisioning Credentials screen is displayed:

**Provisioning Credentials**

This step will start provisioning of an AMI in the region specified below and access will be available to the AWS Account specified below.

AWS Region AWS Account Number **Back****Submit**

5. Select your AWS Region and enter your AWS Account Number. Then click **Submit**.

An **Image request created** confirmation message is displayed with a Request reference ID as show here.

**Image request created**Request reference ID: 

You will also receive a short 'do-not-reply' email from Gemalto.

6. Using the **Requests List** tab, you can track the request using the reference ID under the **Request ID** column. Initially the status shows **pending**, then it is changed to **dispatched** and finally it displays **created**.

After the image is provisioned, you will see your NextGen KeySecure AMI ID in the **Request List** table. You may need to refresh you screen to see the updated table.

You will also receive another 'do-not-reply' email with the NextGen KeySecure AMI ID. The provisioned AMI will then be shared with your AWS account for a minimum two weeks.

7. You can now launch a NextGen KeySecure instance on AWS. Refer to section: ["Deploying in AWS" on the next page.](#)



# Deploying in AWS

This section provides the steps for deploying a SafeNet Virtual KeySecure k170v instance in AWS.

## Prerequisites

- > Gemalto entitlement email: You receive this email after requesting a Trial License or purchasing a Term or Perpetual License. Contact Gemalto Sales representative for assistance in obtaining a license. This email contains:
  - Entitlement ID
  - Link to the EMS Customer Portal to enter your SafeNet Virtual KeySecure k170v instance Lock Code and receive your license.
- > A k170v AMI provisioned using the **SafeNet Cloud Provisioning System**. See ["Provisioning the AMI" on page 56](#).
- > If using a Windows client,
  - use PuTTY or similar utility to SSH to your k170v instance as KeySecure Administrator (ksadmin).
  - If needed, use PuTTYgen or similar utility to format the SSH Key Pair.
- > If using a Linux client: use, SSH to login as KeySecure Administrator (ksadmin).

## Tasks

Deploying a SafeNet KeySecure k170v instance in AWS consists of these tasks:

1. Launch the k170v instance in AWS and access the Lock Code for this instance.
2. Link to the EMS Customer Portal to enter your k170v instance Lock Code and receive your license.
3. Apply your license to your k170v instance.

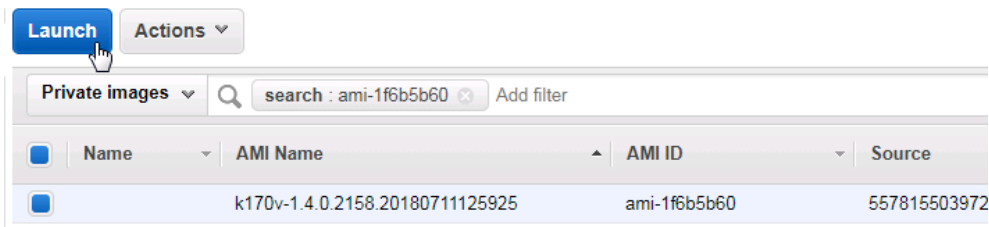
## Steps

### Steps to launch a SafeNet KeySecure k170v instance in AWS and access the Lock Code.

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. See ["Provisioning the AMI" on page 56](#).



4. In the left pane, under the **Images**, select **AMIs** and search for the AMI (e.g. ami-1f6b5b60) that was provisioned for you by the **SafeNet Cloud Provisioning System**.



5. Select your image and then click on **Launch**.
6. In the **Choose an Instance Type** screen, choose **t2.medium** or larger and then select **Next: Configure Instance Details**.
7. In the **Configure Instance Details** screen,
  - a. Create a new **Network** and **Subnet** or select one that you have already created.
  - b. At the bottom of the screen there is a section **Advanced Details**.  
Any information that you need configured before the first boot must be entered here in the user data area. See ["Cloud-init Configuration Options" on page 24](#)).
  - c. Select **Next: Add Storage**.
8. In the **Add Storage** screen, select the desired **Size (GiB)**.
  - Enter **100** GiB if this is for Production.
  - Enter **20** GiB if this is for lab use.
 Select **Next: Add Tags**.
9. In the **Add Tags** screen, select **Add Tag**
  - a. Enter Tag: Name
  - b. Value: <something meaningful to the operator>, e.g "k170v 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 ["Network Access - Security Group Rules" on page 15](#). Consider adding all of the recommended rules.



**NOTE** To launch the NextGen KeySecure, you need at a minimum Port 22 inbound/outbound for **ksadmin**'s 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

☒ 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. 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.)
- Proceed without a key pair (**Do not select this option.**)

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.

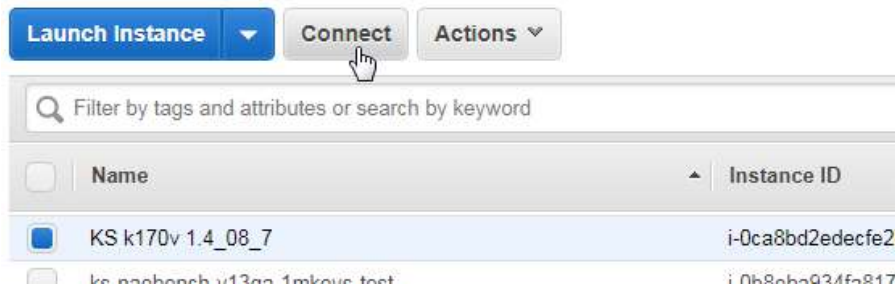
At this point, if you wish to connect as System Administrator (ksadmin) to perform system level duties, duties such as performing network configurations, system upgrades, continue with the next step.



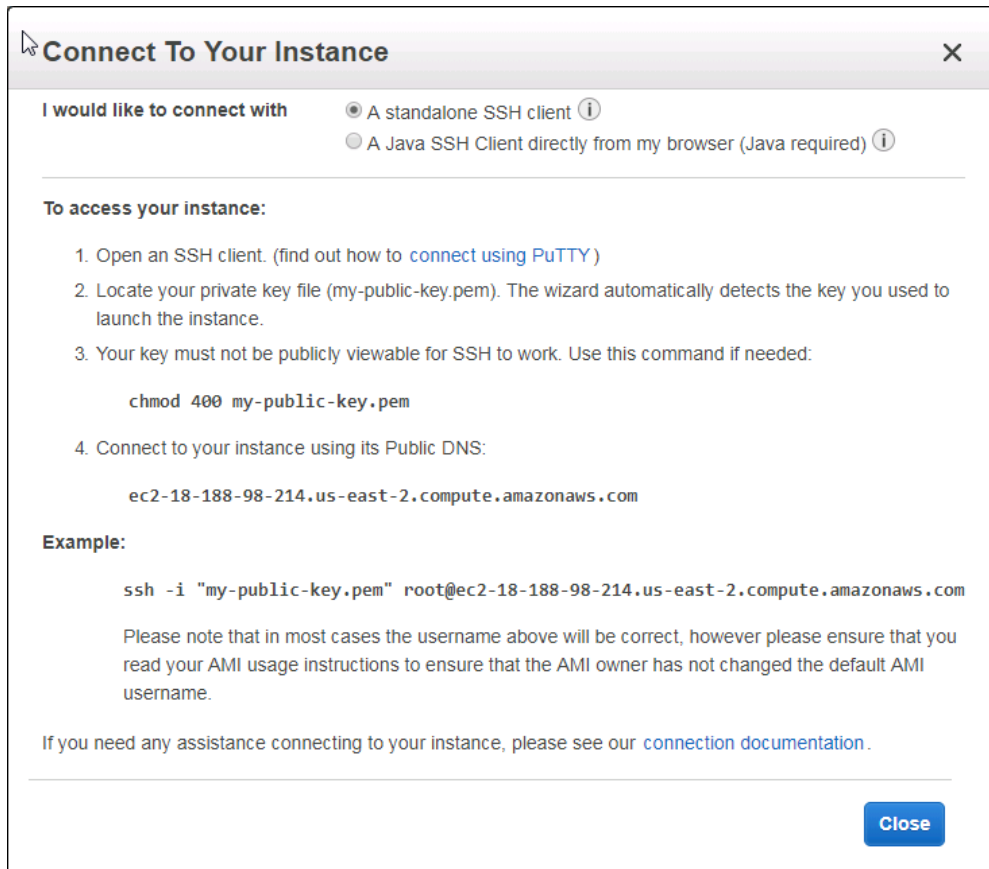
**NOTE** Refer to the 'Administrator Guide > Concepts > Groups for a more detailed list of ksadmin duties.

Otherwise, browse to step 18 to connect to the **SafeNet KeySecure k170v** Web Page.

**14.** Select your instance in the table and then select **Connect**.



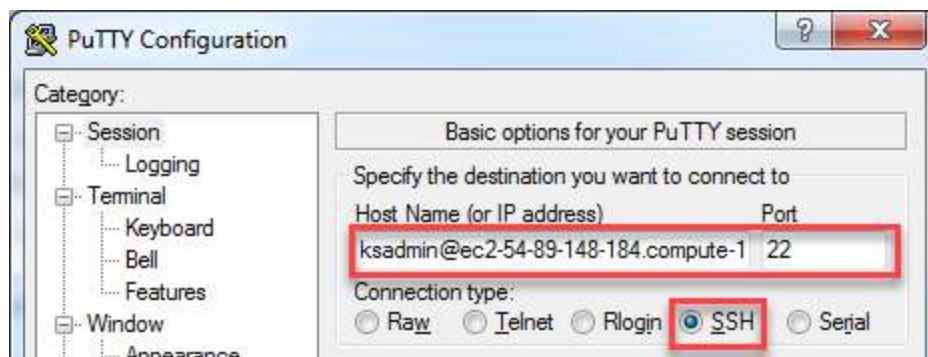
The **Connect To Your Instance** dialog box appears:



**15.** Copy the instance **Public DNS** string, for example: `ec2-54-89-148-184.compute-1.amazonaws.com`

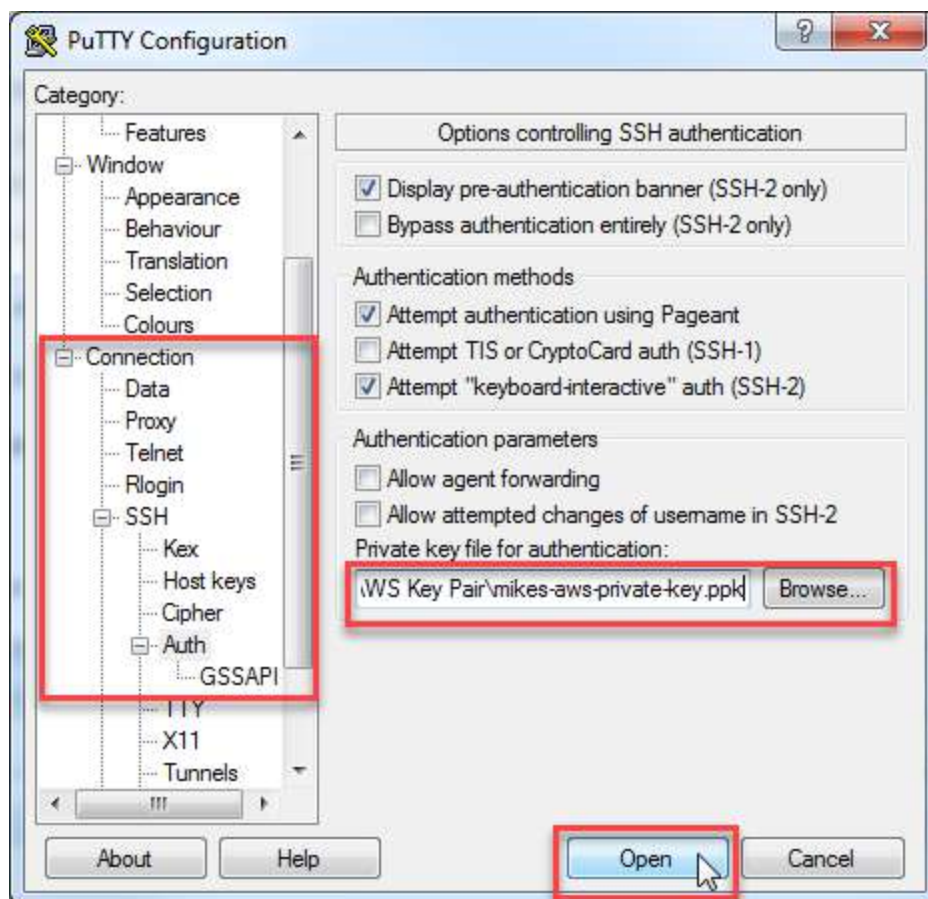
**16.** If using PuTTY, begin a new session.

- a.** Paste the Public DNS in the **Host Name (or IP address)** field and preface this string with "ksadmin@".
- 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**. The following k170v **ksadmin** screen appears:





Username

username or connection/username

Password

password

Log In

By clicking Log In, you are accepting [this EULA](#)

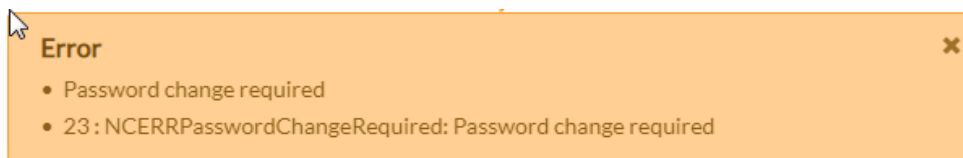
SafeNet KeySecure k170v  
version 1.10.0

[API & CLI Documentation](#)

[Product Documentation](#)

**19.** Log in using the initial default credentials: Username = admin, Password = admin

The following error is displayed:



The **Error** displayed is normal and simply requires you to immediately change the default Password.

**20.** Enter a new password using this default Password Policy:

Min length: 8

Max length: 30

Min number of upper cases: 1

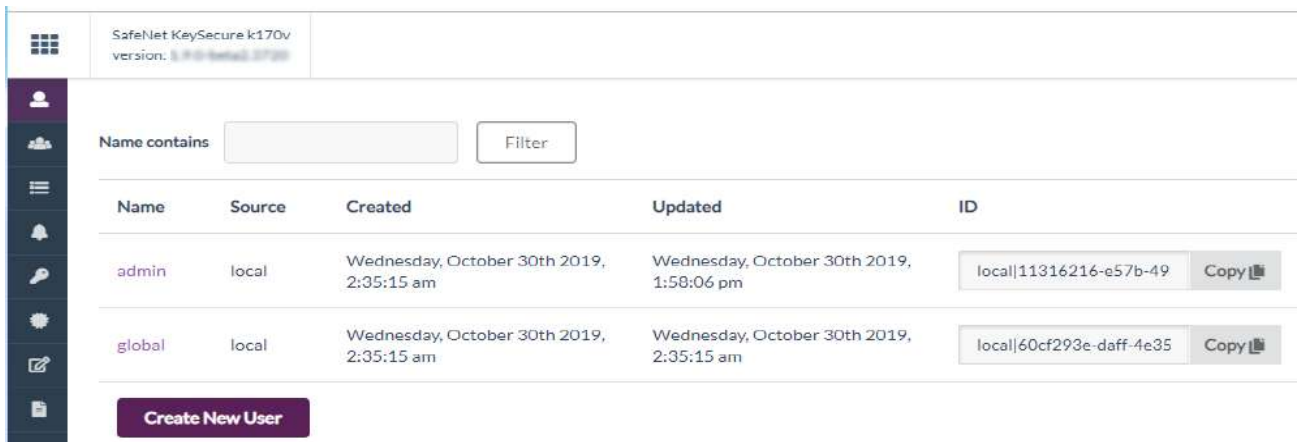
Min number of lower cases: 1

Min number of digits: 1

Min number of other characters: 1

A new Login screen appears.

**21.** Using your new password, log in again. The **SafeNet KeySecure k170v** Web Page appears.



Congratulations! You have successfully deployed your SafeNet KeySecure k170v instance.

## Changing the Initial User Password

The username of the initial NextGen KeySecure user is "admin". This user becomes the initial Application Administrator. This Application Administrator's initial password, whether hard-coded or auto-generated, must be changed before using the Next Generation KeySecure. The GUI will prompt for password change on initial login.



**NOTE** To better understand the different user groups within the NextGen KeySecure, refer to the Administrator Guide > Concepts > Groups.

### Using the Hard-coded initial password (default)

By default, the initial password for the initial Application Administrator is hard-coded to 'admin', in lowercase. On initial log in, the Application Administrator is forced to change his password.

### Retrieving an Auto-generated initial password

Optionally, the initial password for the initial Application Administrator can be auto-generated through cloud-init configuration using the following setting:

```
#cloud-config
keysecure:
 initial-user:
 generate-password: true
```

For more information on cloud-init configuration, see ["Using Cloud-init" on page 24](#).

After the initial user's password is auto-generated, it must be retrieved.



## Retrieving the Initial password for the Application Administrator

The auto-generated, initial password for the Application Administrator must be retrieved by the System Administrator "ksadmin" via SSH or via the CLI.

### To retrieve the initial password for the Application Administrator using SSH.

The initial password is written out to the file `/var/keysecure/password/out`. Retrieve it using SSH:

```
$ ssh ksadmin@{addr} cat /var/keysecure/password/out
```

If the ssh key used to launch the instance is not loaded in your ssh-agent, add the key file argument:

```
$ ssh -i {key file} ksadmin@{addr} cat /var/keysecure/password/out
```



**CAUTION!** After retrieving the initial password for the Application Administrator, the Application Administrator must log in and change it immediately. Failure to do so could allow an attacker to compromise the system.

### To retrieve the initial password for the Application Administrator using the CLI:

On Linux-like systems that have the `ssh` command line tool available, **ksctl** can be used to execute the `ssh` command to retrieve the initial password.

If you installed the CLI (see [the Administrator Guide > Operation > Installation](#)), you can use it to acquire the initial password as well:

```
$ ksctl init getpw --url https://<ipaddr> --identityFile <private key> --user admin
```



**CAUTION!** After retrieving the initial password for the Application Administrator, the Application Administrator must log in and change it immediately. Failure to do so could allow an attacker to compromise the system.

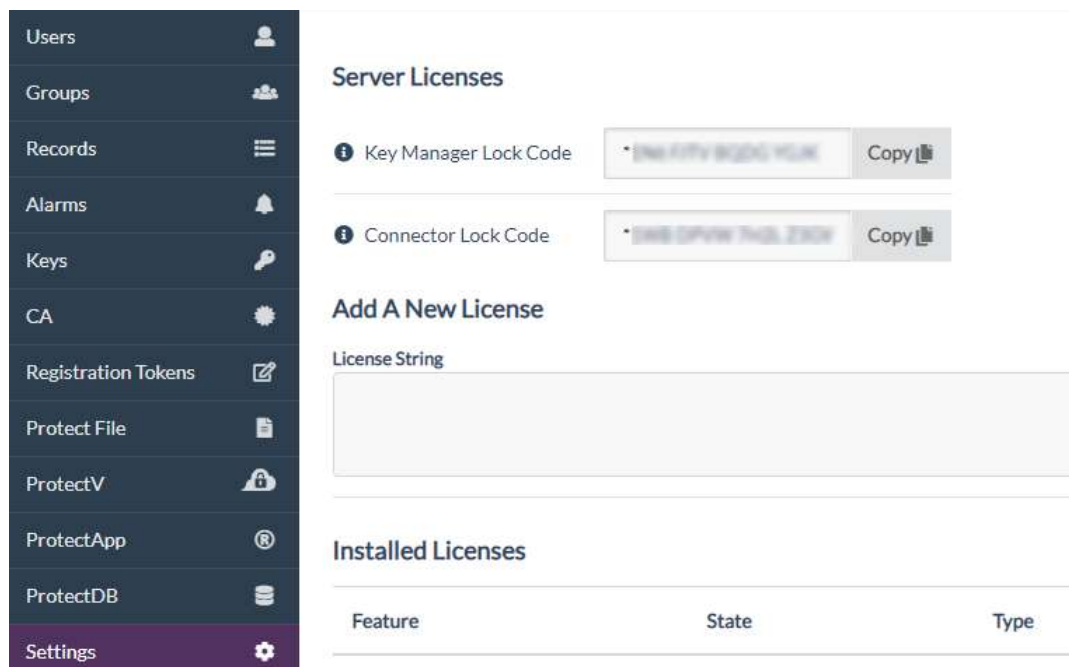
## Licensing

The NextGen KeySecure requires a valid license to create and manage keys. An initial 90-day Trial License is provided for evacuation. 30 days prior to license expiration, the NextGen KeySecure Server will notify you, in the top banner in red, that your license is expiring soon. Before the expiration, a new purchased license must be installed. Contact Gemalto Sales representative for assistance in obtaining a license.

## Lock Codes

Licensing requires a lock code. Each NextGen KeySecure server comes with two lock codes, the Key Manager Lock Code and the Connector Lock Code. These can be accessed via the GUI, KSCTL and API:

To access from the GUI, Select **Settings > Licenses**.



The basic licensing steps are shown here using the GUI and KSCTL:

### To activate a license using the GUI

1. Retrieve the 'Key Manager Lock Code'.  
In the KeySecure GUI, select **Settings > Licenses**.
2. **Copy** the 'Key Manager Lock Code'.
3. To receive a license string, go to the Gemalto Sentinel Cloud EMS Portal at:  
<https://dplicense.gemalto.sentinelcloud.com/ems/customerLogin.html>
4. Past in the 'Key Manager Lock Code' to generate a license string.
5. Copy the License String from the portal.
6. In the KeySecure GUI, activate your product by pasting in the License String in the **License String** field.

### To activate a license using the KSCTL

1. Retrieve the 'Key Manager Lock Code'.  

```
$ ksctl licensing lockdata
```
2. To receive a License String, go to the Gemalto Sentinel Cloud EMS Portal at:  
<https://dplicense.gemalto.sentinelcloud.com/ems/customerLogin.html>
3. Enter the Key Manager Lock Code to generate a license string.
4. Activate your product by entering the License String.  

```
$ ksctl licensing licenses add -l "<license string>"
```

## Connector/Client Licensing

A separate and unique 'Connector Lock Code' is provided by each NextGen KeySecure Server. This lock code is used to license supported SafeNet Connectors/Clients, e.g. SafeNet ProtectFile, as well to activate the KMIP interface.



**NOTE** Unlike the 'Key Manager Lock Code', the 'Connector Lock Code' is cluster-wide. This means that a license applied using the Connector Lock Code replicates across the cluster.

For connector licensing details, refer to Administrator Guide for the specific SafeNet connector you are using.

Refer to the *NextGen KeySecure Release Notes > Compatibility* section for information on supported SafeNet Connectors.

The Connector Lock Code is also used to license the KMIP interface. For details, refer to "KMIP Licensing" in the KMIP Reference Guide.

## Enhanced Networking

---

The SafeNet Virtual KeySecure k170v AMI ships with Elastic Network Adapter (ENA) enabled. ENA provides high-performance networking on supported EC2 instance types. For more information visit <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html>.

# Microsoft Azure Deployment

This chapter describes how to deploy the SafeNet Virtual KeySecure k170v within the Microsoft Azure Cloud Computing Platform.

This chapter contains the following topics:

- > ["Minimum Requirements" below](#)
- > ["Deploying in Azure" below](#)
- > ["Changing the Initial User Password" on page 80](#)
- > ["Licensing" on page 81](#)

## Minimum Requirements

To deploy a SafeNet Virtual KeySecure k170v instance, the system volume requirements must be at least 135 GB in a production environment and 30 GB when evaluating the product.



**NOTE** The system volume holds all data as well as backups.

**Here are the minimal system requirements for Microsoft Azure:**



**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.

System volume: Minimum size: Standard\_A2, 1 volume (135 GB)

Memory: 16 GB

CPUs: 2

NICs: 1

## Deploying in Azure

This section provides the steps for deploying a SafeNet Virtual KeySecure k170v instance in Microsoft Azure. Refer to the Azure documentation for general information on launching a VM in Microsoft Azure.

## Prerequisites

- > Gemalto entitlement email: You receive this email after requesting a Trial License or purchasing a Term or Perpetual License. Contact Gemalto Sales representative for assistance in obtaining a license. This email contains:
  - Entitlement ID
  - Link to the EMS Customer Portal to enter your k170v instance Lock Code and receive your license.
- > If using a Windows client,
  - use PuTTY or similar utility to SSH to your k170v instance as KeySecure Administrator (ksadmin).
  - If needed, use PuTTYgen or similar utility to format the SSH Key Pair.
- > If using a Linux client: use, SSH to login as KeySecure Administrator (ksadmin).

## Tasks

Deploying a k170v instance in Azure consists of these tasks:

1. Launch the SafeNet Virtual KeySecure k170v instance in Azure, alter the launch settings for your requirements, and access the Lock Code for this instance.
2. Link to the EMS Customer Portal to enter your k170v instance Lock Code and receive your license.
3. Apply your license to your k170v instance.

## Steps

### Steps to launch a SafeNet Virtual KeySecure k170v instance and access the Lock Code.

1. Sign in to the Azure portal.
2. Search for **SafeNet KeySecure k170v (BYOL)** on the Marketplace page.
3. Select the **SafeNet KeySecure k170v (BYOL)** image from the Virtual Machines group.



**NOTE** The following steps apply to the Azure recommended 'Resources Manager' deployment model.

4. Click **Create**. The **Basics** screen of the **Create virtual machine** page is displayed and is partially filled-in with default values.

## Create a virtual machine

 Changing Basic options may reset selections you have made. Review all options prior to creating the virtual machine.

Basics Disks Networking Management Guest config 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.

Looking for classic VMs? [Create VM from Azure Marketplace](#)

### PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

\* Subscription ⓘ

\* Resource group ⓘ  [Create new](#)

### INSTANCE DETAILS

\* Virtual machine name ⓘ  ✓

\* Region ⓘ

Availability options ⓘ

\* Image ⓘ  [Browse all images and disks](#)

\* Size ⓘ **Standard A2**  
2 vcpus, 3.5 GB memory  
[Change size](#)

### ADMINISTRATOR ACCOUNT

Authentication type ⓘ ☐ Password ☒ SSH public key

\* Username ⓘ  ✓

\* SSH public key ⓘ  ✓

5. Change the **Subscription** type if desired.
6. Select an existing **Resource group** or enter the name for a new one.
7. Specify the **Region** of an Azure Datacenter. For example, **East US**.
8. Enter a **Virtual machine name**, which is the hostname for the virtual machine your are creating, for example, "mycompany-keysecure".
9. Select the **Size** for the VM. For example, select the k170v recommended minimum size **Standard A2**.
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.

11. For the **Username**, enter "ksadmin", the default name for k170v System Administrator.



**NOTE** You MUST use the name “ksadmin” for this initial user.

12. In the **SSH Public Key** field, paste in your SSH Public Key.



**NOTE** If you do not have an SSH Public Key, you can generate one using 'PuTTYgen' or similar utility.



**\*\*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. Select **Next: Disks** >. The **Disks** screen is displayed.

Home > SafeNet KeySecure k170v (BYOL) > Create a virtual machine

### Create a virtual machine

---

Basics **Disks** Networking Management Guest config 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 HDD ▼

Use unmanaged disks ⓘ ☐ Yes ☒ No

**DATA DISKS**

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 |
|-----|------|------------|-----------|--------------|
|-----|------|------------|-----------|--------------|

[Create and attach a new disk](#) [Attach an existing disk](#)

14. Change the **OS disk type** to 'Standard HDD, unless you desire a faster disk.

15. Change the **Use unmanaged disks** settings if desired.

16. Select **Next: Networking** >. The **Networking** screen is displayed.

## Create a virtual machine

Basics Disks Networking Management Guest config Tags Review + create

Configure a new or existing virtual network for your VM as well as how your VM will be accessed on the virtual network. [Learn more](#)

### NETWORK INTERFACE

When creating a virtual machine, a network interface will be created for you.

|                                    |                                                                                                                             |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| * Virtual network ⓘ                | (new) k170v-test-vnet<br><a href="#">Create new</a>                                                                         |
| * Subnet ⓘ                         | default                                                                                                                     |
| Public IP ⓘ                        | (new) mycompany-keysecure-ip<br><a href="#">Create new</a>                                                                  |
| Network security group             | <input type="radio"/> Basic <input checked="" type="radio"/> Advanced                                                       |
|                                    |  This VM image has preconfigured NSG rules |
| * Configure network security group | (new) mycompany-keysecure-nsg<br><a href="#">Create new</a>                                                                 |
| Accelerated networking ⓘ           | <input type="radio"/> On <input checked="" type="radio"/> Off                                                               |

The selected image does not support accelerated networking.

Note that **Virtual Network**, **Public IP** and **Configure network security group** fields are populated with default values for this VM. Create new values only if needed. For a list of k170v security groups/ports, refer to ["Network Access - Security Group Rules" on page 15](#)

**17.** Select **Next: Management >**. The Management screen is displayed.



## Create a virtual machine

Basics Disks Networking Management Guest config Tags Review + create

Configure monitoring and management options for your VM.

### MONITORING

Boot diagnostics ⓘ ☒ On ☐ Off

OS guest diagnostics ⓘ ☐ On ☒ Off

\* Diagnostics storage account ⓘ

### IDENTITY

Managed service identity ⓘ ☐ On ☒ Off

### AUTO-SHUTDOWN

Enable auto-shutdown ⓘ ☐ On ☒ Off

Note that the **Diagnostic storage account** field is populated with a default value. Change it only if needed.

Select **Next: Guest config >**. The **Guest config** screen is displayed.

## Create a virtual machine

Basics Disks Networking Management Guest config Tags Review + create

Add additional configuration, agents, scripts or applications via virtual machine extensions or cloud-init.

### EXTENSIONS

Extensions provide post-deployment configuration and automation.

Extensions ⓘ [Select an extension to install](#)

### CLOUD INIT

Cloud init is a widely used approach to customize a Linux VM as it boots for the first time. You can use cloud-init to install packages and write files or to configure users and security. [Learn more](#)



The selected image does not support cloud init.



**NOTE** The NextGen KeySecure supports **Extensions** for using cloud-init for post-deployment configuration. However the selected VM **Size** in this example does not allow for this option. If cloud-init configuration is desired, return to the **Basics** screen and select a VM **Size** that supports Extensions.

18. Select **Next: Tags >**. The **Tags** screen is displayed.

19. [Home](#) > [SafeNet KeySecure k170v \(BYOL\)](#) > [Create a virtual machine](#)

### Create a virtual machine

[Basics](#) [Disks](#) [Networking](#) [Management](#) [Guest config](#) **[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](#)

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

| <input type="checkbox"/> | KEY                                 | VALUE                                     | RESOURCE TYPE                                            |
|--------------------------|-------------------------------------|-------------------------------------------|----------------------------------------------------------|
| <input type="checkbox"/> | <input type="text" value="Name"/> ✓ | <input type="text" value="Operations"/> ✓ | <input type="text" value="All resources to be created"/> |
|                          | <input type="text"/>                | <input type="text"/>                      | <input type="text" value="All resources to be created"/> |

20. Enter any desired tags.

21. This is the final screen. Select **Next: Review + 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:

[Home](#) > [SafeNet KeySecure k170v \(BYOL\)](#) > [Create a virtual machine](#)

### Create a virtual machine

Validation failed. Click here to view details.

[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 Guest config Tags **Review + create**

### PRODUCT DETAILS

SafeNet Virtual KeySecure k170v

by Gemalto

[Terms of use](#) | [Privacy policy](#)

Standard A2

by Microsoft

[Terms of use](#) | [Privacy policy](#)

Not covered by credits ⓘ

0.0000 USD/hr

Subscription credits apply ⓘ

0.1200 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         | Free Trial                            |
| Resource group       | k170v-test                            |
| Virtual machine name | mycompany-keysecure                   |
| Region               | East US                               |
| Availability options | No infrastructure redundancy required |
| Authentication type  | SSH public key                        |
| Username             | ksadmin                               |

### DISKS

|                     |              |
|---------------------|--------------|
| OS disk type        | Standard HDD |
| Use unmanaged disks | No           |

### NETWORKING

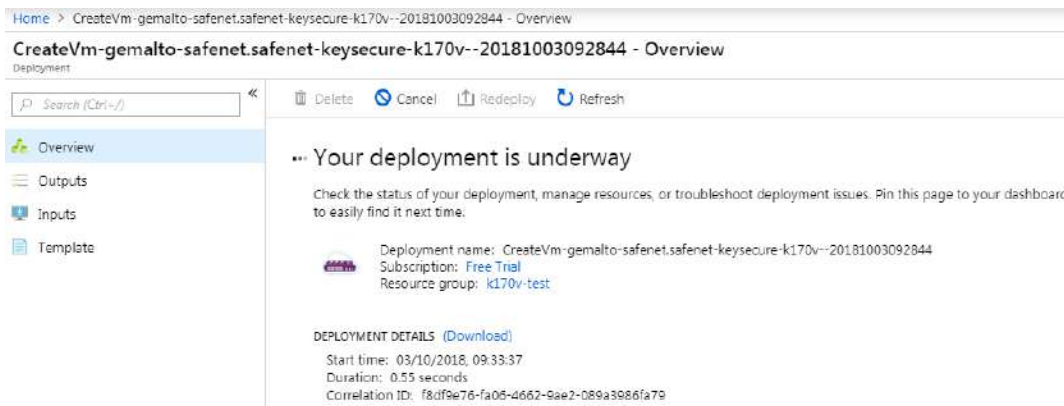
Create

Previous

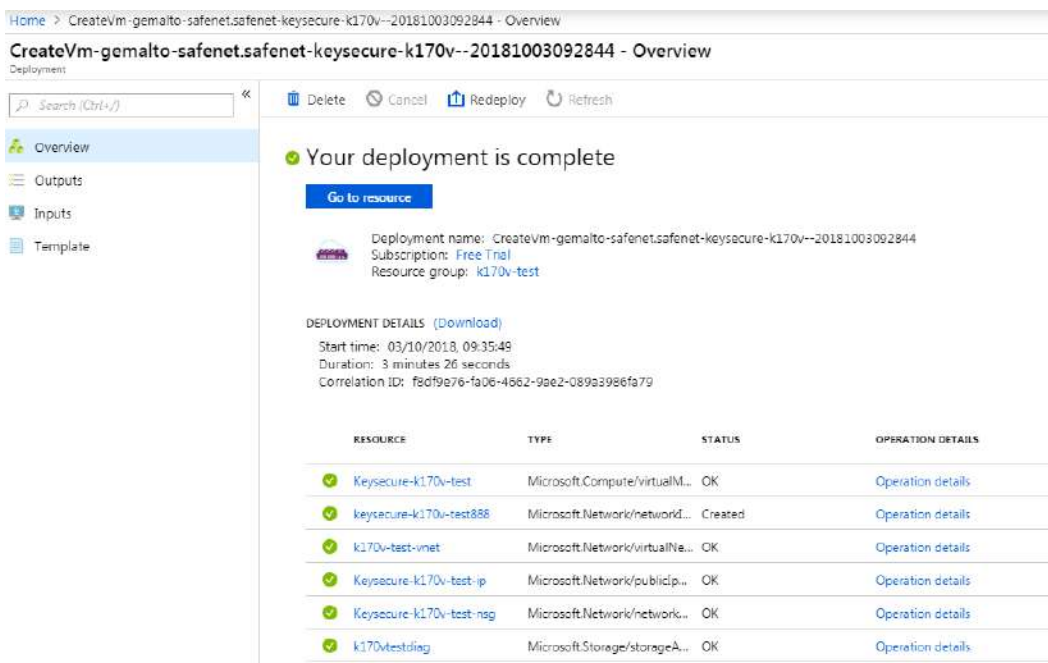
Next

[Download a template for automation](#)

**22.** Select **Create** to begin deployment of this VM. This screen is displayed indicating that deployment is in process.



When deployment completes, this screen is displayed, providing access to all resources supporting the new VM.



## 23. Connect to the SafeNet KeySecure k170v Web Page.

- Select the resource with the IP address, in this example: **Keysecure-k170v-test-ip**.

Home > CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844 - Overview > Keysecure-k170v-test-ip

**Keysecure-k170v-test-ip**  
Public IP address

Search (Ctrl+J)

Overview

Activity log

Access control (IAM)

Tags

Settings

Configuration

Properties

Associate X Dissociate → Move Delete

Essentials ^

|                                      |                         |
|--------------------------------------|-------------------------|
| Resource group (change)              | SKU                     |
| k170v-test                           | Basic                   |
| Location                             | IP address              |
| East US                              | 40.117.142.62           |
| Subscription name (change)           | DNS name                |
| Free Trial                           | -                       |
| Subscription ID                      | Associated to           |
| aabe3632-30f1-4b08-a8a5-fbb489c5d51f | keysecure-k170v-test888 |
|                                      | Virtual machine         |
|                                      | Keysecure-k170v-test    |

b. Browse to this IP address, in this example enter **https://40.117.142.62**). The **Log In** screen appears.

**gemalto**  
a Thales company

**Username**

username or connection/username

**Password**

password

**Log In**

By clicking Log In, you are accepting [this EULA](#)

**SafeNet KeySecure k170v**  
version 1.10.0.0

[API & CLI Documentation](#)

[Product Documentation](#)

24. Log in using the initial default credentials: Username = admin, Password = admin

The following error is displayed:

**Error**

- Password change required
- 23 : NCERRPasswordChangeRequired: Password change required

The **Error** displayed is normal and simply requires you to immediately change the default Password.

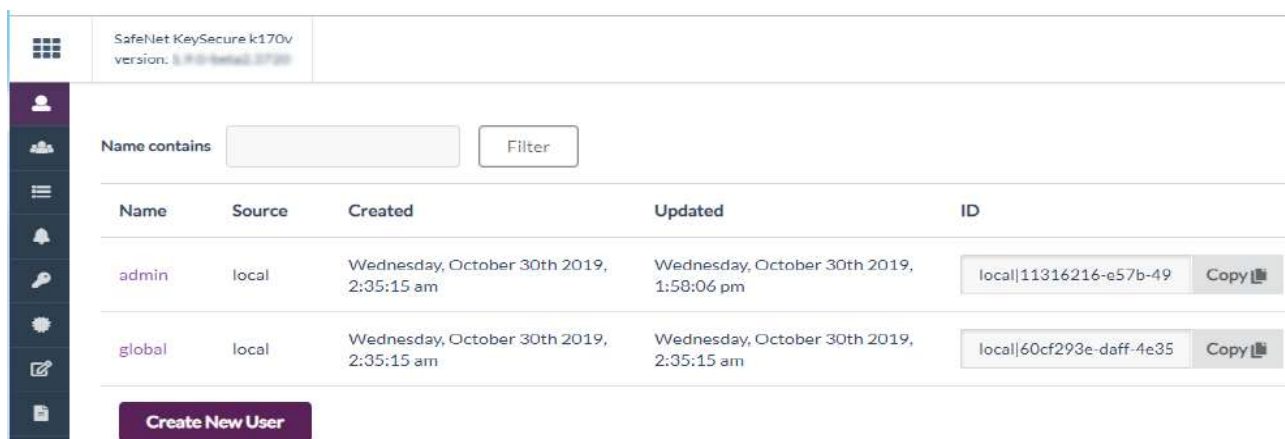
25. Enter a new password using this default Password Policy:

Min length: 8

Max length: 30  
Min number of upper cases: 1  
Min number of lower cases: 1  
Min number of digits: 1  
Min number of other characters: 1

A new Login screen appears.

**26.** Using your new password, log in again. The **SafeNet KeySecure k170v** Web Page appears.



The screenshot shows the SafeNet KeySecure k170v Web Page. At the top, there is a header bar with the product name and version. Below this is a sidebar with navigation icons. The main content area features a search bar labeled 'Name contains' with a 'Filter' button. Below the search bar is a table with columns: Name, Source, Created, Updated, and ID. The table contains two rows: one for 'admin' and one for 'global'. Each row has a 'Copy' button next to the ID. At the bottom of the table is a 'Create New User' button.

| Name   | Source | Created                                  | Updated                                  | ID                       |
|--------|--------|------------------------------------------|------------------------------------------|--------------------------|
| admin  | local  | Wednesday, October 30th 2019, 2:35:15 am | Wednesday, October 30th 2019, 1:58:06 pm | local 11316216-e57b-49   |
| global | local  | Wednesday, October 30th 2019, 2:35:15 am | Wednesday, October 30th 2019, 2:35:15 am | local 60cf293e-daff-4e35 |

Congratulations! You have successfully deployed a k170v instance.

## Changing the Initial User Password

The username of the initial NextGen KeySecure user is "admin". This user becomes the initial Application Administrator. This Application Administrator's initial password, whether hard-coded or auto-generated, must be changed before using the Next Generation KeySecure. The GUI will prompt for password change on initial login.



**NOTE** To better understand the different user groups within the NextGen KeySecure, refer to the Administrator Guide > Concepts > Groups.

### Using the Hard-coded initial password (default)

By default, the initial password for the initial Application Administrator is hard-coded to 'admin', in lowercase. On initial log in, the Application Administrator is forced to change his password.

### Retrieving an Auto-generated initial password

Optionally, the initial password for the initial Application Administrator can be auto-generated through cloud-init configuration using the following setting:

```
#cloud-config
keysecure:
 initial-user:
 generate-password: true
```

For more information on cloud-init configuration, see ["Using Cloud-init" on page 24](#).

After the initial user's password is auto-generated, it must be retrieved.

## Retrieving the Initial password for the Application Administrator

The auto-generated, initial password for the Application Administrator must be retrieved by the System Administrator "ksadmin" via SSH or via the CLI.

### To retrieve the initial password for the Application Administrator using SSH.

The initial password is written out to the file `/var/keysecure/password/out`. Retrieve it using SSH:

```
$ ssh ksadmin@{addr} cat /var/keysecure/password/out
```

If the ssh key used to launch the instance is not loaded in your ssh-agent, add the key file argument:

```
$ ssh -i {key file} ksadmin@{addr} cat /var/keysecure/password/out
```



**CAUTION!** After retrieving the initial password for the Application Administrator, the Application Administrator must log in and change it immediately. Failure to do so could allow an attacker to compromise the system.

### To retrieve the initial password for the Application Administrator using the CLI:

On Linux-like systems that have the ``ssh`` command line tool available, **ksctl** can be used to execute the ``ssh`` command to retrieve the initial password.

If you installed the CLI (see [the Administrator Guide > Operation > Installation](#)), you can use it to acquire the initial password as well:

```
$ ksctl init getpw --url https://<ipaddr> --identityFile <private key> --user admin
```



**CAUTION!** After retrieving the initial password for the Application Administrator, the Application Administrator must log in and change it immediately. Failure to do so could allow an attacker to compromise the system.

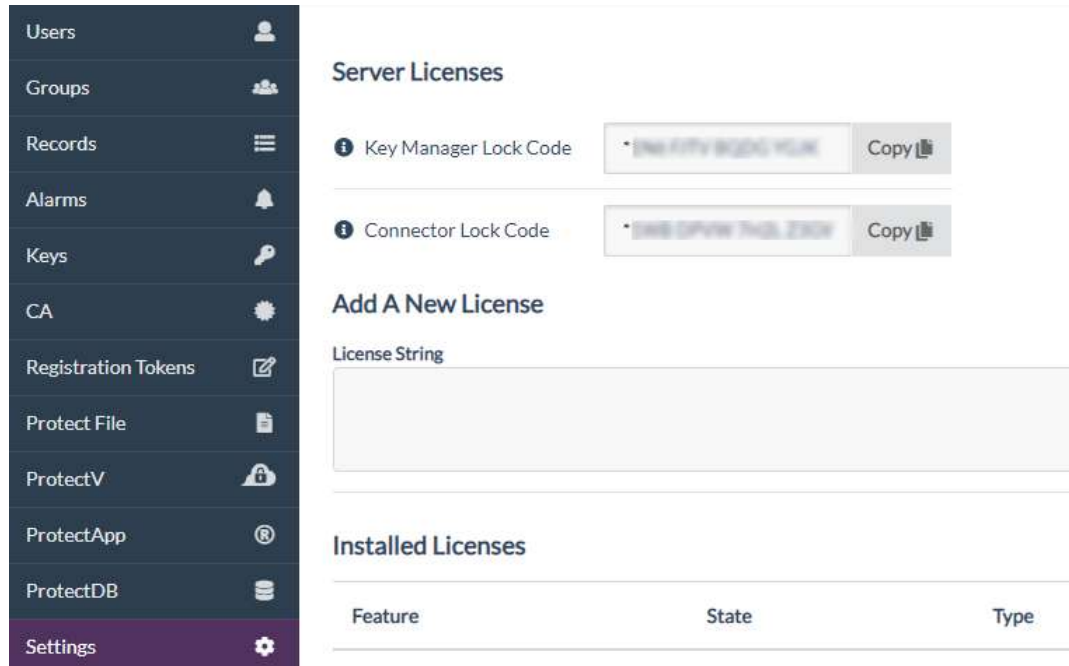
## Licensing

The NextGen KeySecure requires a valid license to create and manage keys. An initial 90-day Trial License is provided for evaluation. 30 days prior to license expiration, the NextGen KeySecure Server will notify you, in the top banner in red, that your license is expiring soon. Before the expiration, a new purchased license must be installed. Contact Gemalto Sales representative for assistance in obtaining a license.

## Lock Codes

Licensing requires a lock code. Each NextGen KeySecure server comes with two lock codes, the Key Manager Lock Code and the Connector Lock Code. These can be accessed via the GUI, KSCTL and API:

To access from the GUI, Select **Settings > Licenses**.



The basic licensing steps are shown here using the GUI and KSCTL:

### To activate a license using the GUI

1. Retrieve the 'Key Manager Lock Code'.  
In the KeySecure GUI, select **Settings > Licenses**.
2. **Copy** the 'Key Manager Lock Code'.
3. To receive a license string, go to the Gemalto Sentinel Cloud EMS Portal at:  
<https://dplicense.gemalto.sentinelcloud.com/ems/customerLogin.html>
4. Past in the 'Key Manager Lock Code' to generate a license string.
5. Copy the License String from the portal.
6. In the KeySecure GUI, activate your product by pasting in the License String in the **License String** field.

### To activate a license using the KSCTL

1. Retrieve the 'Key Manager Lock Code'.  

```
$ ksctl licensing lockdata
```
2. To receive a License String, go to the Gemalto Sentinel Cloud EMS Portal at:  
<https://dplicense.gemalto.sentinelcloud.com/ems/customerLogin.html>



3. Enter the Key Manager Lock Code to generate a license string.
4. Activate your product by entering the License String.

```
$ ksctl licensing licenses add -l "<license string>"
```

## Connector/Client Licensing

A separate and unique 'Connector Lock Code' is provided by each NextGen KeySecure Server. This lock code is used to license supported SafeNet Connectors/Clients, e.g. SafeNet ProtectFile, as well to activate the KMIP interface.



**NOTE** Unlike the 'Key Manager Lock Code', the 'Connector Lock Code' is cluster-wide. This means that a license applied using the Connector Lock Code replicates across the cluster.

For connector licensing details, refer to Administrator Guide for the specific SafeNet connector you are using.

Refer to the *NextGen KeySecure Release Notes > Compatibility* section for information on supported SafeNet Connectors.

The Connector Lock Code is also used to license the KMIP interface. For details, refer to "KMIP Licensing" in the KMIP Reference Guide.

# Google Cloud Deployment

This chapter describes how to deploy a SafeNet Virtual KeySecure k170v instance in Google Cloud.

This chapter contains the following topics:

- > ["Minimum Requirements" below](#)
- > ["Deploying in Google Cloud" below](#)
- > ["Changing the Initial User Password" on page 90](#)
- > ["Licensing" on page 91](#)

## Minimum Requirements

To deploy a SafeNet Virtual KeySecure k170v instance, the system volume requirements must be at least 100 GB in a production environment and 30 GB when evaluating the product.



**NOTE** The system volume holds all data as well as backups.

### Here are the minimal system requirements for Google Cloud:



**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.

Boot Disk: 1 volume, minimum size: (30 GB for evaluation purposes, 135 GB for production)

Memory: 16 GB

vCPUs: 2

NICs: 1

## Deploying in Google Cloud

This section provides the steps for deploying a SafeNet Virtual KeySecure k170v instance in Google Cloud. Refer to the Google Cloud Platform documentation for general information on launching a VM in Google Cloud.

## Prerequisites

- > Gemalto entitlement email: You receive this email after requesting a Trial License or purchasing a Term or Perpetual License. Contact Gemalto Sales representative for assistance in obtaining a license. This email contains:
  - Entitlement ID
  - Link to the EMS Customer Portal to enter your k170v instance Lock Code and receive your license.



**NOTE** If you have not received your Gemalto email with Entitlement ID, you can still launch a k170v instance and use the GUI to browse the Web page. However, you will not be able to create or manage keys until you register the instance's Lock Code and apply your license.

- > <KSShort> Image for Google Cloud.
- > If using a Windows client,
  - use PuTTY or similar utility to SSH to your k170v instance as KeySecure Administrator (ksadmin).
  - If needed, use PuTTYgen or similar utility to format the SSH Key Pair.
- > If using a Linux client: use, SSH to login as KeySecure Administrator (ksadmin).

## Tasks

Deploying a SafeNet Virtual KeySecure k170v instance in Google Cloud consists of these tasks:

1. Launch the k170v instance in Google Cloud and access the Lock Code for the instance.
2. Link to the EMS Customer Portal to enter your k170v instance Lock Code and receive your license.
3. Apply your license to your k170v instance.

## Steps

### Task 1: To launch a SafeNet Virtual KeySecure k170v instance and access the Lock Code:

1. Using your gmail address, sign in to the Google Cloud Platform: <https://console.cloud.google.com/>
2. In the **Google Cloud Platform** top banner, select a project or create a new one.



To create a New Project:

- a. Select **New Project**.

- b.** Enter a **Project Name** and **Project ID**.
- c.** Enter a **Location** if you already have a parent organization you want to use.
- d.** Select **Create**.

The project's **DASHBOARD** appears.

- 3.** In the left pane, under **COMPUTE**, select **Compute Engine > Images**.
- 4.** Search for "k170v" images. The **Images** page appears.
- 5.** On the **Images**, select the latest image in the table (recommended).

- 6.**

- 7.** At the top of the **Images** page, select **CREATE INSTANCE**.

The **Create an instance** form appears.

**NOTE** Not all fields must be filled-in or modified. The following steps address the fields needed for most deployments. Many of the fields can be modified after deployment.

- 8.** To create a k170v instance, modify these fields.
  - a.** Enter a **Name** for the instance, for example, "k170v-test".
  - b.** Select the **Region** and **Zone** closest to your location.
  - c.** For **Machine type**, enter the minimum number of vCPUs. The recommended minimum is 1.  
Then select **Customize** to select a memory size. The recommended minimum is 16 GB.



**NOTE** The **Boot disk** fields are already populated by the selected SafeNet Virtual KeySecure k170v image.

- d. In the **Firewall** field, you must check both boxes **Allow HTTP traffic** and **Allow HTTPS traffic**.

Name 

k170v-test

Region  Zone 

us-east1 (South Carolina) us-east1-b

Machine type  
Customize to select cores, memory and GPUs.

1 vCPU 3.75 GB memory [Customize](#)

Container 

☐ Deploy a container image to this VM instance. [Learn more](#)

Boot disk 

 New 30 GB standard persistent disk  
Image  
k170v-2369-20181008172807 [Change](#)

Identity and API access 

Service account 

Compute Engine default service account

Access scopes 

☒ Allow default access  
☐ Allow full access to all Cloud APIs  
☐ Set access for each API

Firewall 

Add tags and firewall rules to allow specific network traffic from the Internet

☒ Allow HTTP traffic  
☒ Allow HTTPS traffic

[Management, security, disks, networking, sole tenancy](#)

9. At the bottom of the form, select the **Security** tab. The **Security** form appears:

10. Enter your SSH public key in the box provided.

If a valid public key is entered, Google Cloud will place the key name to the left of the box.



**\*\*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.

Management **Security** Disks Networking Sole Tenancy

**Shielded VM** ?  
 Select a shielded image to use shielded VM features.  
 Turn on all settings for the most secure configuration.

☐ Turn on Secure Boot ?  
☐ Turn on vTPM ?  
☐ Turn on Integrity Monitoring ?

**SSH Keys**  
 These keys allow access only to this instance, unlike [project-wide SSH keys](#) [Learn more](#)

☐ Block project-wide SSH keys  
 When checked, project-wide SSH keys cannot access this instance [Learn more](#)

rsa-key-20180921

bVXYIcvWKorn/8Kf1s2vfX5GcZPrBEQE2tfLnX7pI  
 Y7w8YO3dwn3jP6W1Dbf6KgKV8JYrRsBDJ4EcQrEfy  
 EGHFLt8FUTuAn69oQdB1DBPEsZFhoWyYMPw71jCHM  
 weXH/Y+DwoQC3D9Ka11vnymRmwbxEGq7FRYD52G13  
 62ShkdSQ3RODuUw+HYXVSC0rusyTnKFT+/GU2s6Vz  
 JRHVmnHSIvqsb45vPu0pmRdPD8n4+21m6F0+uI9mB  
 cxUhw1gQ== rsa-key-20180921

+ Add item

Less

You will be billed for this instance. [Learn more](#)

**Create** Cancel

11. Select **Create** to create the k170v VM.

Your new VM is created and appears in the **VM instances** table.

VM instances [+ CREATE INSTANCE](#) [IMPORT VM](#) [REFRESH](#) [START](#) [STOP](#) [RESET](#)

Filter VM instances Columns

| <input type="checkbox"/> Name                       | Zone       | Creation time            | Internal IP       | External IP                   |
|-----------------------------------------------------|------------|--------------------------|-------------------|-------------------------------|
| <input type="checkbox"/> <a href="#">k170v-test</a> | us-east1-b | Oct 11, 2018, 4:09:03 PM | 10.142.0.2 (nic0) | <a href="#">35.237.135.34</a> |

12. Click on the **External IP** for the newly created VM.

A new browser tab opens to this address and the **SafeNet KeySecure k170v Web Page Log In** screen appears.



Username

Password

Log In

By clicking Log In, you are accepting [this EULA](#)

SafeNet KeySecure k170v

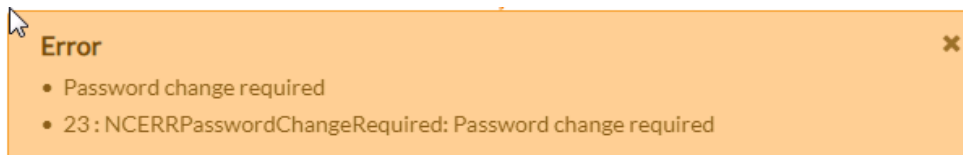
version 1.5.0-beta4.2369

[API & CLI Documentation](#)

[Product Documentation](#)

**13.** Log in using the initial default credentials for the "ksadmin": Username = admin, Password = admin

The following error is displayed:



The **Error** displayed is normal and simply requires you to immediately change the default Password.

**14.** Enter a new password using this default Password Policy:

Min length: 8

Max length: 30

Min number of upper cases: 1

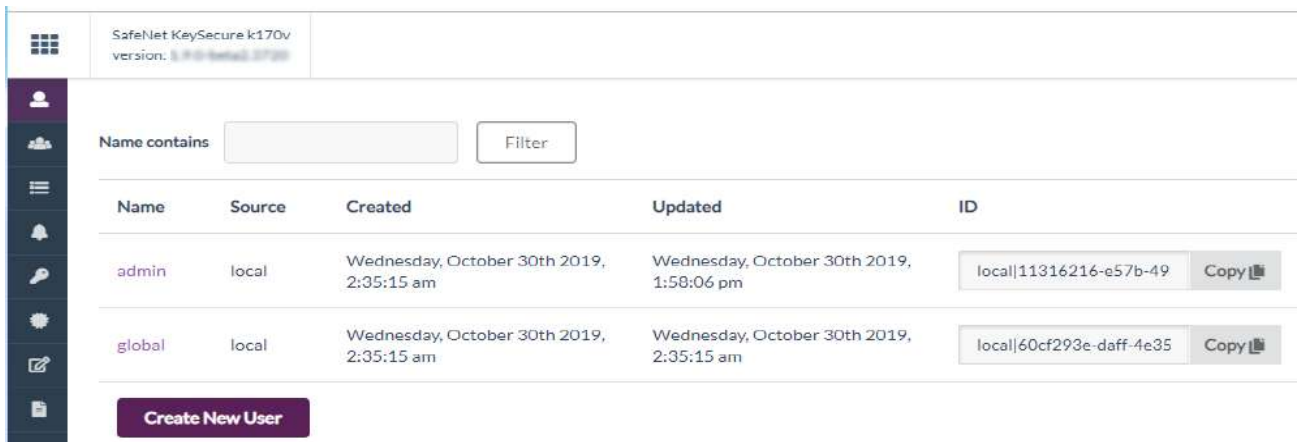
Min number of lower cases: 1

Min number of digits: 1

Min number of other characters: 1

A new Login screen appears.

**15.** Using your new password, log in again. The **SafeNet KeySecure k170v** Web Page appears.



Congratulations! You have successfully deployed a SafeNet Virtual KeySecure k170v instance.

## Changing the Initial User Password

The username of the initial NextGen KeySecure user is "admin". This user becomes the initial Application Administrator. This Application Administrator's initial password, whether hard-coded or auto-generated, must be changed before using the Next Generation KeySecure. The GUI will prompt for password change on initial login.



**NOTE** To better understand the different user groups within the NextGen KeySecure, refer to the Administrator Guide > Concepts > Groups.

### Using the Hard-coded initial password (default)

By default, the initial password for the initial Application Administrator is hard-coded to 'admin', in lowercase. On initial log in, the Application Administrator is forced to change his password.

### Retrieving an Auto-generated initial password

Optionally, the initial password for the initial Application Administrator can be auto-generated through cloud-init configuration using the following setting:

```
#cloud-config
keysecure:
 initial-user:
 generate-password: true
```

For more information on cloud-init configuration, see ["Using Cloud-init" on page 24](#).

After the initial user's password is auto-generated, it must be retrieved.



## Retrieving the Initial password for the Application Administrator

The auto-generated, initial password for the Application Administrator must be retrieved by the System Administrator "ksadmin" via SSH or via the CLI.

### To retrieve the initial password for the Application Administrator using SSH.

The initial password is written out to the file `/var/keysecure/password/out`. Retrieve it using SSH:

```
$ ssh ksadmin@{addr} cat /var/keysecure/password/out
```

If the ssh key used to launch the instance is not loaded in your ssh-agent, add the key file argument:

```
$ ssh -i {key file} ksadmin@{addr} cat /var/keysecure/password/out
```



**CAUTION!** After retrieving the initial password for the Application Administrator, the Application Administrator must log in and change it immediately. Failure to do so could allow an attacker to compromise the system.

### To retrieve the initial password for the Application Administrator using the CLI:

On Linux-like systems that have the `ssh` command line tool available, **ksctl** can be used to execute the `ssh` command to retrieve the initial password.

If you installed the CLI (see [the Administrator Guide > Operation > Installation](#)), you can use it to acquire the initial password as well:

```
$ ksctl init getpw --url https://<ipaddr> --identityFile <private key> --user admin
```



**CAUTION!** After retrieving the initial password for the Application Administrator, the Application Administrator must log in and change it immediately. Failure to do so could allow an attacker to compromise the system.

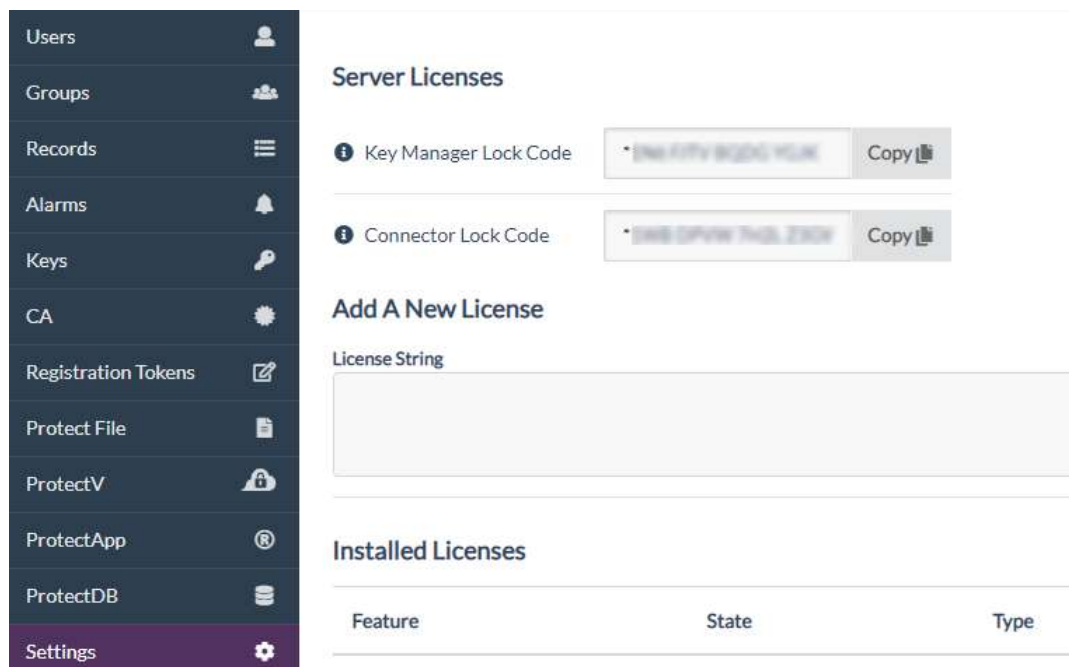
## Licensing

The NextGen KeySecure requires a valid license to create and manage keys. An initial 90-day Trial License is provided for evacuation. 30 days prior to license expiration, the NextGen KeySecure Server will notify you, in the top banner in red, that your license is expiring soon. Before the expiration, a new purchased license must be installed. Contact Gemalto Sales representative for assistance in obtaining a license.

## Lock Codes

Licensing requires a lock code. Each NextGen KeySecure server comes with two lock codes, the Key Manager Lock Code and the Connector Lock Code. These can be accessed via the GUI, KSCTL and API:

To access from the GUI, Select **Settings > Licenses**.



The basic licensing steps are shown here using the GUI and KSCTL:

### To activate a license using the GUI

1. Retrieve the 'Key Manager Lock Code'.  
In the KeySecure GUI, select **Settings > Licenses**.
2. **Copy** the 'Key Manager Lock Code'.
3. To receive a license string, go to the Gemalto Sentinel Cloud EMS Portal at:  
<https://dplicense.gemalto.sentinelcloud.com/ems/customerLogin.html>
4. Past in the 'Key Manager Lock Code' to generate a license string.
5. Copy the License String from the portal.
6. In the KeySecure GUI, activate your product by pasting in the License String in the **License String** field.

### To activate a license using the KSCTL

1. Retrieve the 'Key Manager Lock Code'.  

```
$ ksctl licensing lockdata
```
2. To receive a License String, go to the Gemalto Sentinel Cloud EMS Portal at:  
<https://dplicense.gemalto.sentinelcloud.com/ems/customerLogin.html>
3. Enter the Key Manager Lock Code to generate a license string.
4. Activate your product by entering the License String.  

```
$ ksctl licensing licenses add -l "<license string>"
```

## Connector/Client Licensing

A separate and unique 'Connector Lock Code' is provided by each NextGen KeySecure Server. This lock code is used to license supported SafeNet Connectors/Clients, e.g. SafeNet ProtectFile, as well to activate the KMIP interface.



**NOTE** Unlike the 'Key Manager Lock Code', the 'Connector Lock Code' is cluster-wide. This means that a license applied using the Connector Lock Code replicates across the cluster.

For connector licensing details, refer to Administrator Guide for the specific SafeNet connector you are using.

Refer to the *NextGen KeySecure Release Notes > Compatibility* section for information on supported SafeNet Connectors.

The Connector Lock Code is also used to license the KMIP interface. For details, refer to "KMIP Licensing" in the KMIP Reference Guide.

# Private Cloud Deployment

This chapter describes how to deploy a SafeNet Virtual KeySecure instance in a private cloud. Support is currently provided for Oracle VMware vSphere, Microsoft Hyper-V and OpenStack.

This chapter contains the following topics:

- > ["Minimum Requirements" below](#)
- > ["Deploying on VMware vSphere" below](#)
- > ["Deploying in Hyper-V" on page 99](#)
- > ["Replacing SSH key before system can boot" on page 107](#)
- > ["Changing the Password for the Initial User" on page 120](#)
- > ["Licensing" on page 123](#)

## Minimum Requirements

To deploy a SafeNet Virtual KeySecure k170v instance, the system volume requirements must be at least 100 GB in a production environment and 30 GB when evaluating the product.



**NOTE** The system volume holds all data as well as backups.

**Here are the minimal system requirements for VMware, Microsoft Hyper-V and OpenStack:**



**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.

System volume: Ubuntu [Linux 64 bit], 1 volume (100 GB)

Memory: 16 GB

vCPUs: 2

NICs: 1 (VMXNET 3)

## Deploying on VMware vSphere

This section guides you through the steps needed to deploy a SafeNet Virtual KeySecure k170v VM on VMware vSphere.

Refer to the *VMware vSphere documentation* for general information on launching a VM.

## Prerequisites:

- > Gemalto entitlement email: You receive this email after requesting purchasing a Term or Perpetual License. Contact Gemalto Sales representative for assistance in obtaining a license. This email contains:
  - Entitlement ID
  - Link to the Gemalto support portal to download the k170v OVA file.
  - Link to the EMS Customer Portal to enter your k170v Lock Code and receive your license.
- > OVA file: If your vSphere Client does not support deployment of OVA files directly, see ["Decompressing an OVA file" on page 97](#).

## Tasks

Deploying a k170v VM consists of these three tasks:

1. Launch the NextGen KeySecure VM and access the Lock Code.
2. Provide the Lock Code to Sentinel EMS to obtain a license.
3. Apply the license to the NextGen KeySecure VM.

## Steps

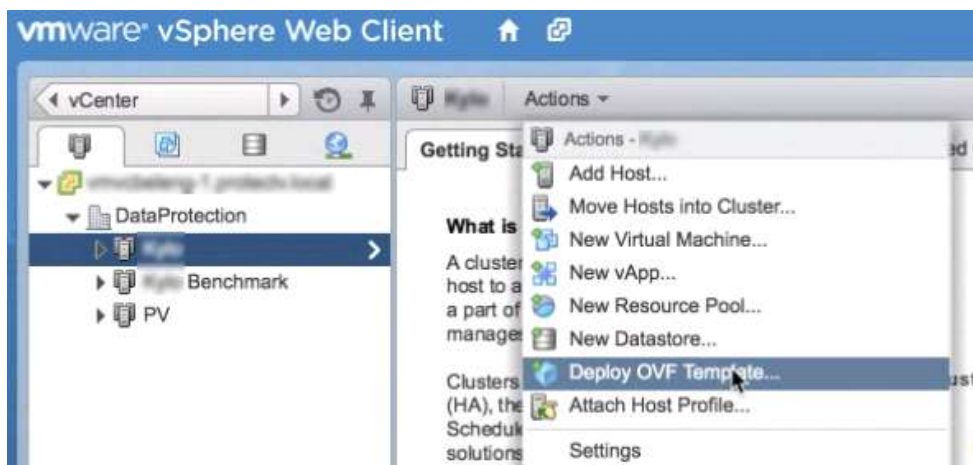
### Steps to launch the SafeNet Virtual KeySecure k170v VM and access the Lock Code:

The k170v supports VMware vSphere/ESXi v5.1 and higher versions. To launch k170v VM on vSphere/ESXi v5.0 or lower versions, you need to change the virtual machine's hardware version from vmx-09 to vmx-08 or vmx-07. Use the VMware OVF Tool to make this change.

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 k170v VM appears under the inventory folder in the left pane.
3. In the left pane, select the newly launched k170v VM.
4. In the middle pane, select the **Summary** tab and click on the **Console Window** at the top-left corner.  
The window opens in your browser.
5. In the **Console Window** select the IP address for your network and access the k170v GUI by browsing to this IP address.

The initial **SafeNet KeySecure k170v** Web Page screen is displayed:



6. The Error displayed is normal and simply requires the default **SSH Public Key** to be replaced. As the initial KeySecure admin (ksadmin) you must paste in your **SSH Public Key** in the box provided and then select **Add**.

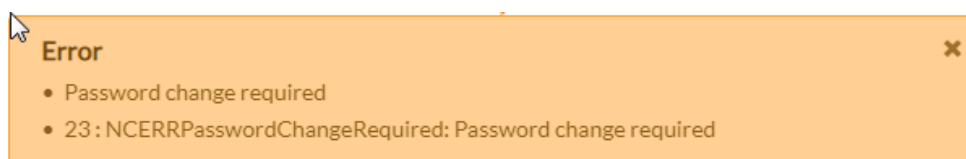
#### NOTE

- > The SSH Public Key must be a 'PEM-formatted RSA key'. You can generate this key using 'PuTTYgen' or similar utility.
- > Save this SSH Public Key at a safe location. You will need this key for future SSH access.

After replacing the default SSH Public Key, the **Log In** screen appears.

7. Log in using the initial default credentials: Username = admin, Password = admin

The following error is displayed:



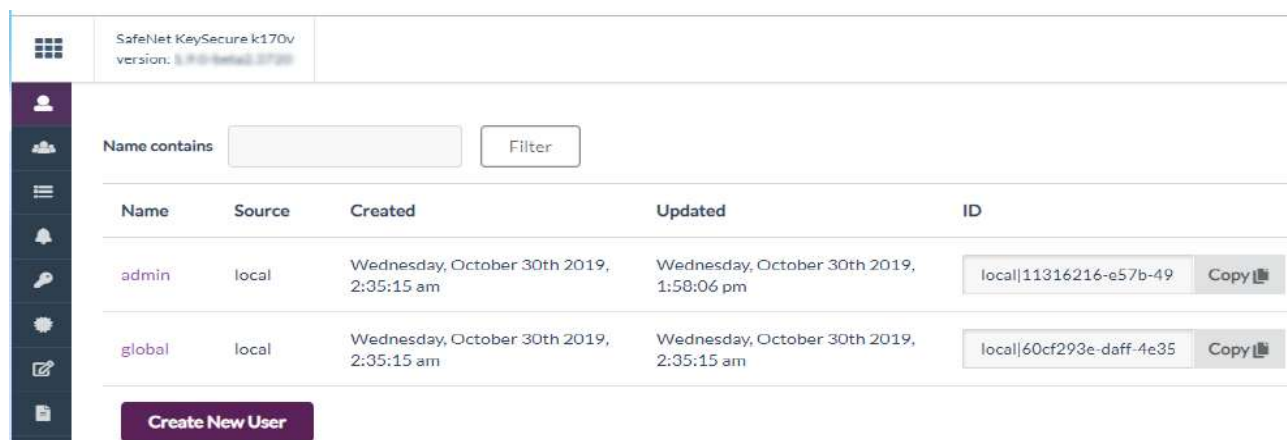
The **Error** displayed is normal and simply requires you must change the default Password.

8. Enter a new password using this default Password Policy:

Min length: 8  
Max length: 30  
Min number of upper cases: 1  
Min number of lower cases: 1  
Min number of digits: 1  
Min number of other characters: 1

A new Login screen appears.

9. Using your new password, log in again. The **SafeNet KeySecure k170v** Web Page appears.



Congratulations! You have successfully deployed your k170v VM.

## Decompressing an OVA file

The k170v package includes an Open Virtual Appliance (OVA) file for launching k170v VM on VMware vSphere. Some versions of the vSphere Client do not support deployment of OVA files directly. In this case it is recommended to extract the .ovf, .vmdk, and .mf files using the VMware OVF Tool. You can then launch k170v VMs using these files.

**Example: To decompress the .ovf file, execute this operation:**

```
$ ovftool.exe --lax <source_OVA_file> <destination_OVF_file>
```

<source\_OVA\_file>: represents the OVA file included in the SafeNet Virtual KeySecure k170v package.

<destination\_OVF\_file>: represents a name for the OVF file.

## Using Cloud-init with VMware vSphere

SafeNet Virtual KeySecure k170v uses Cloud-init to inject configuration information prior to first boot. Cloud-init is a standard configuration mechanism that is supported in vSphere.

Below is an example of setting a static IP address using Cloud-init and vSphere.

### To setup a static IP address using Cloud-init

1. Create your Cloud-init configuration file. This is a text file containing the specific configuration you want to use. Refer to section "[Cloud-init Configuration Options](#)" on page 24 for other available configuration parameters. Here is a simple configuration file for a static IP:

```
#cloud-config
keysecure:
 netcfg:
 iface:
 name: eth0
 type: static
 address: 10.121.107.103
 netmask: 255.255.252.0
 gateway: 10.121.104.1
 dns1: 172.16.2.12
```

2. VMware requires this file to be converted to base64. To convert to base64, use the `openssl` command:

```
$ openssl base64 -in <infile> -out <outfile>
```

The above file converted to base64 is below:

```
I2Nsb3VklWNvbmZpZw0Ka2V5c2VjdXJlOg0KICBuZXRjZmc6DQogICAgZWZhY2U6DQogICAgICBu
YW1lOiBldGgwDQogICAgICB0eXB1OiBzdGF0aWMNCiAgICAgIGFkZHJlc3M6IDEwLjEyMS4xMDcu
MTAzDQogICAgICBuZXRtYXNrOiAyNTUuMjU1LjI1Mi4wDQogICAgICBnYXRld2F5OiAxMC4xMjEu
MTA0LjENCiAgICAgIGRuc2E6IDE3Mi4xNi4yLjEy
```

3. Create a k170v VM in VMware, but do not boot it.
4. Configure the VM with the following:



**NOTE** The vSphere web client should be used since some configuration parameters are not available with the Windows client.

- a. Add an empty CD/DVD drive to the VM, no configuration of it is necessary.
  - b. Under **vApp Options > OVF Settings > OVF Environment Transport**, select ISO Image.
  - c. Under **vApp Options > Properties** create a property with the Key "user-data", and a default value of the base64 encoded file.
5. Boot up your system.





**NOTE** Due to a bug, some versions of SafeNet Virtual KeySecure k170v may not apply the static IP on the initial boot. After a reboot, the IP address will be applied correctly.

## Deploying in Hyper-V

---

The SafeNet Virtual KeySecure k170v Private Cloud Image supports Microsoft Hyper-V in the form of a Virtual Hard Disk (VHDX) image. This image has been tested as a Generation 1 Hyper-V virtual machine.

Refer to the *Microsoft Hyper-V Manager* documentation for general information on launching a VM using an existing disk on Hyper-V.

This section guides you through the steps needed to deploy a SafeNet Virtual KeySecure k170v VM on Microsoft Hyper-V.

### Prerequisites:

- > Gemalto entitlement email: You receive this email after requesting a Trial License or purchasing a Term or Perpetual License. Contact Gemalto Sales representative for assistance in obtaining a license. This email contains:
  - Entitlement ID
  - Link to the Gemalto support portal to download the SafeNet Virtual KeySecure k170v (VHDX) image file.
  - Link to the EMS Customer Portal to enter your k170v instance Lock Code and receive your license.
- > SafeNet Virtual KeySecure k170v Virtual Hard Disk (VHDX) image.

### Tasks

Deploying a SafeNet Virtual KeySecure k170v VM consists of these tasks:

1. Decompress the zipped image file you received from Gemalto Sales.
2. Launch the k170v VM and access the Lock Code.
3. Create a SafeNet Virtual KeySecure k170v VM instance
4. Connect to your new virtual machine

### Steps

---

#### Decompress the zipped image file you received from Gemalto Sales:

1. Locate the image file you received from Gemalto sales; e.g. **k170v-1.x.x.xxxx.vhdx.zip**.
2. Decompress this file by right-clicking on the file and selecting **Extract All....**

This is a large file and will take some time. A single decompressed file will result, which is a Hard Disk Image File; e.g. **k170v-1.x.x.xxxx.vhdx**.

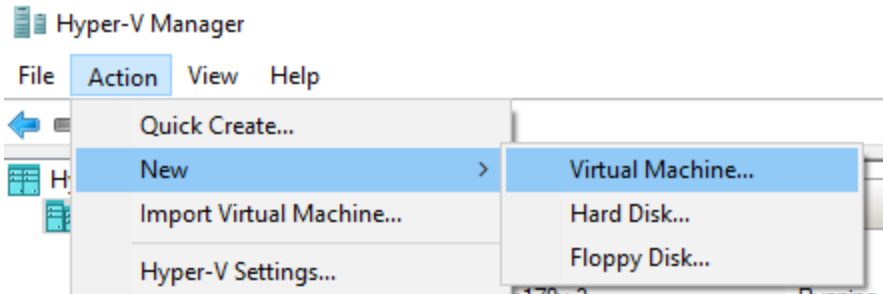
3. Note the location of this file as you will need it below when creating the virtual machine.

## Launch the Hyper-V Manager on your Windows System:

1. If needed, install Hyper-V on your Windows system.  
For example, for Windows 10, refer to: [Install Hyper-V on Windows 10](#)
2. Open Hyper-V Manager on your Windows system.

## Create a SafeNet Virtual KeySecure k170v VM instance:

1. In the Hyper-V Manager UI, select **Action > New > Virtual Machine**.



The 'New Virtual Machine Wizard' begins with the 'Before You Begin' screen.



### Before You Begin

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Before You Begin</b>   | <p>This wizard helps you create a virtual machine. You can use virtual machines in place of physical computers for a variety of uses. You can use this wizard to configure the virtual machine now, and you can change the configuration later using Hyper-V Manager.</p> <p>To create a virtual machine, do one of the following:</p> <ul style="list-style-type: none"><li>• Click Finish to create a virtual machine that is configured with default values.</li><li>• Click Next to create a virtual machine with a custom configuration.</li></ul> |
| Specify Name and Location |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Specify Generation        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Assign Memory             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Configure Networking      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Connect Virtual Hard Disk |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Installation Options      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Summary                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

2. Here you have the choice to either 'create a virtual machine with default values' and then modify them later, or you can 'create a virtual machine with custom configuration'. This procedure uses the option to 'create a virtual machine with custom configuration'. Select **Next**.
3. In the 'Select Name and Location' screen,



## Specify Name and Location

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Before You Begin                 | <p>Choose a name and location for this virtual machine.</p> <p>The name is displayed in Hyper-V Manager. We recommend that you use a name that helps you easily identify this virtual machine, such as the name of the guest operating system or workload.</p> <p>Name: <input type="text" value="k170v_test"/></p> <p>You can create a folder or use an existing folder to store the virtual machine. If you don't select a folder, the virtual machine is stored in the default folder configured for this server.</p> <p><input type="checkbox"/> Store the virtual machine in a different location</p> <p>Location: <input type="text" value="C:\ProgramData\Microsoft\Windows\Hyper-V\"/> <span>Browse...</span></p> <p> If you plan to take checkpoints of this virtual machine, select a location that has enough free space. Checkpoints include virtual machine data and may require a large amount of space.</p> |
| <b>Specify Name and Location</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Specify Generation               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Assign Memory                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Configure Networking             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Connect Virtual Hard Disk        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Installation Options             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Summary                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

- a. Enter a **descriptive name** of the virtual machine you are creating.
  - b. Enter the **folder path** of where the virtual machine will reside, or use the specified default folder path.
  - c. Select **Next**.
4. In the 'Specify Generation' screen, select **Generation 1** and then select **Next**.



## Specify Generation

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Before You Begin          | <p>Choose the generation of this virtual machine.</p> <p><input checked="" type="radio"/> Generation 1<br/>This virtual machine generation supports 32-bit and 64-bit guest operating systems and provides virtual hardware which has been available in all previous versions of Hyper-V.</p> <p><input type="radio"/> Generation 2<br/>This virtual machine generation provides support for newer virtualization features, has UEFI-based firmware, and requires a supported 64-bit guest operating system.</p> <p> Once a virtual machine has been created, you cannot change its generation.</p> |
| Specify Name and Location |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Specify Generation</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Assign Memory             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Configure Networking      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Connect Virtual Hard Disk |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Installation Options      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Summary                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

5. In the 'Assign Memory' screen, select the desired amount of memory for your new virtual machine. Note the Private Cloud minimum requirement for this setting in ["Minimum Requirements" on page 94](#)



## Assign Memory

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Before You Begin          | <p>Specify the amount of memory to allocate to this virtual machine. You can specify an amount from 32 MB through 12582912 MB. To improve performance, specify more than the minimum amount recommended for the operating system.</p> <p>Startup memory: <input type="text" value="4096"/> MB</p> <p><input checked="" type="checkbox"/> Use Dynamic Memory for this virtual machine.</p> <p><b>i</b> When you decide how much memory to assign to a virtual machine, consider how you intend to use the virtual machine and the operating system that it will run.</p> |
| Specify Name and Location |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Specify Generation        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Assign Memory</b>      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Configure Networking      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Connect Virtual Hard Disk |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Installation Options      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Summary                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

- a. Enter the **value**, e.g. 4096.
  - b. If you wish to 'use Dynamic Memory' for this virtual machine, **leave the box checked** (default).
  - c. Select **Next**.
6. In the 'Configure Networking' screen drop down box, choose **Default Switch** and then select **Next**.



## Configure Networking

|                             |                                                                                                                                                                                                                          |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Before You Begin            | <p>Each new virtual machine includes a network adapter. You can configure the network adapter to use a virtual switch, or it can remain disconnected.</p> <p>Connection: <input type="text" value="Default Switch"/></p> |
| Specify Name and Location   |                                                                                                                                                                                                                          |
| Specify Generation          |                                                                                                                                                                                                                          |
| Assign Memory               |                                                                                                                                                                                                                          |
| <b>Configure Networking</b> |                                                                                                                                                                                                                          |
| Connect Virtual Hard Disk   |                                                                                                                                                                                                                          |
| Installation Options        |                                                                                                                                                                                                                          |
| Summary                     |                                                                                                                                                                                                                          |

7. In the 'Connect Virtual Hard Disk' screen,



## Connect Virtual Hard Disk

Before You Begin  
Specify Name and Location  
Specify Generation  
Assign Memory  
Configure Networking  
**Connect Virtual Hard Disk**  
Summary

A virtual machine requires storage so that you can install an operating system. You can specify the storage now or configure it later by modifying the virtual machine's properties.

☐ Create a virtual hard disk  
Use this option to create a VHDX dynamically expanding virtual hard disk.

Name:   
Location:    
Size:  GB (Maximum: 64 TB)

☒ Use an existing virtual hard disk  
Use this option to attach an existing virtual hard disk, either VHD or VHDX format.

Location:

☐ Attach a virtual hard disk later  
Use this option to skip this step now and attach an existing virtual hard disk later.

< Previous   **Next >**   Finish   Cancel

- a. Select the button for **Use an existing virtual hard disk**,
  - b. Enter the location of the 'vhdx' file you decompresses earlier, e.g. **c:/k170v-1.x.x.xxxx.vhdx**
  - c. Select **Next**.
8. In the 'Completing the New Virtual Machine Wizard screen,



## Completing the New Virtual Machine Wizard

Before You Begin

Specify Name and Location

Specify Generation

Assign Memory

Configure Networking

Connect Virtual Hard Disk

**Summary**

You have successfully completed the New Virtual Machine Wizard. You are about to create the following virtual machine.

Description:

|             |                                                        |
|-------------|--------------------------------------------------------|
| Name:       | k170v_test                                             |
| Generation: | Generation 1                                           |
| Memory:     | 4096 MB                                                |
| Network:    | Default Switch                                         |
| Hard Disk:  | C:\k170v-1.8.0.3403.vhdx (VHDX, dynamically expanding) |

To create the virtual machine and close the wizard, click Finish.

< Previous

Next >

**Finish**

Cancel

- Review the Summary description of the virtual machine your are about to create.
- To make a change, select 'Previous' to go back to change setting(s).
- When ready, select **Finish**.

Your new virtual machine is created within a few seconds.

### Connect to you new virtual machine:

- In the Hyper-V Manager, right-click on the new virtual machine and select **Connect** to connect to the new k170v instance.  
The initial screen shows that the VM is turned off.
- Click on **Start**.
- In the **Console Window** of you newly created virtual machine, select the IP address for your network and access the k170v GUI by browsing to this IP address.

The initial **SafeNet KeySecure k170v** Web Page screen is displayed:

**Error**

- The KeySecure will be functional after the default SSH public key for the ksadmin user is replaced

**SSH Public Key**

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDAIIOH4tInSywQu7kBLJ2sdce4BsvaFa5eL6Xh1xgELXArNpjZt
WviKQkz3JD/2TZbyzdLVACvfoVIFx2MT0w83WsuxeUua95Nwr1ehqeaS9Bx1/v50kogEuTzYm1gXyzQQ
bfbPEOhlDR+y7FNvlyvv5++ZUhi2iMIHQEPeZlhoTYHXG1RpR438JXoh/C8ey8aaP6cdZcTudQdsdDP3H
GWXijsP/Np0w4yeBthoy5NFuPpmS0Uvd0k2uCYkq6bkw7vwviRbW21joGBEF2muyK1Cc2FauVlFvgQNOR
aJ1WjnRcWojMlfcf2sIwue8h2YsAfnzeB/gYhH+gdxfdK8aZ user@example.local
```

Add

**SafeNet KeySecure k170v**

version 1.10.0 (build 2018)

[API & CLI Documentation](#)

[Product Documentation](#)

- The Error displayed is normal and simply requires the default **SSH Public Key** to be replaced. As the initial KeySecure admin (ksadmin) you must paste in your **SSH Public Key** in the box provided and then select **Add**.

**NOTE**

- > The SSH Public Key must be a 'PEM-formatted RSA key'. You can generate this key using 'PuTTYgen' or similar utility.
- > Save this SSH Public Key at a safe location. You will need this key for future SSH access.

After replacing the default SSH Public Key, the **Log In** screen appears.

- Log in using the initial default credentials: Username = admin, Password = admin

The following error is displayed:

**Error**

- Password change required
- 23 : NCERRPasswordChangeRequired: Password change required

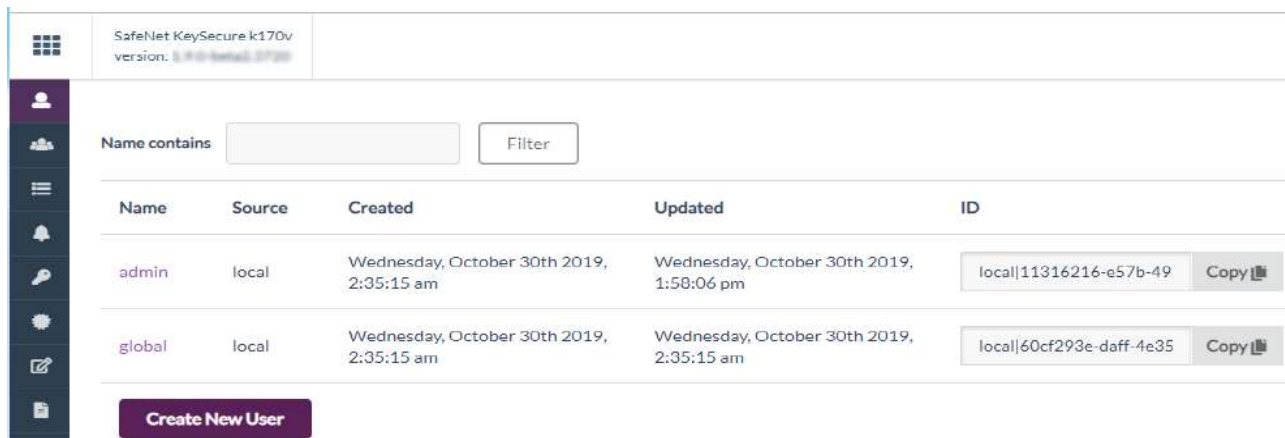
The **Error** displayed is normal and simply requires you must change the default Password.

- Enter a new password using this default Password Policy:

Min length: 8  
Max length: 30  
Min number of upper cases: 1  
Min number of lower cases: 1  
Min number of digits: 1  
Min number of other characters: 1

A new Login screen appears.

7. Using your new password, log in again. The **SafeNet KeySecure k170v** Web Page appears.



The screenshot shows the SafeNet KeySecure k170v web interface. At the top, it says 'SafeNet KeySecure k170v version: 1.10.0-Release.2720'. Below this is a sidebar with icons for user management, settings, and other functions. The main area has a 'Name contains' search bar and a 'Filter' button. Below the search bar is a table with columns: Name, Source, Created, Updated, and ID. The table lists two users: 'admin' and 'global'. The 'admin' user was created on Wednesday, October 30th 2019, at 2:35:15 am, and was updated on Wednesday, October 30th 2019, at 1:58:06 pm. The 'global' user was created on Wednesday, October 30th 2019, at 2:35:15 am, and was updated on Wednesday, October 30th 2019, at 2:35:15 am. Below the table is a 'Create New User' button.

| Name   | Source | Created                                  | Updated                                  | ID                            |
|--------|--------|------------------------------------------|------------------------------------------|-------------------------------|
| admin  | local  | Wednesday, October 30th 2019, 2:35:15 am | Wednesday, October 30th 2019, 1:58:06 pm | local 11316216-e57b-49 Copy   |
| global | local  | Wednesday, October 30th 2019, 2:35:15 am | Wednesday, October 30th 2019, 2:35:15 am | local 60cf293e-daff-4e35 Copy |

Congratulations! You have successfully deployed your NextGen KeySecure virtual machine.

## Using Cloud-init with Hyper-V

The SafeNet Virtual KeySecure k170v uses cloud-init to inject configuration information prior to first boot. Cloud-init is a standard configuration mechanism that is supported in Hyper-V. Cloud-init is supported as an ISO image for Hyper-V, mounted as media to use with the virtual CD/DVD drive. The ISO image must have two text files named **user-data** and **meta-data**.

Below is an example of setting up disk encryption and the user's ssh key using cloud-init and Hyper-V.

### Example: To setup disk encryption and the user's ssh key

Prepare the **user-data** file. This is a text file containing the specific configuration you want to use. Refer to section **"Cloud-init Configuration Options"** on page 24 for other available configuration parameters. Here is a simple **user-data** configuration file for setting up disk encryption and the user's ssh key.

```
#cloud-config
diskenc:
 encrypt: true
ssh_authorized_keys:
 - <replace with user's ssh public key>
```

8. Create the **meta-data** file. This is a text file containing user provided instance parameters, for example:



```
instance-id: <some instance id>
local-hostname: <host name for the instance>
```

**9. Create the ISO image file:**

- a.** Make sure **genisoimage** utility is installed.
- b.** Create the ISO file:

```
$ genisoimage -o config.iso -volid cidata -joliet -rock user-data meta-data
```

**10. Create a SafeNet Virtual KeySecure k170v VM in Hyper-V, but do not boot it.**

**11. In Hyper-V manager:**

- a.** Right click on the instance and go to "Settings...".
- b.** Under **IDE Controller 1 > DVD Drive > Specify** the media to use with your virtual CD/DVD drive, select Image File
- c.** Browse to the **config.iso** file generated in step 3 above.
- d.** Select OK.

**12. Boot up your k170v VM.**

## Replacing SSH key before system can boot

The NextGen KeySecure includes an initial SSH key for the System Admin "ksadmin" to use during launch. After launch this key must be replaced in order for the NextGen KeySecure to start all of its services and become fully functional.

### To determine if the default SSH key needs replacing:

**Using the GUI:**

- 1.** Browse to the KeySecure IP address.
- 2.** If prompted to enter the a new **SSH Public Key**, go to ["To replace the SSH key: " on the next page.](#)

**Using the CLI:**



**NOTE** This operation does not require authentication

Check if the default SSH key needs to be replaced.

```
$ ksctl instance status
```

**Sample Response:**

```
{"status": "bootstrap", "services":[...] "messages":
["The SSH public key for the ksadmin user must be replaced before the KeySecure will be
functional"]}
```

**Using the API:**



**NOTE** This operation does not require authentication.

Check if the default SSH key needs to be replaced.

```
GET https://{address}/system/instance/status
```

### To replace the SSH key:

#### Using the GUI:

After browsing to the KeySecure IP address and being prompted to enter a new **SSH Public Key**, you must paste in your SSH public key in the box provided and then select **Add**.



**NOTE** Your public key must be in the Linux SSH public key (pem) format.

#### Using the CLI:

```
$ ksctl ssh keys add -k "A..."
```

Upon successfully adding an authorized SSH key, the NextGen KeySecure services will start working momentarily.

#### Using the API:

```
POST https://{addr}/system/ssh/keys {"key": "A..."}
```

### To verify that the SSH key has been replaced:

Verify that the NextGen KeySecure services have started.

#### Using the GUI:

After replacing the default SSH Public Key, the **Log In** screen appears, verifying that the SSH key has been successfully replaced.

#### Using the CLI:

```
$ ksctl instance status
```

Sample Response:

```
{"status": "started", "services": [...]}
```

#### Using the API:

```
GET https://{addr}/system/instance/status
```

## Changing the password for the initial user (ksctl)

The username of the initial NextGen KeySecure user is 'admin'. This user becomes the initial Application Administrator.



**NOTE** To better understand the different user groups within the NextGen KeySecure, refer to the Administrator Guide > Concepts > Groups.

By default, the initial password for the initial Application Administrator is 'admin' in lowercase. The Application Administrator's initial password must be changed before using NextGen KeySecure.

## Using the Hard-coded password

By default, the initial password for the initial Application Administrator is 'admin' in lowercase. On initial log in, the Application Administrator is forced to change his password.

## Using an Auto-generated password

Optionally, the initial password for the initial Application Administrator can be auto-generated through cloud-init configuration using the following setting:

```
#cloud-config
keysecure:
 initial-user:
 generate-password: true
```

For more information on cloud-init configuration, see ["Using Cloud-init" on page 24](#).

After the initial user's password is auto-generated, it must be retrieved.

## Retrieving the Initial password for the Application Administrator

The auto-generated, initial password for the Application Administrator must be retrieved by the System Administrator "ksadmin" via SSH or via the CLI.

### To retrieve the initial password for the Application Administrator using SSH.

The initial password is written out to the file `/var/keysecure/password/out`. Retrieve it using SSH:

```
$ ssh ksadmin@{addr} cat /var/keysecure/password/out
```

If the ssh key used to launch the instance is not loaded in your ssh-agent, add the key file argument:

```
$ ssh -i {key file} ksadmin@{addr} cat /var/keysecure/password/out
```



**CAUTION!** After retrieving the initial password for the Application Administrator, the Application Administrator must log in and change it immediately. Failure to do so could allow an attacker to compromise the system.

### To retrieve the initial password for the Application Administrator using the CLI:

On Linux-like systems that have the `ssh` command line tool available, **ksctl** can be used to execute the `ssh` command to retrieve the initial password.

If you installed the CLI (see [the Administrator Guide > Operation > Installation](#)), you can use it to acquire the initial password as well:

```
$ ksctl init getpw --url https://<ipaddr> --identityFile <private key> --user admin
```



**CAUTION!** After retrieving the initial password for the Application Administrator, the Application Administrator must log in and change it immediately. Failure to do so could allow an attacker to compromise the system.

## Licensing

The NextGen KeySecure requires a valid license to create and manage keys. An initial 90-day Trial License is provided for evacuation. 30 days prior to license expiration, the NextGen KeySecure Server will notify you, in the top banner in red, that your license is expiring soon. Before the expiration, a new purchased license must be installed. Contact Gemalto Sales representative for assistance in obtaining a license.

### Lock Codes

Licensing requires a lock code. Each NextGen KeySecure server comes with two lock codes, the Key Manager Lock Code and the Connector Lock Code. These can be accessed via the GUI, KSCTL and API:

To access from the GUI, Select **Settings > Licenses**.

**Server Licenses**

Key Manager Lock Code [REDACTED] Copy

Connector Lock Code [REDACTED] Copy

**Add A New License**

License String

[REDACTED]

**Installed Licenses**

| Feature | State | Type |
|---------|-------|------|
|---------|-------|------|

The basic licensing steps are shown here using the GUI and KSCTL:

### To activate a license using the GUI

1. Retrieve the 'Key Manager Lock Code'.

In the KeySecure GUI, select **Settings > Licenses**.

2. **Copy** the 'Key Manager Lock Code'.
3. To receive a license string, go to the Gemalto Sentinel Cloud EMS Portal at:  
`https://dplicense.gemalto.sentinelcloud.com/ems/customerLogin.html`
4. Past in the 'Key Manager Lock Code' to generate a license string.
5. Copy the License String from the portal.
6. In the KeySecure GUI, activate your product by pasting in the License String in the **License String** field.

### To activate a license using the KSCTL

1. Retrieve the 'Key Manager Lock Code'.  

```
$ ksctl licensing lockdata
```
2. To receive a License String, go to the Gemalto Sentinel Cloud EMS Portal at:  
`https://dplicense.gemalto.sentinelcloud.com/ems/customerLogin.html`
3. Enter the Key Manager Lock Code to generate a license string.
4. Activate your product by entering the License String.  

```
$ ksctl licensing licenses add -l "<license string>"
```

## Connector/Client Licensing

A separate and unique 'Connector Lock Code' is provided by each NextGen KeySecure Server. This lock code is used to license supported SafeNet Connectors/Clients, e.g. SafeNet ProtectFile, as well to activate the KMIP interface.



**NOTE** Unlike the 'Key Manager Lock Code', the 'Connector Lock Code' is cluster-wide. This means that a license applied using the Connector Lock Code replicates across the cluster.

For connector licensing details, refer to Administrator Guide for the specific SafeNet connector you are using.

Refer to the *NextGen KeySecure Release Notes > Compatibility* section for information on supported SafeNet Connectors.

The Connector Lock Code is also used to license the KMIP interface. For details, refer to "KMIP Licensing" in the KMIP Reference Guide.

# Physical Appliance Deployment

This chapter describes how to deploy the NextGen KeySecure Software on a physical appliance.

This chapter contains the following topics:

- > ["Minimum Requirements" below](#)
- > ["Deploying on a Physical Appliance" below](#)
- > ["Replacing SSH key before system can boot" on page 107](#)
- > ["Changing the Password for the Initial User" on page 120](#)
- > ["Licensing" on page 123](#)

## Minimum Requirements

The NextGen KeySecure Software can be deployed on the following appliances:

### SafeNet KeySecure Classic Appliances:

- > SafeNet KeySecure k450
- > SafeNet KeySecure k460

### SafeNet Next Generation Keysecure Appliances:

- > SafeNet KeySecure k470
- > SafeNet KeySecure k570



**NOTE** Refer to these installation guides for specific details on installing and deploying these appliances.

- [SafeNet KeySecure k470 Installation Guide](#)
- [SafeNet KeySecure k570 Installation Guide](#)

## Deploying on a Physical Appliance

This section guides the System Administrator (ksadmin) and the initial Admin user in deploying the NextGen KeySecure software on a supported physical appliance (k450, k460 or k570).

## Prerequisites

The following items are required to deploy the NextGen KeySecure software on a SafeNet KeySecure physical appliance.

- > Supported Appliances: SafeNet KeySecure k450, k460 or k570
- > Firmware: NextGen KeySecure software, release 1.6.1 or later. (Contact Gemalto Technical Support)
- > PED: SafeNet Luna PED connected to the appliance, only if deploying on an appliance with an integrated Luna PCIe HSM card (k460 or k570)
- > Three PED keys (blue, red and black), only if deploying on a k460 appliance.



**NOTE** PED keys are only needed if the user wishes to make use of the k460 appliance's internal HSM card.

## Tasks

This section provides an overview of the tasks needed for deploying on a SafeNet KeySecure physical appliance (k450, k460 or k570).

### Tasks for deploying on the SafeNet KeySecure k450:

1. Install the NextGen KeySecure software as an upgrade to a SafeNet KeySecure k450 appliance. See ["Software Installation" on the next page](#).
2. Initialize the appliance. See ["Appliance Initialization" on the next page](#)
3. Activate the appliance. See ["Appliance Activation" on page 118](#)
  - Login as initial Admin user.
  - Obtain a license.
  - Activate a license.

### Tasks for deploying on the SafeNet KeySecure k460:

1. Install the NextGen KeySecure software as an upgrade to a SafeNet KeySecure k460 appliance. See ["Software Installation" on the next page](#).
2. Initialize the appliance. See ["Appliance Initialization" on the next page](#)
3. If applicable, initialize and configure the SafeNet Luna PCIe HSM Card within a SafeNet KeySecure Appliance, allowing the HSM card to act as the root of trust. See ["SafeNet Luna PCIe HSM Card Initialization" on page 116](#).
4. Setup the HSM as Root of Trust. See ["HSM Root of Trust Configuration" on page 117](#)
5. Activate the appliance. See ["Appliance Activation" on page 118](#)
  - Login as initial Admin user.
  - Obtain a license.
  - Activate a license.

6. Create a partition, and then generate and store root keys. See ["HSM Partition Creation" on page 119](#)



#### NOTE

### Tasks for deploying on the SafeNet KeySecure k470/k570:

The SafeNet KeySecure k470 or k570 appliance is shipped with NextGen KeySecure software installed.

1. Initialize the appliance. See ["Appliance Initialization" below](#)
2. If applicable, initialize and configure the SafeNet Luna PCIe HSM Card within a SafeNet KeySecure Appliance, allowing the HSM card to act as the root of trust. See ["SafeNet Luna PCIe HSM Card Initialization" on page 116](#).
3. Setup the HSM as Root of Trust. See ["HSM Root of Trust Configuration" on page 117](#)
4. Activate the appliance. See ["Appliance Activation" on page 118](#)
  - Login as initial Admin user.
  - Obtain a license.
  - Activate a license.
5. Create a partition, and then generate and store root keys. See ["HSM Partition Creation" on page 119](#)

## Software Installation

An existing SafeNet KeySecure k450 or k460 appliance can be upgraded with the NextGen KeySecure software.



**\*\*WARNING\*\*** During the NextGen KeySecure software upgrade the system is re-formatted. Therefore, before upgrading, a backup must be taken to preserve any existing data. This backup can then be migrated to the appliance after the NextGen KeySecure software upgrade.

After taking a backup of the k450 or k460 appliance, install the NextGen KeySecure software on the appliance:

1. Create a DVD from the ISO provided by Gemalto.
2. Boot the system from the DVD.

The installation proceeds automatically.

## Appliance Initialization

1. A physical appliance requires a console password to be created before the system will start up. Login to the console with the ksadmin user and create a secure password.



**NOTE** Be sure to retain this password - it will be required to access the system in case of network connectivity problems.



2. In the **Console Window** select the IP address for your network and access the NextGen KeySecure Web Page by browsing to this IP address.

The initial NextGen KeySecure Web Page screen is displayed:



3. The **Error** displayed is normal and simply requires the default **SSH Public Key** to be replaced. As the initial KeySecure admin (ksadmin) you must paste in your SSH public key in the box provided and then select **Add**.

#### NOTE

- > The SSH Public Key must be a 'PEM-formatted RSA key'. You can generate this key using 'PuTTYgen' or similar utility.
- > Save this SSH Public Key at a safe location. You will need this key for future SSH access.

After replacing the default SSH Public Key, the **Log In** screen appears.



**NOTE** The initial IP address is set via DHCP, which is displayed in the console. If you need to set a static IP address, set it from the console using the `nmcli` tool. For details, see ["Network Interface Configuration and Bonding" on page 50](#)

## SafeNet Luna PCIe HSM Card Initialization

If the appliance includes a SafeNet Luna PCIe HSM card, initialize this card using the following steps:

1. System Admin must SSH in to the appliance (or connect via serial port using their password) and execute `/usr/safenet/lunaclient/bin/lunacm` utility. The utility displays information on the detected HSM card and allows you to execute various HSM management commands.
2. If initializing a SafeNet Luna PCIe HSM K6 card, execute these commands in the following order and follow the PED instructions to initialize the HSM:



**NOTE** Please refer to Gemalto Luna PCIe HSM documentation for more details on these HSM commands.

```
lunacm:> hsm factoryReset
lunacm:> hsm init -label <partition-label>
lunacm:> hsm login
lunacm:> partition create
lunacm:> partition changepolicy -policy 22 -value 1
lunacm:> partition changepolicy -policy 23 -value 1
lunacm:> partition createChallenge
lunacm:> partition activate
lunacm:> hsm logout
lunacm:> partition login
```

3. If initializing a SafeNet Luna PCIe HSM K7 card, execute these commands in the following order and follow the PED instructions to initialize the HSM:



**NOTE** Please refer to Gemalto Luna PCIe HSM documentation for more details on these HSM commands.

- a. Make sure an HSM admin slot is selected.

### Optional:

To see the available slots, enter:

```
lunacm:> slot list
```

Look for a slot with description "Admin Token Slot".

To select the active slot, enter:

```
lunacm:> slot set -slot <number>
```

- b. Re-initialize HSM and initialize SO role (blue key, red key for domain):

```
lunacm:> hsm factoryReset
lunacm:> hsm init -label <admin token slot label>
```

### Optional:

```
lunacm:> slot list
```

Notice that the slot with description "Admin Token Slot" now has a label.

```
lunacm:> role login -n so
lunacm:> partition create
```

Optional:

```
lunacm:> slot list
```

Notice the slot with the slot description "User Token Slot". Remember the slot ID of this slot as this will be used later.

```
lunacm:> role logout
```

**c. Initialize partition and the partition SO role:**

```
lunacm:> slot set -slot <slot number of user token slot created above>
lunacm:> partition init -label <new partition label>
```

**d. Activate partition and initialize Crypto officer role:**

```
lunacm:> role login -name Partition SO
lunacm:> partition changepolicy -policy 22 -value 1
lunacm:> partition changepolicy -policy 23 -value 1
lunacm:> role init -name Crypto Officer
lunacm:> role createchallenge -name Crypto Officer
lunacm:> role logout
```

**e. Initialize Crypto User role:**

```
lunacm:> role login -name Crypto Officer
lunacm:> role changepw -name Crypto Officer
lunacm:> role changepw -name Crypto Officer -oldpw password -newpw Pa$$w0rd
```

**f. Relogin to cache the User Key and no longer require the PED:**

```
lunacm:> role login -name Crypto Officer
lunacm:> role logout
```



**NOTE** The sequence of commands above activate the newly created partition and enable the HSM auto-activation policy. This instructs the HSM to cache PED credentials and allows k570 appliance to authenticate to the HSM using only the challenge secret (password) without requiring the black PED key to always be connected to the HSM. However, in the event of a power outage of more than 2 hours, the HSM cached PED credentials will expire and the k570 appliance will fail to run its services, until you instruct the k570 appliance to re-authenticate with the HSM using the black PED key.

## HSM Root of Trust Configuration

Configure the KeySecure appliance to use the Luna PCIe HSM card using these steps:

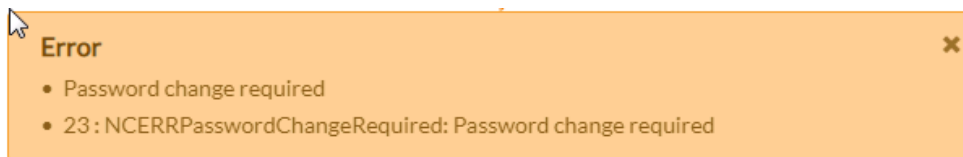
1. Access the 'partition-label' and the 'partition challenge' created during the HSM initialization procedure. These are needed to use the NextGen KeySecure HSM API or CLI `hsm setup` command.
2. You are now ready to configure the HSM as Root of Trust. Refer to NextGen KeySecure HSM Setup API or the CLI `hsm setup` command to configure the appliance to use your newly initialized Luna PCIe HSM card.

## Appliance Activation

To activate the appliance the initial Admin user performs these steps:

1. Log in using the initial default credentials: Username = admin, Password = admin

The following error is displayed:



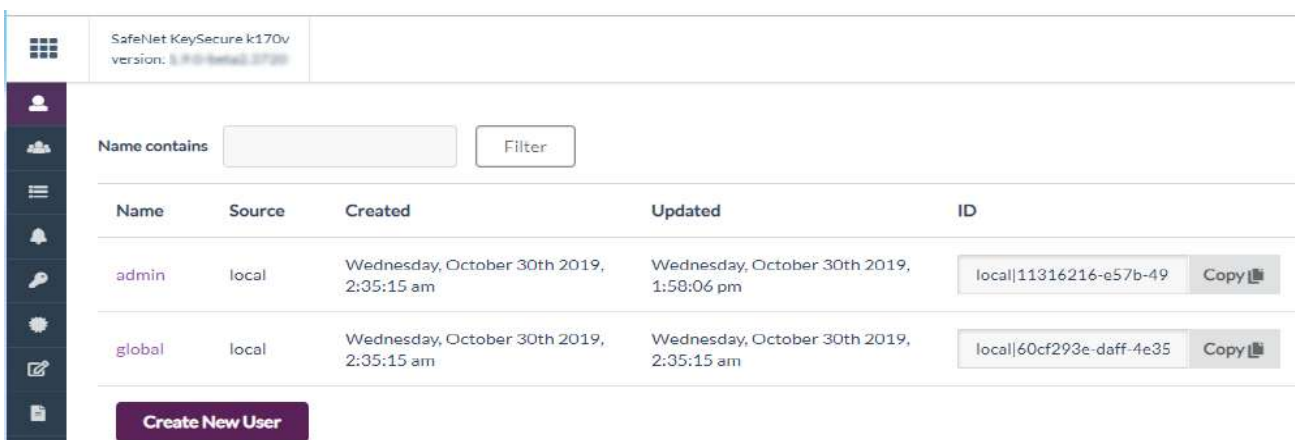
The **Error** displayed is normal and simply requires you must change the default Password.

2. Enter a new password using this default Password Policy:

Min length: 8  
 Max length: 30  
 Min number of upper cases: 1  
 Min number of lower cases: 1  
 Min number of digits: 1  
 Min number of other characters: 1

A new **Login** screen appears.

3. Using your new password, log in again. The **SafeNet KeySecure k170v** Web Page appears.





**NOTE** At the top of this page, the Gemalto bar highlights **One or more licenses are expiring soon**. This message reflects that there is a active 90-day trial license after initial login. You will need to replace the trial license within this period in order to continue using this SafeNet KeySecure k170v instance. To do this, refer to ["Licensing" on page 1](#).

## HSM Partition Creation

You are now ready to create an HSM Partition. To create a partition and then generate and store root keys, go to: Admin Guide > Concepts > Hardware Security Module > SafeNet Luna PCIe HSM (Type:lunapci)

## Troubleshooting

### Appliance BIOS Password:

SafeNet KeySecure k450 and k460 share the following BIOS password: protect

## Replacing SSH key before system can boot

The NextGen KeySecure includes an initial SSH key for the System Admin "ksadmin" to use during launch. After launch this key must be replaced in order for the NextGen KeySecure to start all of its services and become fully functional.

### To determine if the default SSH key needs replacing:

#### Using the GUI:

1. Browse to the KeySecure IP address.
2. If prompted to enter the a new **SSH Public Key**, go to ["To replace the SSH key: " on the next page.](#)

#### Using the CLI:



**NOTE** This operation does not require authentication

Check if the default SSH key needs to be replaced.

```
$ ksctl instance status
```

Sample Response:

```
{"status": "bootstrap", "services":[...] "messages":
["The SSH public key for the ksadmin user must be replaced before the KeySecure will be
functional"]}
```

#### Using the API:



**NOTE** This operation does not require authentication.

Check if the default SSH key needs to be replaced.

```
GET https://{address}/system/instance/status
```

---

### To replace the SSH key:

#### Using the GUI:

After browsing to the KeySecure IP address and being prompted to enter a new **SSH Public Key**, you must paste in your SSH public key in the box provided and then select **Add**.



**NOTE** Your public key must be in the Linux SSH public key (pem) format.

#### Using the CLI:

```
$ ksctl ssh keys add -k "A..."
```

Upon successfully adding an authorized SSH key, the NextGen KeySecure services will start working momentarily.

#### Using the API:

```
POST https://{addr}/system/ssh/keys {"key": "A..."}
```

---

### To verify that the SSH key has been replaced:

Verify that the NextGen KeySecure services have started.

#### Using the GUI:

After replacing the default SSH Public Key, the **Log In** screen appears, verifying that the SSH key has been successfully replaced.

#### Using the CLI:

```
$ ksctl instance status
```

Sample Response:

```
{"status": "started", "services": [...]}
```

#### Using the API:

```
GET https://{addr}/system/instance/status
```

---

## Changing the Password for the Initial User

The username of the initial NextGen KeySecure user is 'admin'. This user becomes the initial Application Administrator.



**NOTE** To better understand the different user groups within the NextGen KeySecure, refer to the Administrator Guide > Concepts > Groups.

By default, the initial password for the initial Application Administrator is 'admin' in lowercase. The Application Administrator's initial password must be changed before using the NextGen KeySecure.

**If using the GUI:**

On initial log in, the Application Administrator is forced to change the password.

**If using the CLI:**

Enter the new password using this command:

```
$ ksctl changepw --url=https://<ip-or-host> --user admin --password <new password>
```

## System Administrator's Initial Private Key

The System Administrator accesses the system via SSH. For Public Cloud Images, the initial Private Key is the one use to launch the instance. For Private Cloud Images, a public SSH key must be uploaded before system services will start. Until that point, the system can be accessed with the following hard-coded key. By default, NextGen KeySecure images use this hard-coded key.

```
-----BEGIN RSA PRIVATE KEY-----
MIIEKQIBAAKCAgEAnwGtotjvpBAukGHp/Y2xCfCuyPrm9KiMFyjreIJUKYe3yWKw
cFdBMh8cn8slAo7B/tIN/tYUnO8u+okx3apQDne4lmiCYeARXlvq3F8V2rVVtEPk
7d2vQ4TMI9CXfYNeJyiliyFlyMg4CDVnYM2bcJefFkoWZRDAnGG++ZHxKSc8Y/ge
nJH2RyLmCJYaYf6K6Y84ELtpTY1ty6Fd1t1mJanTIs4rAtz0cNbHI+1rK05ztiJO
/EOgywffogNXUs7lKHggk28K0gg190gJztQ+tTerdFlVcKwAzXrDOsHv6KHwzr5r
QYt19nmpGpHjGwNuXdx3fdgQLr/ZQzj5QU1VqZICG6aMiMFLMVkP3dnTK5om7RDv
DgPmTkj5DeldagBJrrhYsYdK6UgUqWosq7gQdGZqLuY8WtkKqBYTvP/GTPG7SBnJp
+muQu3E+7WHBvt+dPGlwiygFua5UtZcyldSditbUSDjYrPFEDW1aSQb6YsTHxklx
+AvuEoX6EBMA43yzFS9fEQZzpmDgAm0WXEzCQ0ZKcW322PJ2HEuAeclQuNzcUsiM
H+nLfBCKRQIaRpgRttKoqnzTkGtuDWPhxUDoq5N78XQ1CQDZNeNsBL9b8vpAGvxT
sSAC/dBOPbRUUls3s9f9p5vP3IHlpm/Vnbe+mLnwu8YNcW1U4DQf+m6SfTMCAWEA
AQKCAgBtwwLrv17tqh36nRyhYwYp87LtagW0LgVGl4C014M5Uq6T3gReCF2A8+kW
hWzu7w0LkoUGvVFNY+7SE5LQqZjJ4+s8VrhqR/q1YE4oAZtcdoDLB3MooDc0LJ+e
uYdZpS2BaQCngk54432KbRMHx+bdOk0fXdVHvuC3EuJ59SrJxcPyXn0DJ98SDA0v
MC7WJphN0gepxTVTy6VXQ/U7NVgzR5/wDtGyEETM//ZH3sR0CJvnxcvJ+Kh8zg
Ud4d06Rs4z/ZhnVQyf8pQGpC2rb08bbxghlwxctoDK5oCSHLNizKWrg/brLcCe+p
37seEqhFMsnPo5TIMLCrQGqnoXLXnO4B4cA9IM8wWmVvnyxndKGYz5aYn+tt37Un+
G7q55wJYgZrpR6JslhGdyQqiaLvMJP4cVmx9YpxuB87zir5m2xfq5L0TovukqJw
i1TRNNg8Ct25RtmfMsOrikDDccGo8LXiDrnwPQR6eDifp++tPsgvIwGFNohQNEkg
U/F3FAiCtzbagsZtGHfK5EdGij0EWfpyHXDginLlCpvIws010KJDyeDjSgVjrba
Qb03ogcP9ynRAevjdWSNi70pCCMs9kL3+q8tLEPO3dd/VrZIIHTROy7iXLj/a7cC
cfrR6mqoG7yEdO26yA3sASsFvrYkg8k/GNpRWJQZsherK09wQKCAQEAY2xcEW8l
C52GtkslSN2quxHKUlm+JbBWtE30FwCd0MxQ6MgW2dohTGj7C8PgfaNLDxhTiW
58MgZis4/vymPX73ybobG9ExKV59/fW9RA+TS7cxF594AP0kVmMeRXhleJCJuHiC
uOB/c5Kjduy4YvyPuyI4phXAfhZRU6yWp5RziP9uoy9quub4LCZWVXO9dXWEprRU
M9+TknIBJomCRR2g4Pb4cfG4+GemVjcDqHR7D10vuYZJOST20rwiF2zHE/kJb6Uo
KnLisUs/loFkph98GM5StdKuT4EKPz6DCyz8NHZfdq2IBhYm/NAnbo17IgNM9k2+
GMTr3fm0HY2mTWKCAQEAYBp0qZwmoozW5pM2AUuoyxzYlgZk+Nn64ZI9xaW0TYRX
rbrMA9HXinyeRc83S/ip4XlHej37fVLgXK+8DIicnqa/sZq0vvPdgAYsd04wmL3
Yq0PbwqxfanDmtPtCtZ84jt9SXx23jRbNdXRG+UpvD8U58K7O5O+djRKM5gFhV+
ehdIm78rylK5D+XlF5f+9oxwLhnc3MC8+luVJbrmisGLjTibjvZ+M1zUw+3NMZ/w
N8N9DO2MLhhh6/6EQMnfcUB13GmfuGDIzA3dBt6jB8UjWQPW+FybKtflIqK/9+wK
em0KnW2b+JQiJa0SiUEkxdUjPIeUUoujV8Zc1SF13QKCAQEAuA0yBNv0TKV0BQL
+AGX1+pgxGGuSStjR3+eFveSw5Vn3WqavBiz2M8xIWDzBwdWvpXZ8E9jOg6yIb65
svuk7b8TMtHP9Tsw390+4aiVth7hhoAZSuZeDYb8eaz0lqZ4KL2d3eorXFrC9i6W
9+L2Zuic6xPrbVoA8BQBQVPGe5k1b56s27jrOHAB2vBU7Lvrj3rkqnYfyU9glI1P
TS7dTuufw+uwnme2lAoz1Y2xIZlCPoRxycpWNG081Sejjuc8uDV69CjaTo3jzcXn
oqUwCiX0giMrcpdMX2LBgrdsMY/6IQNargYVzGv+mn87LiuUXRHfcHqmpIKHzi+n
Z49S9wQKCAQEASRKILhcaMTnEaaHXZmoXYjWSi5/49iWe1sHcm+w4k2gLVpc4NiEZ
RV1z43/mwiy/u1LHHyKrUw1/1Xs+77rNtNQ+GDppCW7/dQJTffU3R/zfwZg3UiH6
UqNJjtboAwrfFStjgQX9AkD+fbNKKg8uUYDY9QXjPaNOJbe7+DsR2DkJB01zuUkA
bTeqwCDUNB6y5mX8Vy9XvJlqIlw2rX6oyBzhawSu02HY54a8W000aTeJES+S10AW
jivRa058p2rcaJGMOphVXw12ZuvrWCSkCupTf/SWlFl1i8aev3eNnvQAepAPHnk
VCJx3Tk8b+0Pp2kml0e6vajnkKLePlQgmQKCAQEAhWzJQZAL8Rzcpn6BNKzkXJ/c
OOCj/YR38Ka4wkU4BSflzZjfrWUDsPlM8AfcuOeJEQRK4KE0pZSiTHIe25y+WTMt
Ot+PCjJhTTFAnbH7M73sEYD0Urz3B+5/3pQpMIxIDlCSKGZWroMut6rH8HYn4e6
K1OuVwKRWQR4LnmJvgU/xE6kyjiXdkfHFNEedjCYor2RohHUPluEzC223dYilqD3
+a2oVBdjOuEIUGap0DJ9zMCCARkJRCF5Rpf8/8/CVYVLEdL+vgEiiAP7Gv0Puw30
qXiKlPUJAlSmm2xJGim3DRNrnY1CwLH9pOfaRgtZ0inyOfOWAouRhtisxsz6fQ==
-----END RSA PRIVATE KEY-----
```





**NOTE** This default key can be replaced via cloud-init.

## Licensing

The NextGen KeySecure requires a valid license to create and manage keys. An initial 90-day Trial License is provided for evacuation. 30 days prior to license expiration, the NextGen KeySecure Server will notify you, in the top banner in red, that your license is expiring soon. Before the expiration, a new purchased license must be installed. Contact Gemalto Sales representative for assistance in obtaining a license.

### Lock Codes

Licensing requires a lock code. Each NextGen KeySecure server comes with two lock codes, the Key Manager Lock Code and the Connector Lock Code. These can be accessed via the GUI, KSCTL and API:

To access from the GUI, Select **Settings > Licenses**.

**Server Licenses**

Key Manager Lock Code [Copy]

Connector Lock Code [Copy]

**Add A New License**

License String

[Input Field]

**Installed Licenses**

| Feature | State | Type |
|---------|-------|------|
|---------|-------|------|

The basic licensing steps are shown here using the GUI and KSCTL:

### To activate a license using the GUI

1. Retrieve the 'Key Manager Lock Code'.  
In the KeySecure GUI, select **Settings > Licenses**.
2. **Copy** the 'Key Manager Lock Code'.
3. To receive a license string, go to the Gemalto Sentinel Cloud EMS Portal at:  
<https://dplicense.gemalto.sentinelcloud.com/ems/customerLogin.html>

4. Past in the 'Key Manager Lock Code' to generate a license string.
5. Copy the License String from the portal.
6. In the KeySecure GUI, activate your product by pasting in the License String in the **License String** field.

### To activate a license using the KSCTL

1. Retrieve the 'Key Manager Lock Code'.
2. To receive a License String, go to the Gemalto Sentinel Cloud EMS Portal at:

<https://dplicense.gemalto.sentinelcloud.com/ems/customerLogin.html>

3. Enter the Key Manager Lock Code to generate a license string.
4. Activate your product by entering the License String.

```
$ ksctl licensing licenses add -l "<license string>"
```

## Connector/Client Licensing

A separate and unique 'Connector Lock Code' is provided by each NextGen KeySecure Server. This lock code is used to license supported SafeNet Connectors/Clients, e.g. SafeNet ProtectFile, as well to activate the KMIP interface.



**NOTE** Unlike the 'Key Manager Lock Code', the 'Connector Lock Code' is cluster-wide. This means that a license applied using the Connector Lock Code replicates across the cluster.

For connector licensing details, refer to Administrator Guide for the specific SafeNet connector you are using.

Refer to the *NextGen KeySecure Release Notes > Compatibility* section for information on supported SafeNet Connectors.

The Connector Lock Code is also used to license the KMIP interface. For details, refer to "KMIP Licensing" in the KMIP Reference Guide.

# APPENDIX A: k470/k570 Physical Appliance

## Hardware

This document describes the administrative and maintenance tasks you can perform directly on the SafeNet KeySecure k470 or k570 Appliance . It contains the following sections:

- > ["Physical Features" below](#)
- > ["System Behavior with Hardware Tamper Events" on page 128](#)
- > ["Power-on, Power-off, or Reboot the Appliance" on page 130](#)
- > ["Power Supply and Fan Maintenance" on page 131](#)
- > ["HSM Emergency Decommission Button" on page 137](#)
- > ["Serial Connections" on page 138](#)
- > ["Front Locking Bezel" on page 140](#)
- > ["Power Consumption" on page 141](#)

## Physical Features

The SafeNet KeySecure k470 and k570 Appliances are 1U high and fit into standard 19-inch equipment racks.

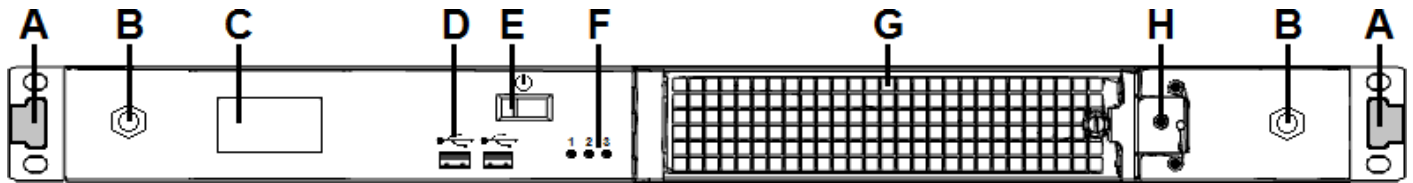


### Available Models:

- k470 - no HSM, 4x1Gbit ports
- k470 - no HSM, 2x1Gbit ports and 2x10Gbit ports
- k570 - HSM, 4x1Gbit ports
- k570 - HSM, 2x1Gbit ports and 2x10Gbit ports

## Front Panel (all models)

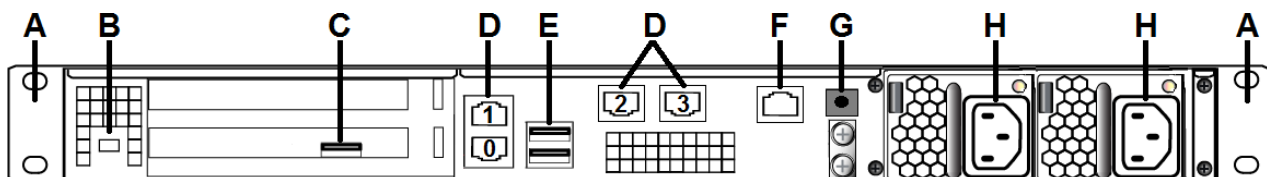
The front panel is illustrated below, with the secure locking bezel removed:



| Item | Name                         | Description                                                                                                                                                                                                                                                                                                                                                          |
|------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A    | Front ear brackets           | Connect to the front of the appliance chassis with the provided screws, allowing it to be mounted in a standard 19-inch equipment rack. The extending tabs act as posts for the locking bezel.                                                                                                                                                                       |
| B    | Mounts for locking bezel     | The secure locking bezel connects to the appliance faceplate here.                                                                                                                                                                                                                                                                                                   |
| C    | Front-panel display          | Displays basic configuration and status information for the appliance. See also <a href="#">"Front-panel LCD Display" on page 1.</a>                                                                                                                                                                                                                                 |
| D    | USB 3.0 ports                | These USB ports are not used and are disabled.                                                                                                                                                                                                                                                                                                                       |
| E    | Start/stop switch            | Powers the appliance on or off. See also <a href="#">"Power-on, Power-off, or Reboot the Appliance" on page 130.</a>                                                                                                                                                                                                                                                 |
| F    | Fan status LEDs              | The appliance has three (3) cooling fans. If these lights are illuminated, the fans are working correctly.                                                                                                                                                                                                                                                           |
| G    | Ventilation fan filter cover | Removable cover allows cleaning of air filter. See also <a href="#">"Power Supply and Fan Maintenance" on page 131.</a>                                                                                                                                                                                                                                              |
| H    | Fan bay securing screw       | Torx screw secures the fan bay.<br><div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <b>**WARNING**</b> Opening to swap fan modules triggers a tamper event on the appliance. See also <a href="#">"Power Supply and Fan Maintenance" on page 131.</a></div> |

## Rear Panel (all models)

The rear panel is illustrated below:

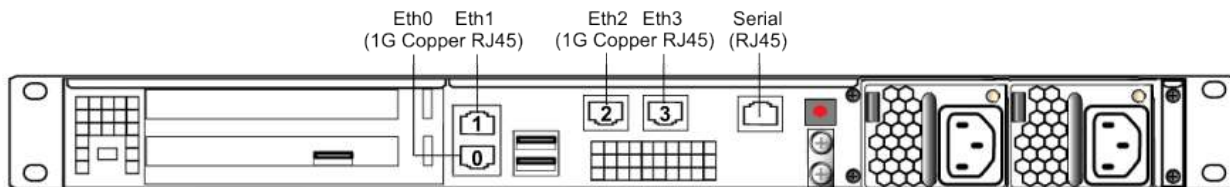


| Item     | Name                                       | Description                                                                                                                                                                                                                       |
|----------|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>A</b> | Sliding rail brackets                      | Connect to the sliding rails mounted on the sides of the appliance chassis, allowing it to be mounted in a standard 19-inch appliance rack.                                                                                       |
| <b>B</b> | Kensington lock connector                  | Allows the appliance to be secured to a desk or equipment rack using Kensington lock.                                                                                                                                             |
| <b>C</b> | Models with HSM:<br>HSM card with USB port | When authenticating with a PED, the PED must be connected directly to the USB port of the HSM card. The other USB ports on the appliance will not work for PED connection.                                                        |
| <b>C</b> | Models without HSM:<br><i>No HSM card</i>  | <i>No HSM card installed - a blank plate is installed.</i>                                                                                                                                                                        |
| <b>D</b> | LAN ports                                  | The appliance has a total of four (4) 1 Gbit RJ45 Ethernet ports that can be bonded in any mode supported by the Linux kernel.                                                                                                    |
| <b>E</b> | USB 3.0 ports                              | These USB ports are not used and are disabled.                                                                                                                                                                                    |
| <b>F</b> | RJ45 serial port                           | Connect a terminal to this port using the included RJ45 to USB cable.                                                                                                                                                             |
| <b>G</b> | Decommission button                        | This button should only be pressed as part of decommissioning and zeroizing the appliance. See also <a href="#">"HSM Emergency Decommission Button" on page 137.</a>                                                              |
| <b>H</b> | Power supplies                             | Connect the appliance to power. For proper redundancy and best reliability, the power cables should connect to two completely independent power sources. See also <a href="#">"Power Supply and Fan Maintenance" on page 131.</a> |

## Next Generation KeySecure Network Interface

### Network Interface for 1Gbps Models

1Gbps models come with four 1 Gbit RJ45 Ethernet network ports (labeled 0, 1, 2, and 3). The network interfaces are mapped to `Ethn`, as detailed in the following illustration.

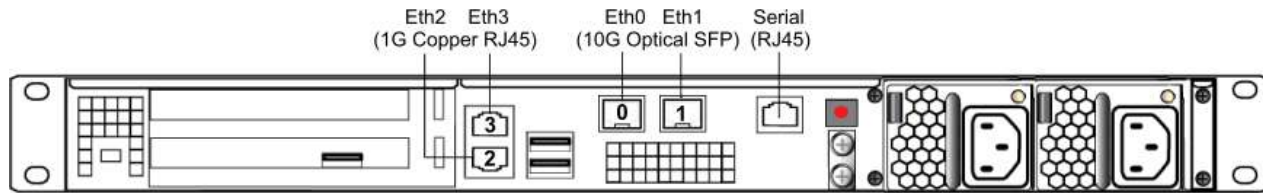


All ports are identical and equally usable and can be bonded in any combination. By default port `Eth0` is configured for DHCP but this can be configured.

Refer to section ["Network Interface Configuration and Bonding" on page 50](#) for more information.

## Network Interface Mapping for 1Gbps/10Gbps Models

10 Gbps models provide two 10Gbps SFP optical Ethernet network ports (labeled 0 and 1) and two 1Gbps copper RJ45 network ports (labeled 2 and 3). The network interfaces are mapped to Eth $n$ , as detailed in the following illustration.



You can optionally bond the network interfaces Eth0 with Eth1 or Eth2 with Eth3 to form a logical interface, providing a redundant active/standby virtual interface. By default port Eth0 is configured for DHCP but this can be configured.

Refer to section ["Network Interface Configuration and Bonding" on page 50](#) for more information.

## System Behavior with Hardware Tamper Events

### Tampering with the Appliance

Hardware tamper events are detectable events that imply intrusion into the appliance interior.

One such event is removal of the lid (top cover). The lid is secured by anti-tamper screws, so any event that lifts that lid is likely to be a serious intrusion.

Another event that is considered tampering is opening of the bay containing the ventilation fans.

You can use the thumbscrew to access the mesh air filter in front of the fans, without disturbing the system. However, if you open the fan-retaining panel behind that, which requires a Torx #8 screwdriver, then the system registers a tamper.

Therefore, cleaning of the filter is encouraged, especially if you work in a dusty environment, but fan module removal and replacement are discouraged unless you have good reason to suspect that a fan module is faulty. See ["Power Supply and Fan Maintenance" on page 131](#) for more information.

### Decommission

The red "Decommission" button recessed behind the back panel is not a tamper switch. Its purpose is different. See ["HSM Emergency Decommission Button" on page 137](#) for a description.

### What Happens When You Tamper - Including Opening the Fan Bay

First, the HSM disconnects and disallows any communication.

Second, the system turns on the "HSM Offline" alarm to alert system admins. This will happen within 30 seconds.

Third, if the system is unable to communicate with the HSM consecutively for 5 minutes, it restarts itself to ensure that all sensitive cached information is erased from the running system. If the system is still unable to communicate with the HSM after a restart, services are prevented from starting up and the status API reflects this problem. In addition, the GUI displays an error indicating a problem communicating with the HSM.

A tampered system does not recover by itself and requires the system admin to intervene. Recovering requires console access (for example SSH) and access to the HSM PED and the corresponding HSM tokens.

## To clear a tamper flag on HSM

1. Make sure an HSM admin slot is selected.

- a. Run the `/usr/safenet/lunaclient/bin/lunacm` tool in the console as the system admin (ksadmin).

- b. Check the available slots.

```
lunacm:> slot list
```

- c. Look for a slot with description "Admin Token Slot".

- d. Select the active slot.

```
lunacm:> slot set -slot <number>
```

2. Verify that the system is unable to communicate with the HSM using this command:

```
lunacm:> hsm tampershow
```

Response:

```
Driver command failed: Input/output error
```

```
Failed to display tamper error=5.
```

```
Command Result : No Error
```

3. An HSM restart is required before any operations can be performed against the HSM using this command:

```
lunacm:> hsm restart
```



**NOTE** If the PED is connected to the system, it is required at this time to reconnect it by physically unplugging and reconnecting the USB cable; otherwise it will not be able to communicate with the HSM.

4. Make sure an HSM admin slot is selected. Refer to step 1 for details.
5. When the HSM has restarted, it is possible to see the tamper event using this command:

```
lunacm:> hsm tampershow
```

6. You can now clear the tamper flag using these commands:

```
lunacm:> role login -name so
```

```
lunacm:> hsm tamperclear
```

At this point the HSM should be operational again.



**NOTE** During tampering all cached HSM tokens are cleared. Therefore, any KeySecure attempt to log in to the HSM will trigger PED authenticate. Re-authentication using the black PED key is required to get KeySecure up and running again.

## Power-on, Power-off, or Reboot the Appliance

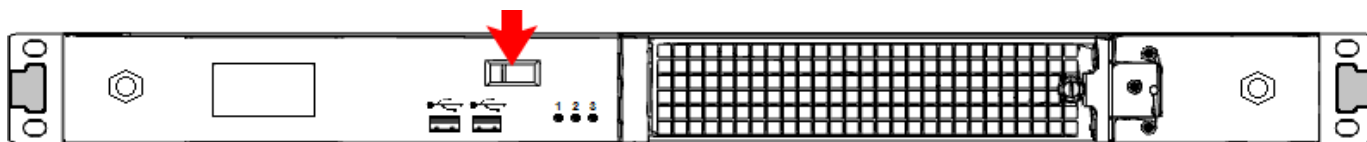
This section describes how to power-on, power-off, or reboot the appliance. It contains the following sections:

- > ["Power On" below](#)
- > ["Power Off" below](#)
- > ["Reboot" on the next page](#)
- > ["Automatic Restart Following a Power Interruption" on the next page](#)

### Power On

On the back panel, ensure that the power supplies are connected and working - the green LED on each power supply glows steadily when in operation.

If the appliance does not immediately begin to start up, press and release the START/STOP switch  on the front panel.



The appliance begins to power up.

If power was removed while the system was on (either a power failure, or the power cable was disconnected), the system restarts without a button press. This behavior allows unattended resumption of activity after power interruption.

The front-panel LCD begins showing activity, then settles into the ongoing system status display once the appliance has completed its boot-up and self-test activity.

### Power Off

To power-off the appliance locally, press and release the START/STOP switch. Do not hold it in. The appliance then performs an orderly shutdown (that is, it closes the file system and shuts down services in proper order for the next startup). This takes approximately 30 seconds to complete. In the unlikely event that the system freezes and does not respond to a momentary "STOP" switch-press, then press and hold the START/STOP switch for five seconds. This is an override that forces immediate shutoff.



**CAUTION!** Never disconnect the power by pulling the power plug. Always use the START/STOP switch.



To switch off the appliance from the console command line, use the command **sudo poweroff**.

## Reboot

To perform a system restart, you can switch the power off and then on again using the momentary-contact START/STOP switch on the front panel of the system, or issue the **sudo reboot** command.

To switch off the system, issue the **sudo poweroff** command, or use the START/STOP switch on the appliance front panel:

- > If you issue the **poweroff** command, the graceful shutdown takes 15 to 20 seconds
- > After you momentarily press and release the START/STOP switch, the system performs a graceful shutdown, which takes 15 to 20 seconds.

If the system does not appear to be properly shutting down, then press and hold the front-panel START/STOP switch, which forces an immediate shutdown. This is not normally required, and should never be done unless it is required, since it bypasses the normal, graceful file-system closing and shutdown procedure.

## Automatic Restart Following a Power Interruption

If the appliance was deliberately powered down, using the START/STOP switch or the **sudo poweroff** command, it remains off until you press the START/STOP switch. However, if power was removed while the system was on (either a power failure, or the power cables were disconnected - not good practice), then the system restarts without a button press.

This behavior allows unattended resumption of activity after power interruption. In most cases, it is assumed that this would never be needed, as you would install the appliance with its two power supplies connected to two completely separate, independent power sources, at least one of which would be battery-backed (uninterruptible power supply) and/or generator-backed.

## Power Supply and Fan Maintenance

The two power supplies in the SafeNet KeySecure k470 or k570 Appliance are hot-swap capable, meaning that one is sufficient to power the appliance while the other is removed and replaced, with no service interruption. The indicator light (LED) on each power supply shows different behavior, depending upon conditions.

| Power Supply Condition                          | Power Supply LED                    |
|-------------------------------------------------|-------------------------------------|
| DC present/only standby output on               | Flashing green (1Hz)                |
| Power supply DC output ON and OK                | Steady green                        |
| Power supply failure                            | Steady RED                          |
| Power supply warning                            | Flashing Blue/Red (1Hz) alternating |
| Input power failure (only in n+1 configuration) | Flashing Red (1Hz)                  |

A power supply controller in the appliance monitors the state of the power supplies. It ensures that a failed power supply still gets sufficient direct current from the remaining power supply to light the indicator LED. The controller also sounds an audible alarm when there is a problem, such as one power supply not being connected to AC main power.

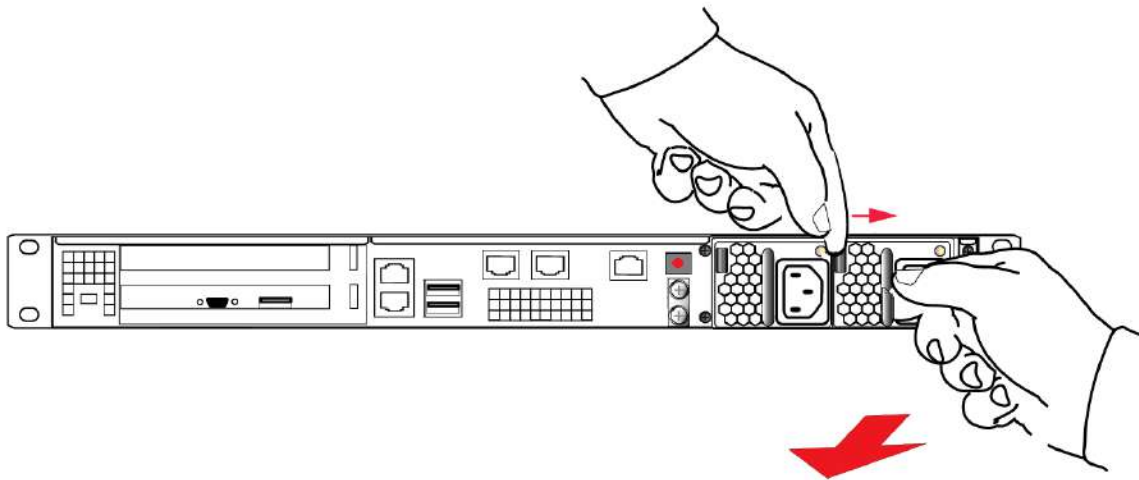
If only one power supply is present, the audible alarm is silent. If you wish to operate your appliance with only one power supply, we recommend that you remove the second supply to silence the audible alarm.

## Replacing a Power Supply

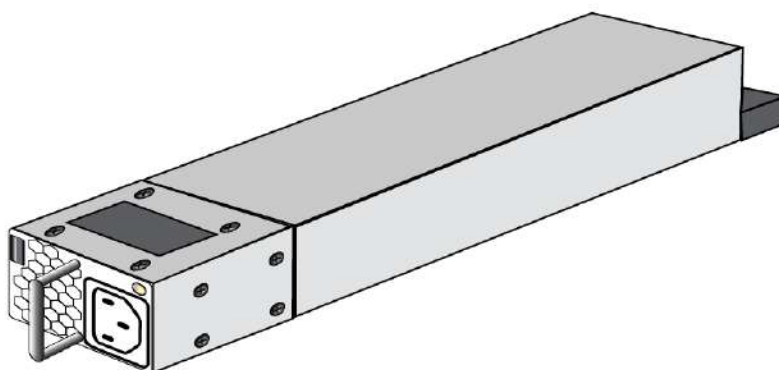
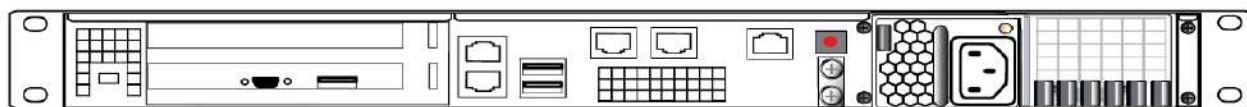
You may need to replace a power supply in the event of a failure.

### To remove a power supply:

1. To remove a power supply, face the back of the appliance.
2. Disconnect/unplug the selected power supply.
3. Press the lever sideways to release the power supply retaining catch, and simultaneously pull the handle out toward you.



4. Withdraw the power supply completely, using your other hand to support the body of the power supply as it emerges.



### To Reinstall a Power Supply:

1. To replace a power supply, reverse the steps above. Press firmly to seat the connector. The power supply can be fully inserted only in its proper orientation.
2. Connect an AC power cord.

## The Fans

In normal operation, the fans should require no maintenance.

You might need to perform the following tasks:

- > Clean the filter (occasionally)
- > Replace a defective fan (rarely)



### **CAUTION!** Opening the fan bay causes a system tamper event

We recommend that you use scheduled system maintenance downtime for this activity, as it will temporarily disrupt your client's access to your HSM partitions. If the system detects a tamper event, the HSM stops responding until you reboot (**sysconf appliance reboot**), or until you use the Stop/Start switch on the appliance rear panel.

## Cleaning the Filter

The ventilation grille, located to the right, on the appliance front panel, is secured in two parts, by two screws - a knurled, captive thumb-screw, and a Torx T8 screw. The knurled screw can be fastened or released without tools. It secures the lattice screen that in turn retains the mesh air filter.

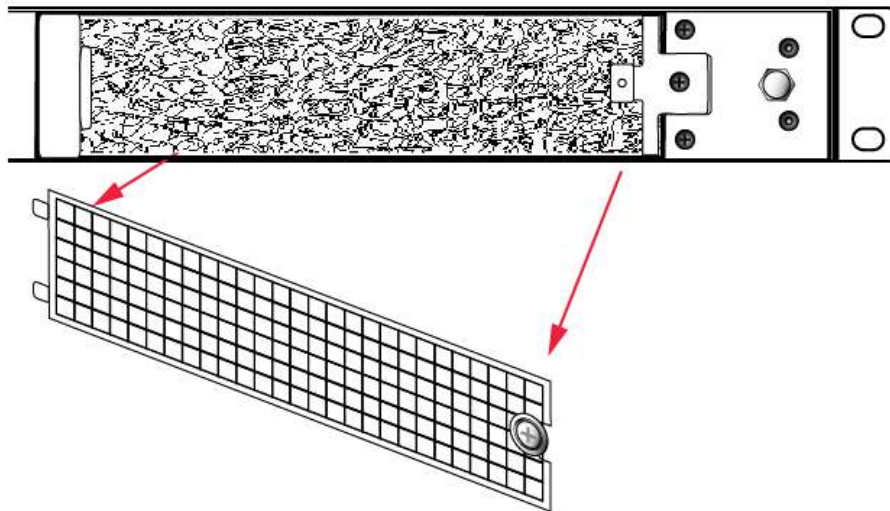
While we recommend controlled-atmosphere environments for greatest longevity and reliability of the equipment, we recognize that some environments might include some dust in the air. The mesh filter traps larger particulate matter before it can be drawn into the interior of the appliance. In less-than-perfect non-clean-room conditions, the mesh might accumulate a buildup of dust, and should be cleaned occasionally for best cooling airflow into the equipment.



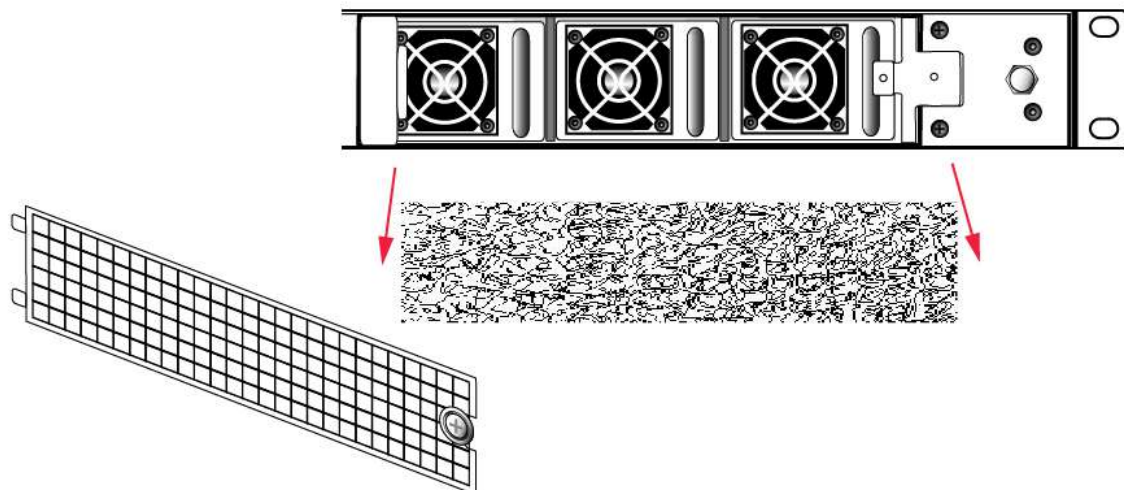
**NOTE** Accessing the air filter mesh in front of the fans (using the thumbscrew to open the retaining grille) does not cause a tamper.

### To clean the filter:

1. Twist the knurled knob counter-clockwise until it no longer secures the airflow lattice. The lattice is anchored at its left end by two tabs, and can be easily pulled off the appliance, once the knurled retaining screw is loosened. Do so.



2. With the air filter exposed, it is easy to grasp the mesh with fingers and tug it free. The mesh is flexible and is held in its cavity only by friction. If it is dusty, handle carefully so as not to dislodge any dirt that could then be sucked in by the fans.



3. To clean the filter, either blow it out with compressed air (away from the vicinity of the appliance), or rinse with water. If using water, ensure that the mesh is dry before reinstalling.
4. To reinstall the mesh, place it in its cavity in front of the fans, and use fingers or a blunt tool to tuck-in the corners.
5. Then, replace the lattice in front of the mesh by inserting the tabs first, then swinging the lattice closed like a door, and securing with the knurled screw.

## Replacing a Fan

The three fan modules (each containing two in-line fans) provide cooling redundancy. If one fan or module fails, it is detected by sensors. View a summary of appliance sensor conditions by running the console command **status sensors**. In the FAN section of the command output, the fans are listed in the order that they appear, left-to-right, as viewed from the front of the appliance. The example shows a fault with the first fan module:

```
----- Front Cooling Fans Status -----
FAN1A lnr 0 RPM Unplugged or Failed
FAN1B lnr 0 RPM Unplugged or Failed
FAN2A OK 3000 RPM
FAN2B OK 2900 RPM
FAN3A OK 2900 RPM
FAN3B OK 3000 RPM
```

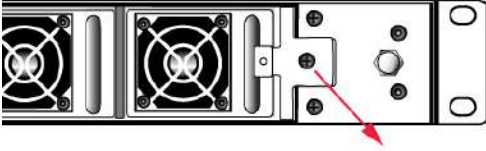
When the system returns from restarting, the HSM returns to find both splits of the MTK available and it immediately reconstitutes the MTK, allowing you to resume operations.



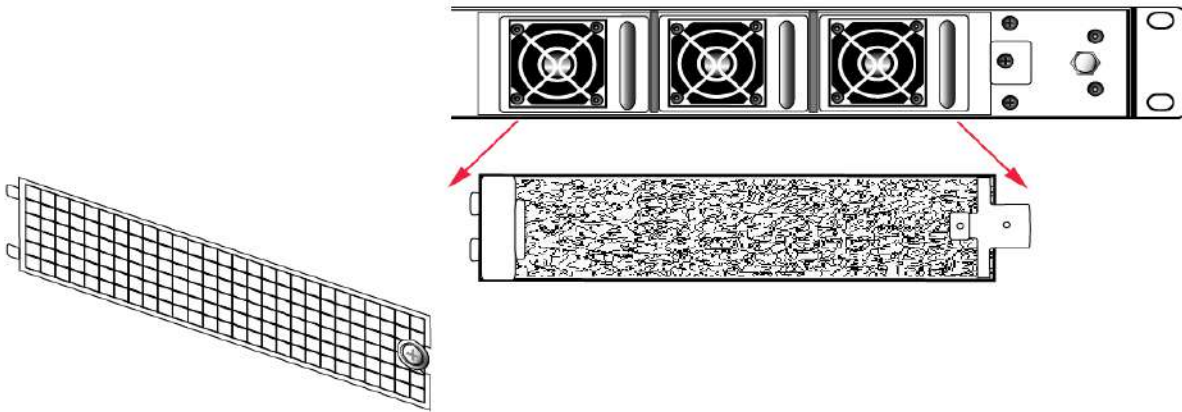
**NOTE** Partition authentication data is de-cached by the tamper - you must execute **partition activate -partition <name\_of\_partition>** each of your HSM partitions before your clients can resume accessing them. Partition activation does not survive a tamper event. In either case, you can examine the log for tamper events: **syslog tail -search tamper -entries 200**

## To replace a fan:

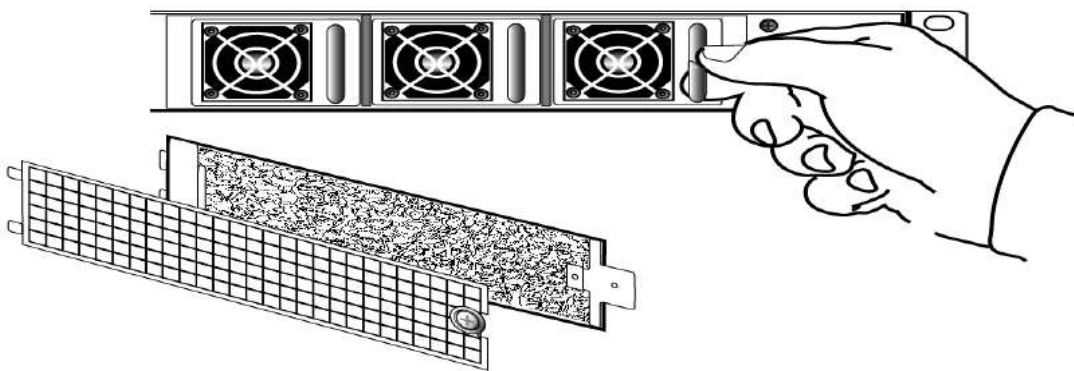
1. To open the fan bay, use a Torx number 8 screwdriver to remove the screw that secures the right-side tab of the fan retainer.



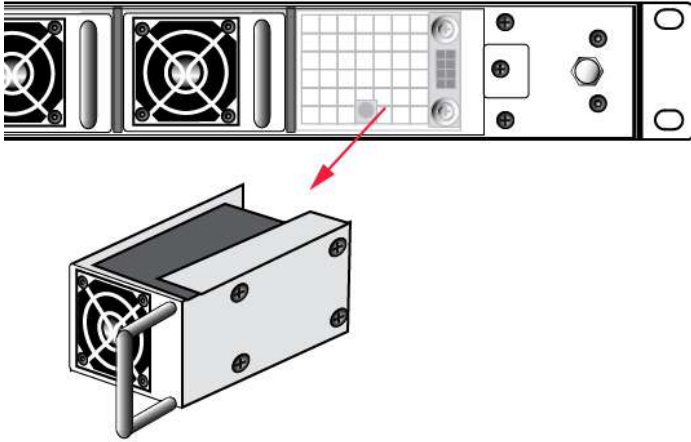
2. The fan retainer is anchored at its left by two tabs - swing the retainer out like a door, and remove it. There is no need to separate the filter mesh and its retainer from the larger fan retainer; the assembly can come out as one piece. The illustration below happens to show them separated.



3. The fan modules are now exposed and are held in place only by the friction of their electrical connectors.
4. Grasp the handle of the selected fan module and pull straight out toward you.



5. After slight initial resistance, the fan module should easily slide free of the appliance.



6. To replace the fan module or install a new one, reverse the above sequence.  
The index peg on the back of the module, and the matching index hole at the back of the fan bay, ensure that the module can be inserted only in its proper orientation.
7. Close up, replace the bezel, reconnect any cables, and return the appliance to service. If the power was left on during the operation, you will nevertheless need to restart (**sysconf appliance reboot**) in order to clear the tamper event caused by opening the fan bay.
8. You will also need to re-Activate your HSM Partitions (**partition activate -partition <name\_of\_partition>**), so that they once more become available to your registered clients.

## Summary

Removing, cleaning, and replacing the fan filter (the black mesh behind the grille) does not cause a tamper, and can be done at any time without disrupting your Clients.

Opening the fan bay (behind the filter), by unscrewing that Torx screw, does cause a tamper and therefore some down-time for your Clients. If only one fan module is showing a defect, you can probably leave replacing it until scheduled down-time, during which there would be no unexpected disruption to your Clients.

## HSM Emergency Decommission Button

The SafeNet KeySecure k470 or k570 Appliance includes a way to decommission the HSM, or permanently deny access to all objects on it, without need for either a serial console or a remote (SSH) connection.

To directly decommission the HSM inside the SafeNet appliance, press and release the small red button on the rear panel.

- > The appliance does not need to be powered on.
- > The appliance does not need to have power cables connected.





You will need a small screw-driver or other tool to reach the Emergency Decommission button. This is intentional, to prevent accidental pressing of that button.

## What the Emergency Decommission Button Does

When you press the Decommission button, all partitions and their contents are deleted, as well as the audit role, and the audit configuration. The HSM policy settings are retained.

To bring the HSM back into service, you need to:

1. Reinitialize the HSM
2. Reinitialize the audit role and reconfigure auditing
3. Recreate the partitions
4. Reinitialize the partition roles

## Serial Connections

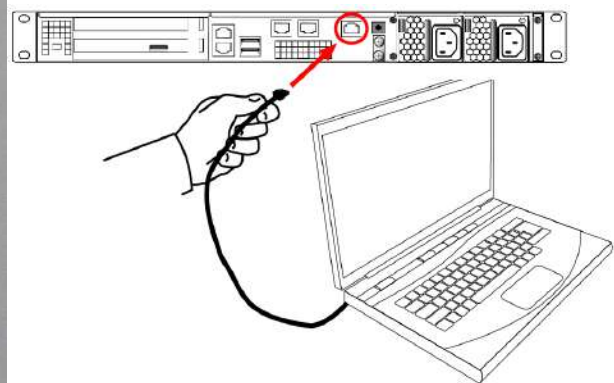
You can connect a computer directly to the console port of the SafeNet KeySecure Appliance using a serial connection. From the console you log in as the System Administrator (aka, 'ksadmin' user), create a secure password, start-up the system and access the IP address of the Graphical User Interface (GUI).

Direct administration connection to the console via serial terminal is required for these reasons:

- > During initial configuration, do not yet know the IP address dynamically allocated by your DHCP server.
- > After deployment, if you re-configure network settings (i.e., change the IP address) via SSH, you will lose the old IP address connection.

### To open a serial connection:

1. Connect the serial port on the appliance's rear panel to a terminal server, dumb terminal, PC, or laptop, using the supplied Prolific Technology Inc. USB to RJ45 (with 8P8C connector) adapter.



2. If the driver for the Prolific Technology Inc. USB to RJ45 (with 8P8C connector) adapter did not download and install automatically, go to <http://www.prolific.com> to download and install the PL2303 USB-to-Serial Windows driver.



3. Open **Device Manager (Control Panel > Hardware > Device Manager)** and expand the **Ports (COM and LPT)** folder. If the driver installed successfully, an entry is displayed for the **Prolific USB-to-Serial Comm Port**, followed by the port associated with the adapter. For example:

Prolific USB-to-Serial Comm Port (COM4)

Record the COM port (COM4 in this example) associated with the adapter. You will need this port number when you open a serial connection.

4. Use a terminal emulation package, such as PuTTY, to open a serial connection to the COM port associated with your Prolific USB-to-Serial adapter. Set the serial connection parameters as follows:

**Baud rate:** 19200

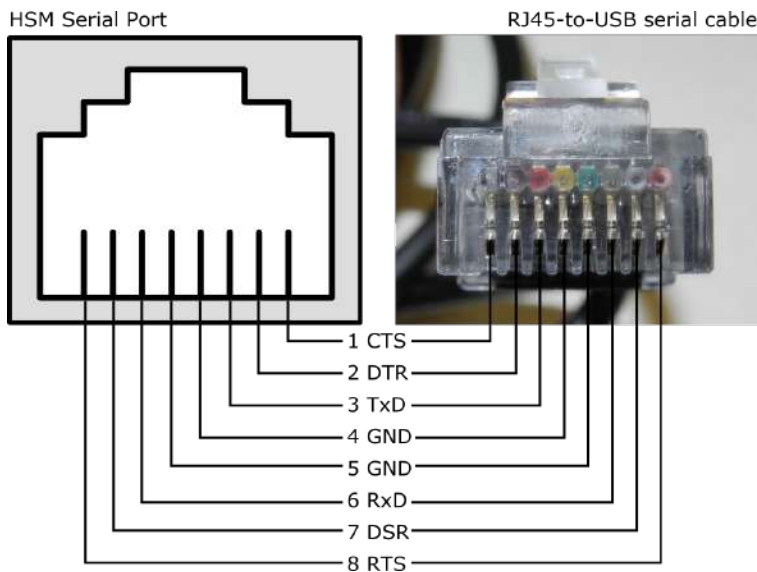
**Data bits:** 8

**Parity:** None

**Stop bits:** 1

#### Serial Port Pinout

The serial port uses a configuration equivalent to the Cisco Terminal Console. The Prolific Technologies Inc. RJ45-to-USB serial adapter cable uses a standard RJ45 pinout configuration:



5. When the connection is made, IP address information appears together with the appliance log in prompt:  
**keysecure login:.**



**NOTE** You may need to press **ENTER** several times to initiate the session.



**NOTE** Windows 10 occasionally crashes when trying to detect a serial port. This is a known issue with the Windows 10 PL2303 drivers. If you experience trouble opening a serial connection using Windows 10, use another supported operating system.

6. As the System Administrator, enter "ksadmin" to log in and follow the prompts to create a secure password.



**CAUTION!** Be sure to retain this password - it will be required to access the system in case of network connectivity problems.

The system starts up, which can take several minutes.

## Front Locking Bezel

The locking bezel (pictured below) fits over the front of the SafeNet KeySecure k470 or k570 Appliance for maximum physical access security. The purpose of the bezel is to:

- > cover the appliance's ports, and the power button,
- > lock the appliance to the rack to prevent removal.



The locking bezel comes with three (3) keys for each lock. The locks are keyed differently so that keys can be issued to different security personnel and kept in secure, separate locations.

### To lock the bezel:

1. The locks fit over the posts highlighted below. Fit the bezel over the posts with both keys in the horizontal position.



2. Turn the keys to the vertical position to lock the bezel.



Remove the keys and store them in a secure location.

## Replacement Keys

To obtain replacement keys, contact Technical Support (see ["Support Contacts" on page 13](#)). Please have the lock serial numbers ready. You can find these numbers on the sides of the bezel by each lock.

## Power Consumption

When installed and connected to appropriate electrical power sources, SafeNet KeySecure k470 or k570 Appliance draws power as follows:

| Activity                                                      | Draw                                                               |
|---------------------------------------------------------------|--------------------------------------------------------------------|
| Standby (connected to AC electrical mains but not powered on) | 26W (typical)                                                      |
| Power-on Input Surge                                          | 15A (typical)<br>40A at 90-132VAC (max)<br>60A at 180-265VAC (max) |
| Active (under load from clients)                              | 84W to 90W (typical)<br>100W to 105W (max)                         |

The appliance has two power supplies, each rated at 350W, either of which is capable of running the system alone.

# APPENDIX B: Migrating objects from KeySecure Classic

This chapter describes how to migrate objects from a SafeNet KeySecure Classic system to the NextGen KeySecure system.

This chapter contains the following topics:

["Overview" below](#)

["Identifying objects to be Backed-up" on the next page](#)

["Creating a Backup File" on page 145](#)

["Migrating the Backup File" on page 149](#)

## Overview

Migration from SafeNet KeySecure Classic is performed by creating a backup file in KeySecure Classic and importing that file into the NextGen KeySecure. You must first decide what data on KeySecure Classic you wish to migrate. You then create queries to export only the data you wish to migrate. If desired, you can also backup and migrate all data.



**NOTE** In NextGen KeySecure release 1.5, supported migration data include keys, local users and groups, as well as LDAP server users and groups.

After all data has been selected for migration, use KeySecure Classic's backup process to create data sets, which are typically downloaded to the user's computer. Once the backup file has been downloaded it serves as input to the migration operations available using CLI tool **ksctl**, where it is first uploaded and then processed. After the data has been processed it can be queried using **ksctl** to verify that all data was correctly migrated.

## User and Group Migration

When migrating a KeySecure Classic backup that has an LDAP configuration, only the LDAP configuration is migrated while all local users and groups are ignored.

There is no complete storage of LDAP users and groups on a KeySecure Classic system. Because of this, LDAP users and groups are created in the NextGen KeySecure system as the keys are parsed from the backup file. This prevents issues with orphan keys – keys that cannot be accessed because there is no user.

The NextGen KeySecure system maintains a mapping between groups stored in the keys and local groups created in the NextGen KeySecure system. For further details on LDAP Group Mapping, go to Admin Guide > Concepts > LDAP Group Mapping.

## Identifying objects to be Backed-up

We begin by identifying the menu objects we intend to backup in KeySecure Classic .

### To view Users and Groups

1. In the Management Console, select the **Security** tab at the top of the page.
2. In the left pane, under **Users & Groups**, navigate to **Local Authentication > Local Users and Groups**.

The Users and Groups screen shows (up to) 10 local users and (up to) 10 local groups.



**NOTE** The following message may also appear on the screen if LDAP is being used. This will not have effect on backing up your data.



**Note:** NAE server is configured to use LDAP user directory. Local users will not be recognized.

Security > Local Authentication > Local Users & Groups

#### User & Group Configuration

| Local Users                                       |                  |                                     |                                     | Help ? |
|---------------------------------------------------|------------------|-------------------------------------|-------------------------------------|--------|
| Filtered by: ---- where value contains Set Filter |                  |                                     |                                     |        |
| Items per page: 10 Submit                         | Page 1 of 101 Go | Next >                              |                                     |        |
| Username                                          | Password         | User Administration Permission      | Change Password Permission          |        |
| admin                                             | *****            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |        |
| admin                                             | *****            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |        |
| admin                                             | *****            | <input type="checkbox"/>            | <input type="checkbox"/>            |        |
| admin                                             | *****            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |        |
| admin                                             | *****            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |        |
| admin                                             | *****            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |        |
| admin                                             | *****            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |        |
| admin                                             | *****            | <input type="checkbox"/>            | <input type="checkbox"/>            |        |
| admin                                             | *****            | <input type="checkbox"/>            | <input type="checkbox"/>            |        |
| admin                                             | *****            | <input type="checkbox"/>            | <input type="checkbox"/>            |        |
| 1 - 10 of 1006                                    |                  |                                     |                                     | Next > |



**Note:** NAE server is configured to use LDAP user directory. Local users will not be recognized.

Edit Add Delete Properties

Local Groups

Help ?

Filtered by

----

where value

contains

Set Filter

Items per page: 10 

Submit

↑ Group

● KeyUsers

● Tester

1 - 2 of 2

Add

Delete

Properties

---

### To view the LDAP Server Configuration

1. In the Management Console, select the **Security** tab at the top of the page.
2. In left pane, under **Users and Groups**, navigate to **LDAP > LDAP Server**.

The **LDAP Server Configuration** screen is displayed showing the LDAP configuration on this machine



**NOTE** Since this KeySecure Classic system is configured to use LDAP, only LDAP users will be migrated.

---

### To view the objects to be backed up.

1. In the Management Console, select the **Security** tab at the top of the page.
2. In left pane under **Managed Objects**, navigate to **Keys > Key List**.

The **Key List** screen is displayed showing (up to) 10 keys that are stored in the platform, with a default **Query** of **[All]**.

A new query can be created to select a subset of the keys in the platform. This is initiated by selecting **Create Query** at the bottom of this screen.

If a subset is selected, make sure that all keys to be migrated are included in one of the queries.



**NOTE** If multiple queries are defined, be sure that no keys match multiple queries. The migration will reject any key that is duplicated. This will not fail the migration but it will create error messages that must be investigated.

For more information about the creation and use of key queries, refer the KeySecure Classic **Help** documentation, **Administration Guide > Key Management > Creating a Key Query**,

---

## Creating a Backup File

All backup operations are done within KeySecure Classic. Familiarity with its user interface is assumed.

For more information about Creating a Backup, refer the KeySecure Classic **Help** documentation, **Administration Guide > Backup and Restore > Creating a Backup**.

---

### To create a backup on KeySecure Classic:

1. Log in to the Management Console as an administrator with the appropriate backup access control. There are specific access controls for backing up configuration, keys & certificates, and local CAs.
2. Select the **Device** tab at the top of the page.
3. In left pane, under **Maintenance**, navigate to **Backup & Restore > Create Backup**.



**NOTE** This is the first of three **Create Backup** screens.

## Backup and Restore

## Create Backup

| Security Items                                 |                                                                                                                                                                                                                                                              | → |
|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| Security Items:                                | <input type="button" value="Select All"/> <input type="button" value="Select None"/>                                                                                                                                                                         |   |
| ProtectDB Manager:                             | <input type="checkbox"/>                                                                                                                                                                                                                                     |   |
| ProtectFile Manager:                           | <input type="checkbox"/>                                                                                                                                                                                                                                     |   |
| Tokenization Manager:                          | <input type="checkbox"/>                                                                                                                                                                                                                                     |   |
| Managed Objects:                               | <input type="radio"/> All keys<br><input type="radio"/> No keys<br><input type="radio"/> One key: <input type="text"/><br><input checked="" type="radio"/> Choose from query: <input type="text" value="[All]"/> <input type="button" value="Show Results"/> |   |
| Managed Object Queries and Options:            | <input type="checkbox"/>                                                                                                                                                                                                                                     |   |
| Authorization Policies:                        | <input type="checkbox"/>                                                                                                                                                                                                                                     |   |
| Local Users & Groups:                          | <input checked="" type="checkbox"/>                                                                                                                                                                                                                          |   |
| LDAP Server for Users & Groups:                | <input checked="" type="checkbox"/>                                                                                                                                                                                                                          |   |
| Certificates:                                  | <input type="radio"/> All certificates<br><input checked="" type="radio"/> No certificates<br><input type="radio"/> Choose from list:<br><input type="text" value="ks-test"/>                                                                                |   |
| Local Certificate Authorities:                 | <input type="radio"/> All certificates<br><input checked="" type="radio"/> No certificates<br><input type="radio"/> Choose from list:<br><input type="text" value="ks-ca"/>                                                                                  |   |
| Known CAs, CRLs, and Trusted CA List Profiles: | <input type="checkbox"/>                                                                                                                                                                                                                                     |   |
| High Security:                                 | <input type="checkbox"/>                                                                                                                                                                                                                                     |   |
| FIPS Status Server:                            | <input type="checkbox"/>                                                                                                                                                                                                                                     |   |



**NOTE** For the purposes of migration, the only items of interest in release 1.5 of NextGen KeySecure system are Keys, Local Users & Groups as well as LDAP Server Users & Groups.

4. Select the configuration items to include in the backup file. Use the **Select All** button to select all items on the page.

When selecting Managed Objects, you have the option of selecting all keys, no keys, specific keys, or backing up the results of a query. You can view query results using the **Show Results** button. When selecting Certificates and Local Certificate Authorities, you can select all, none, or select items from a list. The **All keys** and **All certificates** options are dynamic. This means if you select these options, schedule an automated backup and subsequently create new keys and certificates. The new keys and certificates will be included in the next scheduled backup.





**NOTE** The Log Signing Certificate is not included with the other certificates on the device. To backup the Log Signing Certificate, you must specifically select it on the next screen by selecting **Continue**.

5. Select **Continue**. The second **Create Backup** screen is displayed.

[Device](#) » [Backup & Restore](#) » **Create Backup**

### Backup and Restore

**Create Backup**

Security Items →

| Device Items:            | Select All               | Select None |
|--------------------------|--------------------------|-------------|
| NTP:                     | <input type="checkbox"/> |             |
| Network:                 | <input type="checkbox"/> |             |
| IP Authorization:        | <input type="checkbox"/> |             |
| KeyStore                 | <input type="checkbox"/> |             |
| Lockout Config           | <input type="checkbox"/> |             |
| Session Autologout       | <input type="checkbox"/> |             |
| Administrators:          | <input type="checkbox"/> |             |
| SNMP:                    | <input type="checkbox"/> |             |
| Known Hosts:             | <input type="checkbox"/> |             |
| Log Rotation:            | <input type="checkbox"/> |             |
| Syslog:                  | <input type="checkbox"/> |             |
| SSL:                     | <input type="checkbox"/> |             |
| Key Server               | <input type="checkbox"/> |             |
| Services:                | <input type="checkbox"/> |             |
| Log Signing Certificate: | <input type="checkbox"/> |             |

Continue Back Cancel



**NOTE** None of these items should be selected for migration.

6. Select **Continue**. The third **Create Backup** screen is displayed.

|                                                                                                                                                                                              |                                                                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Backup Name:                                                                                                                                                                                 | <input type="text" value="migration-test"/>                                                                                                   |
| Backup Description:                                                                                                                                                                          | <input type="text" value="migration of users &amp; keys"/>                                                                                    |
| Backup Password:                                                                                                                                                                             | <input type="password" value="....."/>                                                                                                        |
| Confirm Backup Password:                                                                                                                                                                     | <input type="password" value="....."/>                                                                                                        |
| Destination:                                                                                                                                                                                 | <input type="radio"/> Internal<br><input checked="" type="radio"/> Download to browser<br><input type="radio"/> FTP <input type="radio"/> SCP |
| <input type="button" value="Backup Now"/> <input type="button" value="Save Settings for Automated Remote Backup"/> <input type="button" value="Back"/> <input type="button" value="Cancel"/> |                                                                                                                                               |

7. Enter backup file information: name, description, and password:

- a. **Backup Name** - Enter a name for the backup. Use only letters and numbers. Special characters and white space are not allowed in backup names. For backups stored externally, the backup filename is created by appending `_0_bkp` to this name. For large backups, the zero is incremented by 1 for each additional file. For example, backup foo could consist of two files: `foo_0_bkp` and `foo_1_bkp`.
- b. **Backup Description** - Enter a short description for the backup.
- c. **Backup Password** - Enter a password for your backup file. Remember, this file contains very valuable information and will likely have a long life span. Use an appropriately complex password. The password can have a maximum of 30 characters.



**\*\*WARNING\*\*** The backup file cannot be restored without this password.

8. Enter the **Destination** of the file.

The backup configuration can be stored internally on the KeySecure Classic, downloaded to a browser or copied to another machine via FTP or SCP. To use SCP with a new host, first add the destination host to the Known Hosts list, as described in the KeySecure Classic **Help** documentation **Administration Guide > Known Hosts**.

If you download the backup configuration to a browser, the backup configuration is encrypted and downloaded to your local machine. You must specify a name for the file; however, it is not necessary to specify an extension for the file.

If you select FTP or SCP to copy the backup configuration to another machine, you must provide the following:

- the destination host.
- the name of the directory on the destination host. (You must have write permission for this directory.)
- the username of the account on the destination host.
- (FTP) the password for the user account on the destination host.
- (SCP only) The authentication method, Key or Password. To use Key authentication with a new remote host, first add the SafeNet KeySecure's SSH Key to the destination host, as described in the KeySecure Classic **Help** documentation **Administration Guide > SSH Public Key**.

9. Before proceeding, review the **Backup Summary** listing at the bottom of the third **Create Backup** screen. If you need to make changes, use the Back button to return to appropriate screen and make the necessary edits. After making changes, you may have to re-enter data in the third **Create Backup** screen.
10. If you need to complete a one-time backup immediately, select **Backup Now** to create and store the backup file, then click **Save**. Confirm your save when requested to do so.



**NOTE** The backup you have defined is started immediately when you click **Backup Now**. The detailed settings used by Backup Now processing are not preserved, so you may wish to keep a record of these settings yourself.

It is possible to create multiple backups and migrate each of them separately. The only issue is that duplicate data (keys, users, configuration) will display error messages to that effect. If you run the same file multiple times without changes on the NextGen KeySecure system, duplicate data error messages will also occur. There is no harm in doing so. If changes are made to the NextGen KeySecure system between migration attempts (for example, deleting records that were migrated) then those 'duplicate data' errors will not occur.

## Migrating the Backup File

---

The CLI tool **ksctl** is used to perform migrations. If needed, refer to ["CLI" on page 1](#).

To get help with **ksctl** migration commands, type the following:

```
$ ksctl migrations --help
```

The following **ksctl** commands are used for migration:

|        |                                                                        |
|--------|------------------------------------------------------------------------|
| upload | Upload a 'KeySecure Classic' backup file.                              |
| apply  | Migrate the specified 'KeySecure Classic' backup file to a KeySecure.  |
| list   | List available 'KeySecure Classic' backup files.                       |
| get    | Get information about a specific 'KeySecure Classic' backup file.      |
| status | Get the status of the ongoing (or last) migration.                     |
| delete | Delete a 'KeySecure Classic' backup file.                              |
| log    | Get the migration log of the ongoing or last migration of this backup. |

---

### To migrate your backup file to the NextGen KeySecure system:

1. **UPLOAD:** The first step in a migration will be to upload a KeySecure Classic backup file. This process is done with the **ksctl** upload command:

```
$ ksctl migrations upload --file <ks-classic-backup-file>
```

The following flags can be set for the users convenience:

```
--desc <backup description>
```

A description of the backup can be stored

```
--checksum <sha256-sum>
```

If the user computes a checksum of the backup, enter that value here. The checksum of the file will be returned when the file upload is complete, these numbers must match for the upload to be successful

```
--chunk-size 10000000
```

To expedite uploads, this will set the chunksize used for data transfers from the client to the host

```
--upload-id <upload-id>
```

Used when resuming chunked uploads. May be specified when starting a chunked upload.

When the upload is complete a response is received:

```
{
 "id": "36828f00-fad8-4dc2-9854-44ed1dfdfb2b",
 "file_size": 55751,
 "created_at": "2018-09-19T21:24:20.196984173Z",
 "checksum_sha256": "69fdf386bacb9ab322db15acdc25cd4ab2b68c2c1a13b128d2225589db208638"
}
```



**NOTE** The value of the id field is used in many of the following steps

2. **APPLY:** After the file has been uploaded, the user must start the migration process, since it does not start automatically. To start the process use the apply command:

```
$ ksctl migrations apply --id <id> --migpassword <password used in backup>
```

If LDAP configuration information is included in the backup, then this should be indicated on the command line with the `--e` and `--n` flags

```
$ ksctl apply -i <id> -p <backup password> -e ldap -n <ldap conn>
```



**NOTE** Either local or LDAP users can be migrated in a single `apply` command. Also, as detailed earlier, local users are invisible on a system configured for LDAP usage.

3. **STATUS:** The status of the last migration can be queried with the status command:

```
$ ksctl migrations status
```

Response:

```
{
 "id": "878a711e-1dcb-41f6-8cb1-fa0cd2105070",
 "overall_status": "Completed",
 "users_status": {
 "status": "Completed",
 "num_processed": 0,
 "num_failed": 0,
 "num_ignored": 0
 },
}
```

```

"groups_status": {
 "status": "Completed",
 "num_processed": 0,
 "num_failed": 0,
 "num_ignored": 0
},
"groups_members_status": {
 "status": "Completed",
 "num_processed": 0,
 "num_failed": 0,
 "num_ignored": 0
},
"keys_status": {
 "status": "Failed",
 "num_processed": 16,
 "num_failed": 0,
 "num_ignored": 0
},
"keys_links_status": {
 "status": "Completed",
 "num_processed": 0,
 "num_failed": 0,
 "num_ignored": 0
},
"ldap_server_status": {
 "status": "Completed",
 "num_processed": 0,
 "num_failed": 0,
 "num_ignored": 1
}
}

```

If errors are indicated in the status above, then the **ksctl** `migrations log` command should be run (see below).

#### 4. LIST: To get list of migration files loaded on the platform, use the list command:

```
$ ksctl migrations list
```

This will return information about all of the migrations that are stored on the platform as follows:

```

{
 "skip": 0,
 "limit": 10,
 "total": 2,

```

```

 "resources": [
 {
 "id": "36828f00-fad8-4dc2-9854-44ed1dfdfb2b",
 "file_size": 55751,
 "created_at": "2018-09-19T21:24:20.196984173Z",
 "checksum_sha256": "69fdf386bacb9ab322db15acdc25cd4ab2b68c2c1a13b128d2225589db208638"
 },
 {
 "id": "878a711e-1dcb-41f6-8cb1-fa0cd2105070",
 "file_size": 55751,
 "created_at": "2018-09-12T19:24:43.237843362Z",
 "checksum_sha256": "c7dcf3229e3b28edd3fa763b07b756009face6277cee7d80fe4cbad891f"
 }
]
 }
}

```

There are two command line options that can be useful if a large number of migrations are stored on the platform, those options are `--limit` and `--skip`

The purpose of `--limit` is to let **ksctl** know how many records (maximum) should be retrieved and displayed in the response. The purpose of `--skip` is to let **ksctl** know how many records to ignore in the platform before returning responses. For example, if there are 100 migrations stored and you wish to only retrieve information about records 20-50, type the following command:

```
$ ksctl list --limit 30 --skip 20
```

#### 5. GET: To retrieve information about a single migration, use the `get` command:

```
$ ksctl migrations get --id <id>
```

where `<id>` is the migration ID returned in the upload of the backup. When this command is run, the response is:

```

{
 "id": "36828f00-fad8-4dc2-9854-44ed1dfdfb2b",
 "file_size": 55751,
 "created_at": "2018-09-19T21:24:20.196984173Z",
 "checksum_sha256": "69fdf386bacb9ab322db15acdc25cd4ab2b68c2c1a13b128d2225589db208638"
}

```

#### 6. LOG: If the user wishes to retrieve the log created by the NextGen KeySecure system, use the `log` command:

```
$ ksctl migrations log --id <id>
```

This will return a log listing each user or key for which there was an attempt at migration. The success or failure of that attempt will be detailed. This is the command to run where there are errors reported in the `status` command above. The error messages should be explanatory in regards to the problem with the migration. If a resolution cannot be determined, contact Gemalto Support for the NextGen KeySecure system. For contact information, refer to ["Support Contacts" on page 13](#).

### Example log file messages:

```
2018-09-21 15:15:40 | migration start | success | 878a711e-1dcb-41f6-8cb1-fa0cd2105070
2018-09-12 19:30:17 | migrate key | success | 'beta3-2327-aes256', version 0
2018-09-12 19:30:17 | migrate key | ignore | 'beta3-2327-des', version 0 - unsupported key size
64 for TDES;
2018-09-12 19:30:17 | migrate key | fail | 'beta3-2327-desede', version 0 - The default IV size
must be 16
2018-09-12 19:30:17 | migrate key | success | 'beta3-2327-hmacsha1', version 0
2018-09-12 19:30:17 | migrate key | ignore | 'beta3-2327-rc4', version 0 - unsupported
algorithm 'RC4';
2018-09-12 19:30:17 | migrate key | success | 'beta3-2327-rsa', version 0
2018-09-12 19:30:17 | migrate key | ignore | 'beta3-2327-seed', version 0 - unsupported
algorithm 'SEED-CBC';
2018-09-12 19:30:17 | migration finished | success |
```



**\*\*WARNING\*\*** This file will get extremely long for large migrations.

7. DELETE: Finally, to remove a migration file from the NextGen KeySecure system, use the `delete` command:

```
$ ksctl migrations delete --id <id>
```

If the `<id>` value exists in the NextGen KeySecure system, there is no visible response.

If there are errors, **ksctl** will report them.

End of Document



# Thales CipherTrust Manager

---

## ADMINISTRATOR GUIDE





## Document Information

|                        |                   |
|------------------------|-------------------|
| <b>Product Version</b> | 2.0.0             |
| <b>Release Date</b>    | 20 September 2020 |

## Trademarks and Copyrights

Copyright © 2020 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.

## Disclaimer

All information herein is either public information or is the property of and owned solely by Thales and/or its subsidiaries 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, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- > This document shall not be posted on any publicly accessible 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.

# CONTENTS

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| Overview .....                                                                | 8  |
| Product references made in this document .....                                | 8  |
| Audience .....                                                                | 8  |
| Key Management .....                                                          | 8  |
| High Performance .....                                                        | 8  |
| Broad Flexibility .....                                                       | 9  |
| Robust Security .....                                                         | 9  |
| Release Notes .....                                                           | 9  |
| Document Conventions .....                                                    | 9  |
| Notes .....                                                                   | 9  |
| Cautions .....                                                                | 10 |
| Warnings .....                                                                | 10 |
| Command Syntax and Typeface Conventions .....                                 | 10 |
| Support Contacts .....                                                        | 11 |
| Customer Support Portal .....                                                 | 11 |
| Telephone Support .....                                                       | 11 |
| Email Support .....                                                           | 11 |
| Troubleshooting Issues in Conjunction with Customer Support .....             | 12 |
| Concepts .....                                                                | 13 |
| Password Policy .....                                                         | 14 |
| Changing allowed password length and characters .....                         | 14 |
| Changing Password Expiration .....                                            | 14 |
| Changing user account lockout thresholds .....                                | 15 |
| Unlocking a user account .....                                                | 15 |
| Changing the password history .....                                           | 15 |
| Users .....                                                                   | 16 |
| Managing Users .....                                                          | 16 |
| Configuring LDAP Authentication .....                                         | 17 |
| Certificate based Authentication .....                                        | 18 |
| Changing Passwords .....                                                      | 20 |
| Changing the Application Administrator's initial password using the GUI ..... | 21 |
| Changing user passwords .....                                                 | 21 |
| Groups .....                                                                  | 21 |
| Special System Users .....                                                    | 21 |
| System Defined Groups .....                                                   | 23 |
| User Defined Groups .....                                                     | 28 |
| LDAP Group Mapping .....                                                      | 29 |
| Overview .....                                                                | 29 |
| CLI Commands .....                                                            | 29 |
| Use Cases .....                                                               | 30 |

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| Making all Users in a Specific LDAP Group members of Key Users Group ..... | 30 |
| Two LDAP groups share keys .....                                           | 30 |
| Services .....                                                             | 30 |
| Services Reset .....                                                       | 31 |
| Services Restart .....                                                     | 31 |
| Services Status .....                                                      | 31 |
| Hardware Security Module .....                                             | 32 |
| Supported HSMs .....                                                       | 32 |
| Hybrid HSM .....                                                           | 33 |
| Setting up an HSM .....                                                    | 33 |
| Getting and Listing HSMs .....                                             | 36 |
| Adding an HSM .....                                                        | 37 |
| Deleting an HSM .....                                                      | 38 |
| Clusters and Nodes .....                                                   | 38 |
| What is not clustered? .....                                               | 39 |
| Managing Clusters and Nodes .....                                          | 40 |
| Cluster Upgrade .....                                                      | 43 |
| Tokens .....                                                               | 45 |
| API Tokens (JWT) .....                                                     | 45 |
| Refresh Tokens .....                                                       | 45 |
| Revoke Refresh Token .....                                                 | 47 |
| Login Command .....                                                        | 47 |
| Logout Command .....                                                       | 49 |
| Policies .....                                                             | 49 |
| Policy Composition .....                                                   | 50 |
| Policy Attachments .....                                                   | 50 |
| Default Policies .....                                                     | 50 |
| Usage .....                                                                | 51 |
| Records .....                                                              | 51 |
| Enabling NAE Crypto Audit Records .....                                    | 51 |
| Disabling local database audit logging .....                               | 52 |
| Configuring remote Syslog server .....                                     | 53 |
| Configuring alarm triggers based on an audit record .....                  | 53 |
| Syslogs .....                                                              | 55 |
| Configuring Connection to a Syslog Server .....                            | 56 |
| Modifying Connection to a Syslog Server .....                              | 56 |
| Banners .....                                                              | 57 |
| Viewing the Banners using the CLI .....                                    | 57 |
| Updating the Banners using the CLI .....                                   | 57 |
| Interfaces .....                                                           | 58 |
| Enabling/Disabling Interfaces .....                                        | 61 |
| Interface Default Connection (NAE and KMIP) .....                          | 62 |
| SSL/TLS Certificate (NAE, KMIP, and WEB) .....                             | 63 |
| Interface Mode (NAE, KMIP, and WEB) .....                                  | 64 |
| certUserField Interface (NAE, KMIP, and WEB) .....                         | 64 |
| Auto-generation of Server Certificate (NAE, KMIP, and WEB) .....           | 65 |
| Trusted CAs (NAE, KMIP, and WEB) .....                                     | 67 |

|                                                           |     |
|-----------------------------------------------------------|-----|
| Adding, Removing, and Modifying Interfaces (NAE and KMIP) | 67  |
| SNMP                                                      | 69  |
| SNMP Overview                                             | 69  |
| SNMP on CipherTrust Manager                               | 69  |
| SNMP v1/v2c                                               | 69  |
| SNMP v3                                                   | 70  |
| MIBs                                                      | 70  |
| Access Control                                            | 70  |
| Enterprise MIBs                                           | 70  |
| Notifications                                             | 71  |
| Configuring SNMPv1/v2c                                    | 71  |
| Configuring SNMPv3                                        | 72  |
| SMTP                                                      | 73  |
| Adding a New SMTP Server                                  | 73  |
| Adding/Deleting Emails in the Notification Address List   | 74  |
| Backups                                                   | 74  |
| System Backups                                            | 75  |
| Domain Scoped Backups                                     | 76  |
| Typical Backup and Restore Operations                     | 76  |
| Scheduling Backups                                        | 86  |
| Disk Encryption                                           | 87  |
| Scheduler                                                 | 89  |
| Time Specification                                        | 89  |
| Allowed Special Characters                                | 90  |
| Creating Job Configurations                               | 90  |
| Viewing Existing Job Configurations                       | 91  |
| Viewing Details of Job Configurations                     | 92  |
| Modifying Job Configurations                              | 92  |
| Manually Running Job Configurations                       | 93  |
| Deleting Job Configurations                               | 93  |
| Job Runs                                                  | 93  |
| Viewing Job Runs                                          | 93  |
| Viewing Details of a Particular Job Run                   | 95  |
| Deleting a Job                                            | 96  |
| Licensing                                                 | 96  |
| Lock Codes                                                | 96  |
| CipherTrust Manager Licenses                              | 96  |
| Connector Licenses                                        | 98  |
| License Enforcement                                       | 98  |
| Viewing Installed Licenses                                | 99  |
| Deleting Installed Licenses                               | 99  |
| System Logs                                               | 99  |
| Downloading Logs                                          | 100 |
| Verifying and viewing the downloaded logs                 | 100 |
| System Upgrade/Downgrade                                  | 101 |
| System Upgrade                                            | 101 |
| System Downgrade                                          | 102 |

---

|                                                          |            |
|----------------------------------------------------------|------------|
| Client Management .....                                  | 102        |
| Tokens .....                                             | 102        |
| Clients .....                                            | 104        |
| Connection Manager .....                                 | 104        |
| Accessing the Connection Manager .....                   | 104        |
| Adding a New Connection .....                            | 105        |
| Editing a Connection .....                               | 106        |
| Deleting a Connection .....                              | 107        |
| Configuring DNS Hosts .....                              | 107        |
| <b>Operation .....</b>                                   | <b>108</b> |
| CLI toolkit .....                                        | 109        |
| Installation .....                                       | 109        |
| Try it! .....                                            | 113        |
| Batching Commands with Tokens .....                      | 117        |
| Support CLI .....                                        | 119        |
| Keys .....                                               | 120        |
| Setting Key Permissions .....                            | 122        |
| Versioned Keys .....                                     | 123        |
| Key States (KMIP and REST) .....                         | 124        |
| Effects of Negative Actions on Keys .....                | 125        |
| Key States (NAE) .....                                   | 125        |
| Key Rotation .....                                       | 126        |
| Crypto Operations .....                                  | 127        |
| Symmetric Encryption .....                               | 127        |
| Asymmetric Encryption .....                              | 127        |
| Asymmetric Sign and Verify .....                         | 127        |
| MAC and MAC Verify .....                                 | 128        |
| Format Preserving Encryption .....                       | 128        |
| Certificate Authority .....                              | 129        |
| Creating a local CA .....                                | 129        |
| Issuing a signed certificate .....                       | 130        |
| Certificate Expiration Check .....                       | 130        |
| Creating an external CA .....                            | 130        |
| Verifying Revocation Status of Client Certificates ..... | 131        |
| OCSP and CRL Caching .....                               | 132        |
| Enabling/Disabling Certificate Revocation Check .....    | 133        |
| Alarms .....                                             | 133        |

---

# Overview

---

The Thales CipherTrust Manager is built on prevailing cloud-based technologies, providing a cloud friendly, key management solution. It employs a REST interface and a microservice based architecture, allowing for easy deployment and scalability within your environment. With this new architecture, the CipherTrust Manager simplifies administration, helps ensure compliance, and maximizes security by providing centralized management of keys, policies, and essential functions.

## Product references made in this document

- > Thales CipherTrust Manager - refers to the physical and virtual appliance that centralizes management, keys, and policies for the CipherTrust Data Security Platform.
- > Thales CipherTrust Manager k170v (or simply k170v) - refers specifically to virtual and private cloud instances running the Thales CipherTrust Manager Software.
- > Thales CipherTrust Manager k470 and k570 Appliances (or simply k470 and k570) - refers specifically to the k470 and k570 physical appliances running the Thales CipherTrust Manager Software.
- > SafeNet KeySecure Classic (or simply KeySecure Classic) - refers to the current generation of SafeNet KeySecure Appliances (models k450 and k460). These appliances can be upgraded to the Thales CipherTrust Manager Software.

## 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.

It is assumed that users of this document are proficient with security concepts.

## Key Management

---

Thales CipherTrust Manager offers robust capabilities for managing cryptographic keys across their lifecycle, including key generation, key import and export, and key rotation. With the CipherTrust Manager, all cryptographic keys are stored in a centralized, hardened appliance to simplify administration while ensuring tight security for the broadest array of data types.

## High Performance

---

Customers can deploy multiple CipherTrust Manager appliances in a clustered configuration with real-time replication of keys, policies, and configuration information across multiple appliances - enabling complete disaster recovery and business continuity. For large distributed enterprises that use multiple encryption

solutions, keys can be centrally managed without making any perceptible impact on system performance.

## Broad Flexibility

---

Thales CipherTrust Manager offers key management capabilities that can be integrated with virtually any commercial encryption product. Supported technologies include:

- > SafeNet Luna Network HSM and AWS HSM partitions
- > Application encryption
- > Database encryption, including native database encryption
- > File and storage level encryption solutions

The CipherTrust Manager supports a wide range of open standard cryptographic interfaces, including PKCS #11, JCE, and .NET. The CipherTrust Manager also supports the Key Management Interoperability Protocol (KMIP). Further, customers and partners can take advantage of SafeNet's REST interface to develop their own custom software utilizing the enterprise key management functionality of the CipherTrust Manager.

## Robust Security

---

Thales CipherTrust Manager offers a range of robust security features:

- > Granular Attribute Based Access Control (ABAC) authorization capabilities
- > Secure key distribution through support of SSL
- > Secure storage of key encryption keys on a SafeNet Luna Network HSM

## Release Notes

---

The customer release notes (CRN) provide important information about this release that is not included in the customer documentation. It is strongly recommended that you read the CRN to fully understand the capabilities, limitations, and known issues for this release. You can view or download the latest version of the CRN from the Technical Support Customer Portal at <https://supportportal.thalesgroup.com>.

## Document Conventions

---

This document uses standard conventions for describing the user interface and for alerting you to important information.

### Notes

Notes are used to alert you to important or helpful information. They use the following format:

**NOTE** Take note. Contains important or helpful information.



## Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss. They use the following format:

**CAUTION!** Exercise caution. Contains 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. They 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

| Format                     | Convention                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>bold</b>                | The bold attribute is used to indicate the following: <ul style="list-style-type: none"><li>&gt; Button names (Click Save As.)</li><li>&gt; Check box and radio button names (Select the Print Duplex check box.)</li><li>&gt; Dialog box titles (On the Protect Document dialog box, click Yes.)</li><li>&gt; Field names (User Name: Enter the name of the user.)</li><li>&gt; Menu names (On the File menu, click Save.) (Click Menu &gt; Go To &gt; Folders.)</li><li>&gt; User input (In the Date box, type April 1.)</li></ul> |
| mono-spaced                | Command-line commands and options (Type dir /p.)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <i>italics</i>             | In type, the italic attribute is used for emphasis or to indicate a related document. (See the <i>Installation Guide</i> for more information.)                                                                                                                                                                                                                                                                                                                                                                                      |
| <variable>                 | In command descriptions, angle brackets represent variables. You must substitute a value for command line arguments that are enclosed in angle brackets.                                                                                                                                                                                                                                                                                                                                                                             |
| [optional]<br>[<optional>] | Represent optional <b>keywords</b> 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.                                                                                                                                                                                                                                                                                                               |
| {a b c}<br>{<a> <b> <c>}   | Represent required alternate <b>keywords</b> 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.                                                                                                                                                                                                                                                                                                                        |

| Format                   | Convention                                                                                                                                                                                              |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [a b c]<br>[<a> <b> <c>] | Represent 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. |

## 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](mailto: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.

# Concepts

Thales CipherTrust Manager's NAE-XML interface is an adapter layer that translates XML requests into Thales CipherTrust Manager's REST API. However, the REST API is structured differently than NAE-XML. The functions and configuration of Thales CipherTrust Manager is modeled as a set of resources that can be managed via the REST API. The following sections describe those abstract resource models. The REST API online documentation goes into more detail about the specific attributes of the resources, and what operations you can perform on them.

**NOTE** The CipherTrust Manager also includes a CLI toolkit, named **ksctl**, that can be downloaded and run locally to perform management functions. **ksctl** exclusively uses the REST API to communicate with the CipherTrust Manager. Many examples used in the chapter make use of **ksctl**. For more information on **ksctl**, go to ["CLI toolkit" on page 109](#)

This chapter describes in detail concepts relevant to the administration of the Thales CipherTrust Manager. It contains the following sections:

- > ["Password Policy" on the next page](#)
- > ["Users" on page 16](#)
- > ["Changing Passwords" on page 20](#)
- > ["Groups" on page 21](#)
- > ["LDAP Group Mapping" on page 29](#)
- > ["Services" on page 30](#)
- > ["Hardware Security Module" on page 32](#)
- > ["Clusters and Nodes" on page 38](#)
- > ["Tokens" on page 45](#)
- > ["Policies" on page 49](#)
- > ["Records" on page 51](#)
- > ["Banners" on page 57](#)
- > ["Interfaces" on page 58](#)
- > ["SNMP" on page 69](#)
- > ["Backups" on page 74](#)
- > ["Disk Encryption" on page 87](#)
- > ["Scheduler" on page 89](#)
- > ["System Upgrade/Downgrade" on page 101](#)
- > ["Client Management" on page 102](#)

## Password Policy

There is one universal password policy for all users. You can configure the password policy using API playground or CLI.

This section reviews the password policies and provides CLI examples. For details on each policy, review the online "[CLI Documentation](#)".

### Changing allowed password length and characters

These are the password policy settings that define the allowed password length and characters:

Default: Minimum length: 8

Default: Maximum length: 30

Default: Minimum number of upper cases: 1

Default: Minimum number of lower cases: 1

Default: Minimum number of digits: 1

Default: Minimum number of other characters: 1

#### To set the entire password policy

You can set the entire password policy. See the "[CLI Documentation](#)" for details on each parameter.

```
-t, --minlength minimum length of the password
-z, --maxlength maximum length of the password
-m, --minupper minimum number of upper case letters
-w, --minlower minimum number of lower case letters
-d, --mindig minimum number of digits
-o, --minother minimum number of other characters
```

To set the allowed password length and characters in one string:

```
$ ksctl users pwdpolicy update --minlength 8 --maxlength 30 --minupper 1 --minlower 1 --minother 1
--mindig 1
```

#### To update a single aspect of the password policy

```
$ ksctl users pwdpolicy update --maxlength 100
```

## Changing Password Expiration

User passwords can be set to expire after the configured number of days from the last password change.

Default: Password lifetime: 0 (password never expires)

#### To set password to expire in 30 days

```
-L, --lifetime maximum lifetime of the user password.
$ ksctl users pwdpolicy update --lifetime 30
```

#### To set password to never expire:

```
$ ksctl users pwdpolicy update --lifetime 0
```

## Changing user account lockout thresholds

User's account can be temporarily locked out for a specified duration after a specified number of user authentication failure attempts. It is applicable only for users in a local account, i.e. not valid for LDAP users.

Default: disabled (no lockout)

### To set the user account lockout thresholds

`-f, --failed-logins-lockout-thresholds` List of lockout durations in minutes for failed login attempts.

In this example, values `[0, 5, 30]` means that the first failed login attempt, with lockout duration of zero, will not lockout the user account. The second failed login attempt will lockout the account for 5 minutes. The third and subsequent failed login attempts will lockout for 30 minutes.

```
$ ksctl users pwdpolicy update -f [0, 5, 30]
```

### To disable user account lockout

To disable user account lockout, set an empty array `[]`.

```
$ ksctl users pwdpolicy update -f []
```

## Unlocking a user account

A user account that has been locked due to failed log in attempts can be unlocked by an Application Administrator.

### To unlock a user account

```
$ ksctl users modify --id "local|c9161a90-0838-469b-87e9-726d8c539f3f" -u
```

## Changing the password history

The CipherTrust Manager retains the user's password history to prevent users from reusing their passwords.

Default: Password History: 5.

- > The minimum value for history is '0'. This value prevents the users from reusing their current password.
- > With the default value of 5, the user is prevented from reusing their current password and 4 previous passwords.
- > The maximum value for password history is 10.

### To set the password history to the value 7

`-b, --history` number of past passwords saved. This sets how frequently old passwords can be reused

To set the password history to the value 7, issue this command:

```
$ ksctl users pwdpolicy update --history 7
```

## Users

Users are the authenticated entities that make KMIP, REST or NAE-XML calls. To authenticate to the NAE-XML interface, you send an AuthRequest with the user credentials. To authenticate to the REST API, you POST the user credentials to an authentication endpoint, and receive an API token, which you can use in subsequent API calls.

### Managing Users

Users can be created and managed via the REST API and the CLI tool. The legacy NAE-XML requests for user management are also supported, through the NAE-XML interface.

On launch, a single user exists, named "admin". This user can create, modify and delete other users. The user that is created in the `ksctl` setup is this admin user; see ["Installation" on page 109](#)

The following options are available for managing users:

- > Create
- > Delete
- > Get a single record
- > List records
- > Modify

Note that when you create a new user there is a 'username' that is set. It is used as the login of that user and cannot be changed. Another field is 'user' that is the full name of the use; this is an optional field and can be modified. Finally, there is also a 'User ID' field created by the platform that is used for all updates to the user. This field is of the form: local|<UUID>

#### To create a user named "bob":

```
$ ksctl users create --name bob --pword bob
```

The response looks like:

```
{
 "created_at": "2018-04-27T21:10:53.191577Z",
 "email": "",
 "last_login": null,
 "logins_count": 0,
 "name": "bob",
 "nickname": "bob",
 "updated_at": "2018-04-27T21:10:53.191577Z",
 "user_id": "local|c7560d06-3628-4fb0-81e3-5881d2f8d7b3",
 "username": "bob",
 "failed_logins_count": 0,
 "account_lockout_at": null,
 "failed_logins_initial_attempt_at": null,
 "last_failed_login_at": null,
 "password_changed_at": "2018-04-27T21:10:53.192083Z",
 "password_change_required": false
}
```

## Configuring LDAP Authentication

The Thales CipherTrust Manager authenticates users against its internal database and also provides an option to authenticate with an external authentication provider (LDAP). It is possible to have a mix of both local and LDAP users.

### Overview

When LDAP is configured the users and credentials are located in an external LDAP server and Thales CipherTrust Manager acts as a forwarder to the LDAP server. The first time a new LDAP user is successfully authenticated, a corresponding user record is stored in the CipherTrust Manager and is marked to belong to the respective LDAP connection. All CipherTrust Manager user records associated with a LDAP connection are deleted when the LDAP connection is deleted from the CipherTrust Manager.

To authenticate users against LDAP, it is required to configure an LDAP connection in the CipherTrust Manager. Only CipherTrust Manager users with administrator privilege are authorized to configure LDAP connections. It is possible to configure LDAP connections through either the REST API, the CLI or the GUI.

Each LDAP connection is given a name that needs to be known by the end user and be provided during authentication; this is often referred to as the domain. For example, if a user **joe** exists in the LDAP directory `ldap://192.168.0.1:389` and this LDAP server is created in CipherTrust Manager with the name **myco**, the end user would have to authenticate with either **myco\joe**, **myco/joe** or **myco|joe** in the GUI. On the CLI, you would instead specify the name of the connection as a separate parameter.

LDAP authentication also works through NAE-XML. 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, LDAP 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>**.

### Setting Up a New Connection

Using either the CLI or the GUI it is possible to test a connection, create (bind) a new connection, get a list of connections, modify a connection and delete a connection.

Each LDAP connection has the following required fields:

- > `name` - Unique identifier for the connection and commonly referred to as the domain, for example **myco**.
- > `server_url` - Hostname and port of the LDAP server, for example `ldap://192.168.0.1:389`.
- > `root_dn` - The base DN as the starting point for the search, for example `DC=myco,DC=local`.
- > `strategy` - Type of connection to create, must be "ldap" in the case of an LDAP connection.
- > `uid_field` - The field containing the user's login name, for example `uid`.

Additionally, each LDAP connection has the following optional fields:



- > `search_filter` - Optional LDAP search filter which can further restrict the set of users who will be allowed to log in, for example `memberOf="CN=example,OU=Users,DC=myco,DC=local"`.
- > `guid_field` - The LDAP field name of a globally unique identifier for the user, for example `objectGUID`. If this field is specified, the user record in the CipherTrust Manager would have the value of this field as identifier of the user instead of the provided username. If for example the user **joe**, on the LDAP connection **myco**, has the objectGUID value `3450be9a6ee76540956e7c6b10d98831`, then his user record would be `myco|3450be9a6ee76540956e7c6b10d98831` instead of **myco|joe**. The friendly name **joe** would still be in the user record and be search-able.
- > `bind_dn` - If authentication is required to perform search this is the name of the entity to bind to, for example `CN=poweruser,CN=Users,DC=myco,DC=local`.
- > `bind_pass` - Password of the user provided in `bind_dn`.  
If `bind_dn` and `bind_pass` fields are not provided, the LDAP search will be performed using the 'end user provided' user id and password.
- > `group_id_field` - The field that contains the group name, for example 'cn'. If `group_id_field` is not provided, it will default to whatever is in `uid_field`.
- > `group_filter` - Search filter for listing groups. If `group_filter` is not provided, it will default to "(objectclass=Group)".
- > `group_member_field` - The field which contains a user's group membership information, for example 'member'. If `group_member_field` is not provided, it will default to 'member'.

Once a connection is created, it is not possible to modify these fields: `name`, `strategy` or `bind_pass`.

## CLI command example

Test the connection configuration to verify it can authenticate one of the LDAP users.

```
$ ksctl connections test \
 --test-username=jdoe --test-password=ji32k7au4a83 \
 --name=myco --strategy=ldap \
 --bind-password=admin --bind-dn="CN=poweruser,CN=Users,DC=myco,DC=local" \
 --server-url="ldap://192.168.0.1:389" --user-id-field=uid \
 --root-dn="DC=myco,DC=local" \
 --search-filter="(memberOf=CN=example,OU=Users,DC=myco,DC=local)"
```

Bind the connection configuration.

```
$ ksctl connections bind \
 --name=myco --strategy=ldap \
 --bind-password=admin --bind-dn="CN=poweruser,CN=Users,DC=myco,DC=local" \
 --server-url="ldap://192.168.0.1:389" --user-id-field=uid \
 --root-dn="DC=myco,DC=local" \
 --search-filter="(memberOf=CN=example,OU=Users,DC=myco,DC=local)"
```

## Certificate based Authentication

The CipherTrust Manager authenticates a user's login request by verifying the username and password against its internal database. The CipherTrust Manager can also be configured to authenticate login requests using browser-based web certificates. This section elaborates upon the steps that you need to perform to enable Certificate based Authentication for logging in the CipherTrust Manager.

## Step 1: Enable the "Certificate based Login" Option for a User

1. Log on to CipherTrust Manager as an administrator. Navigate to **Keys & Access management > Users**.
2. Enable the "Certificate based Login" option for the user:

- **For existing users:**

- i. Click the action button for that user, then click **Manage**.
- ii. Click **CONFIGURE CERTIFICATE LOGIN**. Select **Allow user to login using certificate**.
- iii. Specify **Certificate Subject Distinguished Name** for the user.

### CAUTION!

Specifying a Common Name (CN) is a mandatory for this feature to work. The entities must be specified carefully in this field, and separated by commas (,).

For example: O=Thales, OU=CipherTrust, CN=User\_1

- iv. Click **Update Certificate Login**.

- **For new users:**

- i. Click **Create New User**. Specify **Username** and **Password** for the user.
- ii. Select **Allow user to login using certificate**.
- iii. Specify **Certificate Subject Distinguished Name** for the user.

### CAUTION!

Specifying a Common Name (CN) is a mandatory for this feature to work. The entities must be specified carefully in this field, and separated by commas (,).

For example: O=Thales, OU=CipherTrust, CN=User\_1

- iv. Click **Create**.

## Step 2: Create and Download the Web Certificate

### If using Local CA

1. Go to **Keys & Access Management > CA**.
2. Click **Local Certificate Authority**, and then click **Create New Certificate**.
  - a. Enter the Common Name for this certificate.

**NOTE** This common name should be the same common name that was specified while creating the user ("User\_1" in previous example).

- b. Select desired algorithm (RSA or ECDSA).
- c. In the Name field, specify the same details that were specified in the "certificate\_subject\_dn" property of the user.
- d. Click **New Certificate**, then click **Save Private Key**. The **Save As** window opens up.

- e. Save the key (*.pem* file) in a secure location on your system.
  - f. Click **Issue Certificate**. The newly created certificate is now displayed in the certificate list.
3. Download this certificate, and save it in the same location where the Private Key is saved.

### If using External CA

1. Upload the external CA.
2. Navigate to **Admin Settings > System > Interface**. The Interface Configuration page is displayed.
3. Click the action button on the Web Interface Configuration, select **Edit**.
4. Add the new external CA in the **External Trusted CAs** section. Click **Update**.
5. Navigate to **Admin Settings > System > Services**.
6. Restart the web service.

**NOTE** Restarting the web service can take few seconds.

### Step 3: Create and Install pkcs12 Formatted Certificate

1. Install OpenSSL on your machine.
2. Use the following command to convert the key and certificate into a pkcs12 formatted *.pfx* file:

```
openssl pkcs12 -export -out example.pfx -inkey key.pem -in certificate.pem
```

Where:

- **key.pem** is the private key
- **certificate.pem** is the certificate file
- **example.pfx** is the pkcs12 formatted web certificate that will be installed in the web browser

This creates a *.pfx* certificate (*example.pfx* in the above command) at the same location.

3. Go to the web-browser's settings.
4. Import and install the *.pfx* certificate.

You can now use the web certificate for logging on to CipherTrust Manager. Before logging on, you will be prompted to select the web certificate at the login page.

**NOTE** After performing the above steps, if you are not prompted by the browser to select the certificate at login page, then clear your browser's cookies and try again.

## Changing Passwords

This section describes how to change the Application Administrator's initial password and the more general case of changing the password for a user.

**NOTE** Passwords must meet the current Password Policy settings. For more information on Password Policy and on changing the settings, see ["Password Policy" on page 14](#)

## Changing the Application Administrator's initial password using the GUI

At first boot, the default Application Administrator is "admin" with the password "admin", which must be changed before using the Thales CipherTrust Manager.

### To change the Application Administrators initial password using the GUI:

Open the URL: `https://{k170v ip address}`

1. Login as user "admin" using the default initial password "admin". If the password is not the hard-coded "admin", contact your System Administrator for an auto-generated password.
2. After successfully logging in, the user is prompted to change the initial password immediately. Failure to do so could allow an attacker to compromise the system.

For more information changing the password for the Initial User, see "Changing the Password for the Initial User" in section Thales CipherTrust Manager Deployment Guide > Private Cloud Image Deployment > Changing the Password for the Initial User .

## Changing user passwords

### To change a user password using the CLI:

```
$ ksctl changepw --user admin
```

User is prompted to enter and confirm the new password:

```
Please provide the new password of the user:
Please confirm the new password:
```

## Groups

A group carries with it permissions for performing specific tasks. A group also consists of a set of users and/or clients that have been authorized to perform these tasks.

The Thales CipherTrust Manager defines Special System Users, System Defined Groups, and User Defined Groups. This section consists of the following topics:

- > ["Special System Users" below](#)
- > ["System Defined Groups" on page 23](#)
- > ["User Defined Groups" on page 28](#)

**CAUTION!** It is critical that credentials of these users are kept in a secure location. If a credential is compromised, an attacker could gain access to sensitive data.

## Special System Users

There are a few special system users. These are described here:

## 'ksadmin' user

The "ksadmin" user, is a special System Administrator that can access the CipherTrust Manager via SSH or via password authentication on a physical server console port.

```
$ ssh ksadmin@<ip or hostname>
```

For public cloud providers, the SSH key used for authentication is the key used to launch the instance. For Private Cloud Images (e.g. VMware, Hyper-V), the SSH key must be replaced before the system will fully boot, which can be done via the CLI, API, or through a web browser.

The **ksadmin** user has permission to run a specific set of commands using **sudo**. These commands allow the user to troubleshoot problems on the CipherTrust Manager server, and perform upgrades. To see the list of commands that can be run with **sudo**, type the command **sudo -l**.

The duties of the System Administrator ("ksadmin" user) are:

- > Deploying and configuring the CipherTrust Manager, refer to:
  - Run cloud-init commands, refer to Deployment Guide > Cloud-init Configuration Options
  - Run **kscfg** commands, refer to Deployment Guide > System Configuration Utility (kscfg)
- > Retrieving the initial application admin user password, if configured to generate a unique password on first boot.
- > Troubleshooting CipherTrust Manager issues in conjunction with Thales Technical Support.
 

When making a support call, the System Administrator may be asked to log in to the CipherTrust Manager using ssh to retrieve CipherTrust Manager Logs.
- > Applying system upgrades (see ["System Upgrade/Downgrade" on page 101](#)).

## 'admin' user

Initially, there is only one Application Administrator and the name of this user is 'admin'. The 'admin' user is a special user who cannot be deleted and who will always have access to all resources. This is enforced by the policy engine to prevent accidental lockouts.

An 'admin' user, and all other Application Administrators if created, are responsible for:

- > Creating and managing Users and Groups
- > Configuring the CipherTrust Manager ports and licenses
- > Viewing audit logs
- > Managing backups
- > Administrating clusters

**NOTE** A user that is not in the "admin" group is known as an Application User. An Application User must also be part of the System Defined 'Key Users Group' to create and manage their keys. Refer to the ["'Key Users' group" on page 26](#)

The Application Administrator can also perform all duties of the Application User.

## 'Global' user

The 'Global' user exists to support specific NAE-XML functionality for compatibility with SafeNet KeySecure Classic and should not be deleted or modified.

## System Defined Groups

System Defined Groups exist on CipherTrust Manager at launch time. Each System Defined Group carries with it permissions to perform specific tasks. These groups are:

### 'admin' group

There is a System Defined Group named "admin". Users within the "admin" group are referred to as Application Administrators. Application Administrators have full privileges and are able to perform any operation via the REST API, CLI, NAE-XML or GUI interface.

**NOTE** Initially, there is only one Application Administrator and the name of this user is "admin". The "admin" user is a special user who cannot be deleted and who will always have access to all resources. This is enforced by the policy engine to prevent accidental lockouts.

An Application Administrator is responsible for:

- > Creating and managing Users and Groups
- > Configuring the CipherTrust Manager ports and licenses
- > Viewing audit logs
- > Managing backups
- > Administrating clusters

The Application Administrator can also perform all duties of the Application User.

### 'All Clients' group

A client, upon successful registration with CipherTrust Manager, is made a member of All Clients group. These clients have permissions to:

- > enroll with their respective CipherTrust Manager services, namely CTE, ProtectFile, and ProtectV.

### 'Audit Admins' group

Users who belong to "Audit Admins" group are audit records administrators. These users have permissions to:

- > View audit records

### 'Backup Admins' group

Backup Administrators have permissions to:

- > create backups
- > create backup keys

### 'CA Admins' group

CA Administrators have permissions to:

- > create Certificate Authorities on the CipherTrust Manager
- > manage Certificate Authorities on the CipherTrust Manager

**'Domain Admins' group (new in release 1.9)**

Domain Administrators have permissions to:

- > list the domains for a specific account
- > create a domain
- > access information about a domain
- > delete a domain

**'CCKM Admins' group**

There is a System Defined Group named "CCKM Admins". Users within the "CCKM Admins" group are CCKM Administrators.

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

- > AWS KMS
- > AWS Keys
- > CCKM Schedules

**'CCKM Users' group**

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 Admins" permissions to perform the AWS key operations
- > (Optional) The "Audit Admins" permissions to view the log messages

**'CTE Admins' group**

There is a System Defined Group named "CTE Admins". Users within the "CTE Admins" group are CTE Administrators.

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

- > Clients and client groups
- > Profiles, policy elements, and policies
- > GuardPoints
- > Client registration tokens

**NOTE** Only users of the "CTE Admins" group can delete CTE keys.

**'CTE Clients' group**

There is a System Defined Group named "CTE Clients". CTE clients registered with the CipherTrust Manager are part of this group.

**'DDC Admins' group**

DDC Administrators can create and manage all DDC resources. For example, they can:

- > Create and manage branch locations, classification profiles, data stores, and scans
- > Configure, run, and view reports

- > View sensitivity levels
- > Manage Hadoop configuration
- > Decrypt scan packages coming from databases

### **'DDC Full Reports Admins' group**

DDC Full Report Administrators can:

- > Create, view, and run reports
- > View available data stores
- > View available scans
- > View available classification profiles
- > View available sensitivity levels
- > View branch locations

### **'DDC Reports Admins' group**

DDC Reports Administrators can:

- > Create, view, and run reports
- > View available data stores
- > View available scans
- > View available sensitivity levels

### **'DDC L3 Support' group**

DDC L3 Support Administrators can help identify and troubleshoot issues you may encounter when using DDC. They can decrypt scan packages coming from databases.

### **'DDC Profiles Admins' group**

DDC Profile Administrators can:

- > Create and manage classification profiles
- > View available scans

### **'DDC Profiles Viewer' group**

DDC Profile Viewers can only view available classification profiles.

### **'DDC Scans Admins' group**

DDC Scan Administrators can:

- > Create and manage scans
- > View available classification profiles
- > View available data stores



**'DDC Scans Viewer' group**

DDC Scan Viewers can only view available scans.

**'DDC Stores Admins' group**

DDC Store Administrators can create and manage:

- > Data stores
- > Branch locations

**'DDC Stores Viewers' group**

DDC Store Viewers can only view available:

- > Data stores
- > Sensitivity levels
- > Branch locations

**'HSM Admins' group**

HSM Administrators have permissions to:

- > configure an HSM for the CipherTrust Manager
- > manage an HSM for the CipherTrust Manager

**'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

**'Key Users' group**

Users that is not in the "admin" group are Application Users. An Application User must also be part of the System Defined 'Key Users' group for permission to do the following:

- > Create keys
- > Perform operations with any key they own or to which they have been granted access

**'Restore Admins' group**

Restore Administrators have permissions to:

- > restore backups
- > read and restore backup keys

**'System Admins' group**

Members of the 'System Admin' group have permissions to configure the following:

- > Interfaces
- > LDAP connections
- > Logging

- > NTP
- > Instance
- > Cluster

**NOTE** Do not confuse members of this group with the 'ksadmin', the System Administrator who is responsible for deploying the CipherTrust Manager server using an SSH connection or the console port on a physical appliance. For more information on the 'ksadmin' refer to ["'ksadmin' user" on page 22](#).

### 'User Admins' group

User Administrators have permissions to create users and groups. They can:

- > Create other sub-administrator users, for example, policy administrator, key administrator, etc., and regular users
- > Assign users to the 'admin' group

### 'ProtectDB User' group

There is a System Defined Group named "ProtectDB Users". Users within this group can perform the following ProtectDB operations:

- > Configuring databases
- > Managing database connections
- > Managing database tables
- > Managing user mappings

### 'ProtectFile Administrator' group

There is a System Defined Group named "ProtectFile Admins". Users within the "ProtectFile Admins" group are ProtectFile Administrators.

**NOTE** CTE UserSpace also uses the 'ProtectFile Administrator' group.

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

- > Client profiles and clients
- > Network shares, and share-clients and share-rules associations
- > Clusters, and cluster-clients and cluster-rules associations
- > Access policies, access policy groups, and their associations
- > Rules and client-rule associations
- > Client Registration Tokens (with additional rights of System Defined Group named "CA Admins")

### 'ProtectFile User' group

There is a System Defined Group named "ProtectFile Users". CipherTrust Manager clients enrolled for ProtectFile are part of this group.

**NOTE** CTE UserSpace also uses the 'ProtectFile User' group.

### 'ProtectV Administrator' group

There is a System Defined Group named "ProtectV Admins". Users within this group are ProtectV Administrators.

A ProtectV Administrator is responsible for:

- > Managing ProtectV server settings
- > Managing ProtectV clients and their instances
- > Managing Client Registration Tokens (with additional rights of System Defined Group named "CA Admins")

### 'ProtectV Client' group

There is a System Defined Group named "ProtectV Clients". CipherTrust Manager clients enrolled for ProtectV are part of this group.

### 'ProtectV User' group

There is a System Defined Group named "ProtectV Users". Users within this group can manage ProtectV clients and their instances. This user can be common for all clients or different for each client.

## User Defined Groups

User Defined Groups are created by Application Administrators. Application Administrators can:

- > create and delete User Defined Groups
- > add users to a User Defined Group
- > remove users from a User Defined Group

Administrators may use groups solely for organizing users, or may create Policies that use group membership to assign other permissions.

Adding group permissions to keys grants users in a User Defined Group the privileges to perform operations with those keys. The semantics of the NAE-XML requests and the permissions they grant to keys are identical to SafeNet KeySecure Classic.

Groups are stored in CipherTrust Manager's internal database.

### Managing User Defined Groups

Using the GUI, REST API or the CLI, an Application Administrator can create a User Defined Group and add users/clients to this group. The following are examples using the CLI:

#### To create a new User Defined Group called "eng" :

```
$ ksctl groups create -n eng
```

The response looks like:

```
{
 "name": "eng",
 "created_at": "2018-04-27T21:15:36.644959Z",
 "updated_at": "2018-04-27T21:15:36.644959Z"
}
```

## To add a user to the new User Defined Group "eng":

You specify the group name and the ID of a user that you previously created.

```
$ ksctl groups adduser -n eng -u "<id of user>"
```

The response looks like:

```
{
 "name": "eng",
 "created_at": "2018-05-02T16:47:51.248735Z",
 "updated_at": "2018-05-02T17:24:20.015915Z"
}
```

## LDAP Group Mapping

### Overview

Groupmaps extend group-based configuration of keys to LDAP groups. A groupmap maps an LDAP group belonging to a specific LDAP connection to a CipherTrust Manager group. For example, an LDAP group can be mapped to the system defined `Key Users` group in order to allow the LDAP group's members to encrypt data. Alternatively, an LDAP group can be mapped to a newly created CipherTrust Manager group where group-based key permissions can be configured.

The groupmaps API can be used to create, delete, get, list, and modify mappings. A mapping is created by passing three parameters (LDAP group name, LDAP connection name, and CipherTrust Manager group name) to the create API. This returns an ID along with other information. The ID can be used to get, modify or delete the mapping. The list API supports searching for all mappings associated with (a) an LDAP connection, (b) a CipherTrust Manager group, and/or (c) a LDAP group. Wildcards can be used in the list API.

The API playground and the CLI tool (**ksctl**) can be used to explore the Groupmap API.

### CLI Commands

The CLI Tool (**ksctl**) can be used to create and manipulate group mappings. Help about group mappings can be obtained via the `ksctl groupmaps -h` command.

A groupmap can be created via the following command:

```
$ ksctl groupmaps create -c <ldap-connection-name> -g <ldap-group-name> -k <key-secure-group-name>
```

The LDAP connection and key secure group must exist prior to creating a connection. The mapping links the LDAP group to the key secure group. The response shows information about the mapping; here is a sample response:

```
"id": "26a980d2-c20a-4307-b12a-4898aa575122",
 "created_at": "2018-09-28T20:50:28.978112Z",
 "updated_at": "2018-09-28T20:50:28.978112Z",
 "connection_name": "ldap22",
 "ldap_group_name": "admin_staff",
 "group_name": "group1"
```

The ID that is returned by the can be used in the `ksctl groupmaps get`, `ksctl groupmaps delete` and `ksctl groupmaps modify` commands. The `ksctl groupmaps modify` command allows modification of the CipherTrust Manager group name associated with a groupmap:

```
$ ksctl groupmaps modify -i <group-map-id> -k <key-secure-group-name>
```

The `ksctl groupmaps list` command allows listing of all the available groupmaps:

```
$ ksctl groupmaps list [-s <skip>] [-l <limit>] [-c <ldap-connection-name>] [-g <ldap-group-name>]
[-k <key-secure-group-name>]
```

For example, the following command will list all groupmaps associated with an LDAP connection named "hr-ldap":

```
$ ksctl groupmaps list -l 1000 -c hr-ldap
```

The \* character can be used to perform wild card searches over connection names and group names.

## Use Cases

The utility of group mappings is illustrated by the following examples.

### Making all Users in a Specific LDAP Group members of Key Users Group

Assume that there is a LDAP connection named bababini containing a group named IT. All users in the IT group should have the ability to create keys. This can be achieved by creating a group map that maps the LDAP IT group into the built-in CipherTrust Manager 'Key Users' group:

```
$ ksctl groupmaps create -c bababini -g IT -k "Key Users"
```

Users in the LDAP group IT can create keys after issuing this command.

### Two LDAP groups share keys

Assume that there is a LDAP connection named bababini containing two groups: IT and Engineering. It is desired to share cryptographic keys between the two LDAP groups. This can be achieved by the following steps:

1. Create a key secure group called `it-engg-shared-keys`.
2. Create cryptographic keys and allow all users in the `it-engg-shared-keys` group access to those keys
3. Create a group mapping between the IT and `it-engg-shared-keys` via the command:

```
$ ksctl groupmaps create -c bababini -g IT -k it-engg-shared-keys
```

4. Create a group mapping between the Engineering and `it-engg-shared-keys` via the command:

```
$ ksctl groupmaps create -c bababini -g Engineering -k it-engg-shared-keys
```

Users in both groups can now share the keys created in step 2 above.

## Services

The Thales CipherTrust Manager contains operations for the management of CipherTrust Manager services. It supports operations such as a reset of the service (erases all data) and restart of a service (without erasing all data).

**NOTE** The following operations can have a major impact on the availability of CipherTrust Manager and the data inside of it. We recommend taking a backup of your devices and information before using these operations.

The following are typical operations initiated using the **ksctl**. If you are using the API, see the "API playground".

## Services Reset

This operation can be used to return a CipherTrust Manager to its initial state. The **Reset** operation will wipe all data from the CipherTrust Manager. This includes keys, users, and audit logs.

**\*\*WARNING\*\* This is a destructive operation and will wipe all data in the CipherTrust Manager.**

This operation requires authentication and an optional parameter called `--delay int` that defines the delay, in seconds, before the CipherTrust Manager initiates the operation.

### To initiate a services reset:

```
$ ksctl services reset
```

This returns the following response:

Please note that this operation could take some time.

```
WARNING - Reset is a destructive operation and will wipe all data in the CipherTrust Manager!
Are you really sure you want to continue? [y/N] y
Resetting CipherTrust Manager services...
```

## Services Restart

This operation restarts the CipherTrust Manager without causing a hard reboot of the appliance. It requires authentication and an optional parameter called '**delay**' that defines the delay, in seconds, before the CipherTrust Manager initiates the operation.

This operation can be used in the following scenarios:

- > configuration changes
- > general troubleshooting

### To initiate a services restart:

```
$ ksctl services restart
```

This returns the following response:

```
Please note that this operation could take some time and would temporarily disrupt the service.
Are you really sure you want to continue? [y/N] y
Restarting CipherTrust Manager services...
```

## Services Status

This operation reports the status of a CipherTrust Manager services (nae and kmip).

### To check on the status of each service:

```
$ ksctl services status
```

This returns the following response:

```
{
 "status": "started",
 "services": [
 {
 "name": "nae",
```

```

 "status": "started"
 },
 {
 "name": "kmip",
 "status": "started"
 }
]
}

```

### To check on the overall status of the services:

```
$ ksctl services status -o
```

This returns the following response:

```
"started"
```

## Hardware Security Module

A Hardware Security Module (HSM) is a physical device that provides more secure management of sensitive data, such as keys, inside Thales CipherTrust Manager.

Thales CipherTrust Manager internally uses a chain of key encryption keys (KEKs) to securely store and protect sensitive data such as user keys. When an HSM is used, the CipherTrust Manager generates and uses a set of keys on the HSM partition that protect the KEKs chain and become the root of trust. If multiple CipherTrust Manager instances are configured to use the same HSM partition they end up using shared 'root of trust' keys.

**NOTE** Configuring an HSM as the root-of-trust can be easily done using the GUI. This can also be done using the API or the CLI toolkit **ksctl**, as shown below. Using the cloud-init method is not recommended because it exposes sensitive information, such as the partition password, that can be viewed by anyone with access to the associated AWS console.

**NOTE** HSM setup might fail if time of the system is out of sync. It is recommended to set up time sync using NTP before continuing.

With the CipherTrust Manager API or CLI toolkit **ksctl**, it is possible to:

- Setup HSMs
- Get and List detailed information on existing HSMs
- Add HSMs
- Delete HSMs

Refer to the CipherTrust Manager API and CLI online documentation for more information.

## Supported HSMs

The CipherTrust Manager supports the following HSMs:

- SafeNet Luna Network HSM (version 5, 6 and 7)
- Luna T-Series Network HSM

- SafeNet Luna PCIe HSM
- AWS CloudHSM (Cavium)
- SafeNet Data Protection On Demand (DPoD) Key Vault HSM

## Hybrid HSM

Since release 1.3.0, any CipherTrust Manager can be clustered with another CipherTrust Manager, regardless of the HSM partition it is using, or even if it is not using an HSM partition at all. This is different from prior releases that required all CipherTrust Manager's in a cluster to be connected to the same HSM partition. When cluster nodes have different HSMs, it is called a "Hybrid HSM" configuration. When cluster nodes share the same HSM partition, it is called a "Shared HSM" configuration. There are several implications to a "Hybrid HSM" configuration that should be taken into consideration when deploying a cluster.

1. **System Security:** A system is only as secure as its weakest link. In a Hybrid HSM configuration, a CipherTrust Manager instance that is rooted in an HSM can be clustered with an instance that is not - this reduces the overall security of the system. The most secure configuration is to have all CipherTrust Manager instances use an HSM.
2. **Cluster Join Security:** When a system is joined to a cluster, it must receive certain keys that allow it to join the cluster. A shared HSM partition can protect these keys from ever being exposed. In this scenario, the join operation should use the **ksctl** utility's "`--shared-hsm-partition`" flag for the best security.
3. **Backup Security:** When a backup is created, there is a **ksctl** option called "`--tied-to-hsm`" that can tie the backup to the HSM of the instance. This is the most secure backup, but also limits it to being restored on a system with access to that specific HSM partition. By default, a backup is not tied to a specific HSM partition and can be restored to any compatible CipherTrust Manager system.

## Setting up an HSM

Creating the first HSM requires that you supply the hsm type and connection information (connInfo).

The general **ksctl** hsm setup command is:

```
$ ksctl hsm setup <hsm type> <config parameters>
```

The valid options for hsm type are:

- "luna" for SafeNet Luna HSM,
- "lunapci" for SafeNet Luna PCIe HSM,
- "lunatct" for Luna T-Series Network HSM
- "aws" for AWS CloudHSM,
- "dpod" for DPoD Key Vault HSM

The configuration info (config parameter) depends on the hsm type and is described below.

**NOTE** A reset is performed after adding the HSM, with a delay in seconds (default: 5) specified before the reset is performed.



**\*\*WARNING\*\***

Be aware that performing a Reset is destructive; it deletes all information in the system and so must be used with great care. You must backup all information you wish to retain.

**SafeNet Luna Network HSM and Luna T-Series HSM(hsm type: luna, lunatct)**

For the SafeNet Luna Network HSM or Luna T-Series HSM, the required parameters for initial configuration are:

- hsm-host  
IP or hostname of the HSM
- partition-name  
The name of the HSM partition to use
- partition-password  
The password of the initial partition to use. This password will also be used if more servers and high-availability (HA) mode is used, in which case all HSM must have the same password.
- serial  
Serial number of the partition to use
- server-cert-file  
File containing server certificate in PEM format
- client-cert-file  
File containing client certificate in PEM format
- client-cert-key-file  
File containing client private key in PEM format
- delay  
Delay in seconds before reset, defaults to 5 seconds
- reset  
Reset instance as part of operation

Following is a full example for a SafeNet Luna Network HSM setup command:

```
$ ksctl hsm setup luna --reset --partition-name "partition name" --partition_password
"sOmEP@ssword"
--hsm-host 192.168.0.1 --serial 1234 --server-cert-file server_cert.pem --client-cert-file client_
cert.pem
--client-cert-key-file client_cert_key.pem
```

**SafeNet Luna PCIe HSM (hsm type: lunapci)**

For the SafeNet Luna PCIe HSM, the required parameters for initial configuration are:

- partition-name  
The name of the HSM partition to use for Luna SA
- partition-password  
The password of the initial partition to use

- delay  
Delay in seconds before reset, defaults to 5 seconds
- reset  
Reset instance as part of operation

The SafeNet Luna PCIe HSM does not require any extra configuration information.

Following is a full example for a Luna PCIe HSM setup command:

```
$ ksctl hsm setup lunapci --reset --partition-name "partition name" --partition_password "sOmeP@ssword"
```

### AWS CloudHSM (Cavium) (hsm type: aws)

Prerequisite - Prepare an instance of CloudHSM in AWS:

- > Prior to configuring CipherTrust Manager instance to use AWS CloudHSM service, an instance of CloudHSM must be prepared in AWS. Refer to AWS CloudHSM documentation to create, initialize and activate a CloudHSM cluster. This step involves running a separate AWS client instance to connect and setup HSM users as well as sign the cluster CSR of your cloudhsm cluster instance. Upon completion of this step you will have a cloudhsm cluster in "Active" state. See:  
<https://docs.aws.amazon.com/cloudhsm/latest/userguide/getting-started.html>
- > Create a new cryptouser on your new cloudhsm instance. Similar to the cluster activation step when you reset the CO password, use the cloudhsm\_mgmt\_util utility on the AWS client instance to create a new user with cryptouser role (CU). CipherTrust Manager will communicate to the HSM instance on behalf of this CU user you setup.

For AWS CloudHSM (Cavium), the required parameters for initial configuration are:

- partition-name  
(Optional) The name of the HSM partition to use for CloudHSM. Defaults to "cavium"
- cryptouser-password  
Specify the credentials of a cryptouser in the form of "username:password"
- hsm-host  
IP Address or host name of cloudhsm cluster
- server-cert-file  
This is the CA public certificate that was used to sign the CloudHSM Cluster CSR during the cluster initialization. Restated, it is the certificate that signed the Cluster certificate. Refer to:  
[<https://docs.aws.amazon.com/cloudhsm/latest/userguide/initialize-cluster.html>]
- delay  
Delay in seconds before reset, defaults to 5 seconds
- reset  
Reset instance as part of operation

Following is a full example of an AWS CloudHSM setup command:

```
$ ksctl hsm setup aws-cloud-hsm --reset --cryptouser_password "sOmeP@ssword" --hsm-host 192.168.0.1 --server-cert-file server_cert.pem
```

## SafeNet Data Protection On Demand (DPoD) Key Vault HSM (hsm type: dpod)

Prerequisites:

- > The connection information (partition\_name and partition\_password) is obtained while Provisioning your HSM on Demand service.
- > The configuration information (certificates, cv\_app\_specific\_data and cv\_partition\_data) is obtained from the HSM client package archive file.

For DPoD Key Vault, the required parameters for initial configuration are:

- partition-name  
The name of the partition configured during partition initialization
- co-password  
The password of the Crypto Officer
- server-cert-file  
Server certificate in PEM format from server-certificate.pem file
- partition-cert-file  
Partition certificate in PEM format from partition-certificate.pem file
- partition-ca-cert-file  
Partition CA certificate in PEM format from partition-ca-certificate.pem file
- cv-app-specific-data  
CVAppSpecificData value from Chrystoki.conf file
- cv-partition-data  
PartitionData00 value from Chrystoki.conf file
- delay  
Delay in seconds before reset, defaults to 5 seconds
- reset  
Reset instance as part of operation

The following is a full example for a DPoD Key Vault HSM setup command:

```
$ ksctl hsm setup dpod --reset --partition-name "partition name" --co-password "sOmeP@ssword"
--server-cert-file server_cert.pem --partition-cert-file partition_cert.pem
--partition-ca-cert-file partition_ca_cert.pem --cv-app-specific-data "Chrystoki-conf-
CVAppSpecificData" --cv-partition-data "Chrystoki-conf-PartitionData00"
```

## Getting and Listing HSMs

`Get` and `List` return detail information about the instance(s) added previously, `Get` returns a single instance and `List` returns an array of HSM data.

Following is a full example of a HSM Servers `List` command:

```
$ ksctl hsm servers list
{
 "total": 2,
 "resources": [
 {
```

```

 "id": "2f81163b-f53d-45c1-b151-9bc333ce3a04",
 "type": "luna",
 "config": {
 "host": "192.168.0.1",
 "serial": "1234",
 "server-cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----"
 }
 },
 {
 "id": "be942b82-b00e-4f73-9081-28186e1580e6",
 "type": "luna",
 "config": {
 "host": "192.168.0.2",
 "serial": "4321",
 "server-cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----"
 }
 }
]
}

```

## Adding an HSM

**NOTE** This operation is only supported by the SafeNet Luna Network HSM and Luna T-Series HSM.

Adding subsequent HSMs to a CipherTrust Manager inserts them into the HA group. The first `hsm servers add` command that you do automatically creates an HA group. Adding an HSM only requires that you supply the configuration information.

**NOTE** To add extra AWS CloudHSM instances to the existing HSM cluster, no change is required on CipherTrust Manager; consult AWS CloudHSM documentation.

The configuration information (config) is a JSON blob; it has the following keys:

- host
- serial
- server-cert
- forceClear (optional)

When set to "true", forcefully clears all existing data on the partition of the joining node before adding it to the HA group. Make sure the existing objects are not used elsewhere. They cannot be recovered after the deletion. Use with caution.

- forceCopy (optional)

When set to "true", all existing objects on the joining partition are retained and propagated within the HA group. This option can be useful if the joining partition is shared among CipherTrust Manager instances that use the partition only as a member of a current HA group. This option can also be useful if the joining partition is shared among different applications other than CipherTrust Manager, and the goal is to retain the existing security objects.

Do not use the `forceCopy` option if the joining partition was previously used as root of trust for other CipherTrust Manager instance(s) and that it was not part of the current HSM HA group. Using this option in such cases can break the existing CipherTrust Managers that use the HA group servers. Use with caution.

Following is a full command example for adding a SafeNet Luna Network HSM:

```
$ ksctl hsm servers add --config '{ "host": "192.168.0.2", "serial": "4321",
"server-cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----" }'
```

If the joining HSM partition contains data, the operation will fail unless `forceClear` or `forceCopy` flags are specified. The following example command shows how to forcefully add a partition by first wiping all its data:

```
$ ksctl hsm servers add --config '{ "host": "192.168.0.2", "serial": "4321", "forceClear": "true",
"server-cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----" }'
```

**\*\*WARNING\*\* DO NOT use `forceClear` and `forceCopy` flags by default when adding a new HSM server. If the command fails indicating the joining node contains objects:**

1. Manually log on to the partition.
2. Carefully investigate the use of existing objects.
3. Repeat the command with `forceClear` or `forceCopy` flag.

## Deleting an HSM

`Delete` will remove the record for the `--id` provided, with optional parameters for `reset` and `delay`. Just as in the `Setup` function, deleting the last HSM requires a reset of the CipherTrust Manager. After the `reset` operation, newly encrypted data is longer rooted to an HSM.

**NOTE** Just as in the `Setup` function (see ["Setting up an HSM" on page 33](#), after deleting the last HSM, a reset of the CipherTrust Manager must be performed before the HSM is reused.

**\*\*WARNING\*\* Performing a Reset is destructive; it deletes all information in the system and so must be used with great care. You must backup all information you wish to retain.**

Following is a full example for the `delete` command:

```
$ ksctl hsm servers delete --id ce6da730-c266-44f8-a0f6-97e49c7596eb --reset
```

## Clusters and Nodes

Clusters and Nodes are the resources used to create and manage Thales CipherTrust Manager clustering. A cluster is a group of connected Thales CipherTrust Manager appliances that share data.

The main purpose of clustering is 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 any operation. A cluster supports any combination of appliance models; both virtual and physical appliances can join the same cluster.

**NOTE** Cluster Node Limitation - The maximum number of cluster nodes is 10. Joining an 11th node will fail.

If a member of the cluster gets temporarily disconnected, an alarm is set and syslog messages are sent. When it comes back it synchronizes with other members of the cluster. If a member of the cluster gets permanently disconnected, other members should be notified by sending a `ksctl cluster nodes delete` command to one of them - the others will tell each other. This prevents the remaining nodes from continually storing catch-up changes for the missing node, which will eventually fill up their local volumes.

Nodes in a cluster communicate over port 5432. Members of the cluster must have bi-directional access to each other on port 5432.

Use the **ksctl** command line interface for managing clusters and nodes. The relevant commands are:

```
ksctl cluster csr
ksctl cluster delete
ksctl cluster fulljoin
ksctl cluster info
ksctl cluster join
ksctl cluster new
ksctl cluster nodes list
ksctl cluster nodes get
ksctl cluster nodes create
ksctl cluster nodes delete
```

**NOTE** All global flags can be configured using a configuration file. Otherwise, you will need to specify the URL, username and password for each call as demonstrated in a few examples below.

**NOTE** The cluster commands require an IP address (or Hostname) of both the cluster member and of the joining node. It is important to note that these will be used by the nodes to talk to each other, and not by the CLI to talk to the nodes. For example, in AWS if the nodes are on the same VPC, the internal IP address of the member and joining node should be used, not a public IP.

## What is not clustered?

Most things are clustered, however the following node specific items are not clustered, that is, are not replicated across the cluster.

- > Backup Files
- > Backup Keys
- > NTP Configuration
- > HSM Configuration
- > Server Name
- > Server License

**NOTE** The 'Connector Lock Code', unlike the 'Key Manager Lock Code', is cluster-wide. This means that a license applied using the Connector Lock Code is replicated across the cluster.

## Managing Clusters and Nodes

### To check the status of a cluster

To check the status of a cluster enter the command:

```
$ ksctl cluster info
```

This returns the following response:

```
{
 "nodeID": "",
 "status": {
 "code": "none",
 "description": "not clustered"
 },
}
```

### To create a new cluster

Make the following call on a running appliance (you will need to insert the hostname/IP of the appliance). When running cluster new, --host and --url are most likely going to be identical except that --host will not have the protocol. To create a new cluster enter the command:

```
$ ksctl cluster new --host=localhostName --url=urlOfCurrentNode --user=username --
password=Password_1
```

This returns the following response:

```
{
 "nodeID": "ab40e178-5f1d-4f03-8b26-7ca378f74988",
 "status": {
 "code": "r",
 "description": "ready"
 },
 "nodeCount": 1
}
```

### To show the status of all nodes in the cluster

Enter the following command on any node:

```
$ ksctl cluster nodes list
```

This returns the following response:

```
{
 "skip": 0,
 "limit": 256,
 "total": 1,
 "resources": [
 {
 "nodeID": "c05ead39-94ac-4459-a371-9915e3e16ebf",
 "status": {
 "code": "r",
 "description": "ready"
 },
 },
],
}
```

```

 "host": "kylo_pg_1",
 "isThisNode": true
 }
]
}

```

### To show the status of a single node in the cluster

To show the status of a single node in a cluster enter the command:

```
$ ksctl cluster nodes get --id=c05ead39-94ac-4459-a371-9915e3e16ebf
```

This returns the following response:

```

{
 "nodeID": "c05ead39-94ac-4459-a371-9915e3e16ebf",
 "status": {
 "code": "r",
 "description": "ready"
 },
 "host": "kylo_pg_1",
 "isThisNode": true
}

```

### To join a node to a cluster

When joining a node to a cluster, three steps need to happen:

1. Get the csr from the joining node by entering the command **ksctl cluster csr**
2. Get the certificate and CA chain from a member node by entering the command **ksctl cluster nodes create**
3. Run the join command from the joining node by entering the command **ksctl cluster join**

While all of these commands can be run individually, it is simpler to use the **fulljoin** command, which does all of these steps. For the cluster **fulljoin** command, you are required to input a member's IP or DNS, the joining node's IP or DNS, and either the joining node's configuration file, or its username, password, and URL. To complete the process using the **fulljoin** command enter the command:

```
$ ksctl cluster fulljoin --member=<member_IP_or_DNS> --newnodehost=<joining_IP_or_DNS> --newnodeconfig=<config_File_of_Joining_Node>.
```

CipherTrust Manager internally uses a flexible security architecture to allow clustering instances with both shared and distinct 'root of trust' configurations.

If the member node and the new node share the same HSM partition, it is recommended to pass the flag **--shared-hsm-partition** during nodes **create** or **fulljoin**. The flag further increases the security of cluster join procedure by ensuring that the new node indeed has access to the same HSM partition root keys before it can join the cluster.

When the new node uses a distinct HSM partition or does not use any HSM, it is still possible to join it to an existing cluster by not specifying the **--shared-hsm-partition** flag.



**NOTE** Each CipherTrust Manager node in the cluster uses its own root of trust configuration to protect the KEKs chain that secure the sensitive data of the cluster. When a new node, not connected to an HSM, joins a CipherTrust Manager cluster with all HSM connected nodes, it becomes the weakest point in the key hierarchy and potentially weakens the overall security of the cluster. Use with caution.

### An example of the cluster fulljoin command using a configuration file:

```
$ ksctl cluster fulljoin --member=memberIPorDNS --newnodehost=joiningIPorDNS --
newnodeconfig=configFileOfJoiningNode
```

Response:

When you add a node to a cluster, the existing data of the node is deleted. Are you sure want to join? [y/N]

y

Response:

```
Attempting to get the CSR from the joining node...
Finished getting the CSR from the joining node...
Attempting to get the Certificate and CA Chain from the member node...
Finished getting the Certificate and CA Chain from the member node...
Attempting to join the new node to the member's cluster...
```

```
{
 "nodeID": "",
 "status": {
 "code": "creating",
 "description": "joining: creating (1/5)"
 }
}
```

When a node joins a cluster, the node adopts the credentials of the cluster. Would you like to write the new cluster credentials to the provided configuration file? [y/N]

y

### An example of the cluster fulljoin command without using a configuration file:

```
$ ksctl cluster fulljoin --member=memberIPorDNS --newnodehost=joiningIPorDNS --
newnodepass=joiningNodePass
--newnodeuser=joiningNodeUser --newnodeurl=joiningNodeURL
```

Response:

When you add a node to a cluster, the existing data of the node is deleted. Are you sure want to join? [y/N]

y

```
Attempting to get the CSR from the joining node...
Finished getting the CSR from the joining node...
Attempting to get the Certificate and CA Chain from the member node...
Finished getting the Certificate and CA Chain from the member node...
Attempting to join the new node to the member's cluster...
```

```
{
 "nodeID": "",
 "status": {
 "code": "creating",
 "description": "joining: creating (1/5)"
 }
}
```

## To remove a node from a cluster

To remove a node from a cluster, first remove that node from the cluster then delete the cluster configuration on the removed node. Deleting the cluster configuration allows a node to rejoin the original cluster, join another cluster, or create a new cluster.

1. Remove the node from the cluster. A node cannot remove itself, so you must call this on some other node in the cluster:

```
$ ksctl cluster nodes delete --id=ebab0738-6e09-4b0d-8c99-850d7f24dfac
```

2. Delete the cluster configuration on the **removed node**:

```
$ ksctl cluster delete
```

**NOTE** If `ksctl cluster delete` doesn't work for any reason it is always possible to perform a full system reset to ensure any left over data is removed from the node. A node can be reset using the `ksctl services reset` command.

**\*\*WARNING\*\*** Refer to section ["Services" on page 30](#) for important information on using this command.

## To rejoin a node to a cluster

Before rejoining it is recommended to reset the node.

To reset the node enter the command:

```
$ ksctl services reset
```

**\*\*WARNING\*\*** Refer to section ["Services" on page 30](#) for important information on using this command.

Join the node as described in ["To join a node to a cluster" on page 41](#).

## To create a new cluster from a removed node

Ensure the node have been successfully removed from the cluster and that `ksctl cluster delete` has been performed, as described in ["To remove a node from a cluster" above](#).

Create a new cluster as described in ["To create a new cluster" on page 40](#).

## Cluster Upgrade

A cluster can be upgraded in two ways:

- > ["Using the Cluster Remove/Rebuild Method" on the next page](#)
- > ["In-place Cluster Upgrade" on the next page](#)

## Using the Cluster Remove/Rebuild Method

### To upgrade a cluster using the cluster remove/rebuild method:

1. On one of the cluster nodes, create and download a backup with corresponding backup key, in case there are any problems.
2. Remove all nodes from the cluster except one.
3. Perform the upgrade on that remaining node.
  - a. Ensure there is at least 12GB of space available (not including the upgrade file) before proceeding.
  - b. scp the archive file to the CipherTrust Manager:
 

```
$ scp -i <identity_file> <update file name> ksadmin@<ip>:.
```
  - c. SSH into the CipherTrust Manager as ksadmin and run the following command:
 

```
$ sudo /opt/keysecure/ks_upgrade.sh -f <~/filename>
```

The signature of the archive file is verified and the upgrade is applied.
4. Re-build the cluster by creating a new cluster on this node.
5. Perform the upgrade on all other removed nodes.

**NOTE** If a previously used node is to be re-used, the cluster must first be deleted from that system.

6. Join new instances to the cluster.

### In-place Cluster Upgrade

A cluster can be upgraded in-place since version 1.8.0. The upgrade is limited to one minor version at a time, for example, from 1.8.0 to 1.9.0; from 1.9.0 to 1.10.0; or from 1.10.0 to 2.0.0. Be aware of the following considerations when performing an in-place cluster upgrade.

- > The node being upgraded will be inaccessible during the upgrade. This may be as long as 10 or more minutes. Clients must be able to handle this outage.
- > There will be a brief period of time (under 30 seconds) where the database will be locked while upgrading the first node. This affects all nodes at the same time, and some nodes may give error responses during this time.
- > All nodes in the cluster should be upgraded as soon as possible - nodes running different version of the software will behave differently, potentially causing problems with applications.

### To perform an in-place cluster upgrade, do the following:

1. Before doing any upgrade operation, ensure that you have a backup, and that you have downloaded the backup and associated backup key.
2. Ensure all nodes in the cluster are up and operating normally. Resolve any issues (like removing any obsolete nodes) before performing the upgrade.
3. Perform a system upgrade on each node, one at a time. Ensure the upgrade of each node is complete and that the node is operating normally, before proceeding to the next node.

**NOTE** When updating the first node in a cluster, the cluster nodes may briefly experience slower than usual response times. This occurs because the shared database schema for the cluster is updated with the first node.

# Tokens

The Thales CipherTrust Manager supports two types of tokens:

- > API Tokens (JWT)
- > Refresh Tokens

Both tokens can be issued (created) using username and password but the API Token (JWT) can alternatively be issued using a refresh token. API Tokens (JWT) are short lived tokens and are used for accessing the REST API as a "Bearer" token. API Tokens (JWT) are valid for 5 minutes.

Refresh tokens are long lived tokens and can be used as an alternative way to issue API Tokens (JWT). One typical use case for refresh tokens is for a long lived browser session where a user enters credentials to get a refresh token. This allows the browser to automatically issue new API Tokens (JWT) using the refresh token during the session.

Another typical use case for refresh tokens is to give an application access to the CipherTrust Manager without the need to share the username and password to the application. Instead the application can be given a refresh token to allow it to issue short lived API Tokens (JWT). The same CipherTrust Manager account (username/password) can issue and revoke several independent refresh tokens, providing access control per application.

## API Tokens (JWT)

Tokens are string values used to authenticate calls to the REST API. You use an endpoint to trade a user credential for an API token:

```
GET /api/v1/auth/tokens/
```

### Example: Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJkMzEzYzcwZS01NmYyLTQ5MTU0OGJkNy01ZmMyYmUyNzk2YzEiLCJzdWIiOiJsbnBnbHximi2ZjOGZlNylnN2ZlLTQ1YzEtOWU1OS0zYmUxNTRkMTZjYmQiLCJpc3MiOiJreWxvIiwiaWF0Ijoi2017-08-03T19:00:00ZCjE6MTQ4MTEyNDU0MH0.tWlkVnYTicAzJL746F2HB5HAqxOdnHJ-QyjK84

## Refresh Tokens

API Tokens expire, but the same endpoint can be used to "refresh" a token, that is, exchange a token that is close to expiration for a fresh token. This is done using a "refresh token".

Refresh tokens are longer lived tokens that can be used instead of user credentials for requesting a new API access token when the current API access token becomes invalid or expires. The refresh token might expire as well and should be renewed periodically depending on the nature of the client application.

Refresh tokens can be created by passing an additional flag "issueRefreshToken" when requesting an API token.

### Example: Refresh token

ujeViKSNKXUbiL8fEjliukE8AgSzH8DxojwnYuDL4hoX4OqpZX1OEPPDgRPtkdQY

### Example use cases for refresh tokens:

- > Provide an application with a refresh token to allow access to the service.
- > Provide a browser with a refresh token to allow a more long lived session.

## To issue an API token (JWT) using ksctl:

You can issue an API token (JWT) in one of three ways.

- > Issue using username and password:

```
$./ksctl tokens create --user admin --password admin --issue-rt | jq -r .refresh_token
```

Refresh token reply (example):

e7UySu7fj8ZAwPKVIjEGwOTmt8AidxFHK8Wc4tL0RiaHVf7sd3oBQ74y2M8pPhNI

- > Issue using a refresh token:

```
$./ksctl tokens create --refresh-token
e7UySu7fj8ZAwpKVIjEGwOTmt8AidxFHK8Wc4tL0RiaHVf7sd3oBQ74y2M8pPhN
```

Response:

```
{
 "jwt":
 "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJkOWNhZTVlOC0xNDJmLTQ2NmQtOTlhZSliODI5YmQ4ZTc0YWQiaLCJzdBWLiOiJsb2NhbmhwMTdiNTNmYyIjA5LTA0TUtYmFiZS03NmM3MTF1NDI5ZGMiLCJpc3MiOiJreWxvIiwiaWF0IjoiYWNja3lsbyIsInByZWZlcnlZLF91c2VybmfTZSI6ImFkbWluIiwiaXYzVzdCI6eyJncm91cHMlOlsiYWRTaW4iXX0sImp3dGlkiJoINTgOMDI0N2YtZTQ5ZS00YmNkLTk0NjctNzE5YTU0QGQ0NjViIiwiaWF0IjoxNTMwODk1Nzg4LCJleHAiOiElMzA4OTYwODh9.JMRtLevUUv5YOsxg4BJhHMT5pyzlvwXOTWLD49EzTU4",
 "duration": 300,
 "token_type": "Bearer"
}
```

- > Assign the API Token (JWT) to the environment variable using a refresh token:

```
export KSCTL_JWT=$(ksctl tokens create --refresh-token
e7UySu7fj8ZAwPKVIjEGWoTmt8AidxFHK8Wc4tL0RiaHvF7sd3oBQ74y2M8pPhNI | jq -r ".jwt")
```

### To refresh the refresh token using ksctl:

Use this command refresh the refresh token without the username and password.

```
$./ksctl tokens create --issue-rt --refresh-token
cIb6wlPzazBnJM35v9QyP6LWN70lGnGfNVUMGQfzjrcY8Hfv9cJjuMIXFHqNiO9g
```

Response:

```
{
 "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiI2NzI3ZWQ1Ni0yNjkwLmZctOGNhMC1iYmJhMzNhMjBkNmZlLCJzdWIiOiJsbnhbmHxmdDBiNTcwYi04NDhlLTRlNDAtOGEeYS0xYTEyZTJjZDA4MjQiLCJpc3MiOiJreWxvIiwiaWF0IjoiOTAxODg4ZjYtODIzNy00ZWQwLTlkNWItOWJmNjBmNGQzMDE3IiwiaWF0IjoxOTMTMxMTQ3OTMxLCJleHAiOiE1MzExNDgyMzF9.5IKirMRFYfLeDXnN1LOqvxfnjZLL4lwCmbhZazHOhiU",
 "duration": 300,
```

```

 "token_type": "Bearer",
 "refresh_token": "1xy4tVzBzjrzsVho4xWQAtN5Whil3a71eH6nYMTi4VNSIEWXHv7NpKGAItdQRUip"
 }

```

## Revoke Refresh Token

A refresh token can be revoked by passing the refresh token and client id or user credentials to the following endpoints:

POST /api/v1/auth/revoke

**NOTE** A revoked refresh token can no longer be used to renew API tokens.

## Login Command

Use the login command to create and save a long lived refresh token that makes it easier to access your CipherTrust Manager server. This method is safer than storing a password in the `config.yaml` file because a refresh token can easily be revoked and renewed if ever exposed. For more information on refresh tokens, refer to ["Refresh Tokens" on page 45](#).

### Configuration file with a saved refresh token

The `login` command can be used as an alternative to creating a `config.yaml` file as defined in the Admin Guide > CLI toolkit > Installation . When using the `login` command, if no `config.yaml` file exists, one is created in the `.ksctl` subdirectory of the user's home directory, or the configuration file that is specified in the `login` command is used. See examples below.

The `login` command requires:

- > User name
- > URL of the CipherTrust Manager instance
- > Flag indicating that SSL verification should not take place

### Example 1: When no config.yaml file exists

```
$ ksctl login --user <username> --url https://<kylo-system> --nossllverify
```

**NOTE** Replace `<kylo-system>` with the name or the IP address of the host with the CipherTrust Manager instance.

You are then prompted for the password. If the password is successful, you will see the following message:

```
this login will expire in 30 days
```

This command also creates a `config.yaml` file similar to the following:

```

KSCTL_VERBOSITY: false
KSCTL_RESP: json
KSCTL_URL: https://kylo-system
KSCTL_USERNAME: admin
KSCTL_REFRESH_TOKEN: icmtiodYvNfCAr5XXXODlgJjATkYPpYs3oHRXXxCdGWhGpSXXX70x3AQF1tkcpGO
KSCTL_CLIENTID: ffbdd794-68a3-4da9-83bc-9b0dd7a3ee48
KSCTL_NOSSLVERIFY: true

```

### Example 2: When specifying the config.yaml file to use

The following `login` command can be run to create a custom configuration file by adding the `configfile` variable:

```
$ ksctl login --user <username> --url https://<kylo-system> --nossllverify --configfile localconfig.yaml
```

As in Example 1, you are prompted for the password. On successful creation of the refresh token, a custom configuration file is created with the name `localconfig.yaml`.

### Example 3: When `config.yaml` file already exists

When the URL and the username or refresh token are already stored in a `config.yaml` file, the following command can be run:

```
$ ksctl login
```

### Example 4: Initial `login` command using password

```
$ ksctl login --user <username>
```

You are prompted for a password. If successful, a refresh token with a default lifetime of 30 days is created.

You will see the following message:

```
$ this login will expire in 30 days
```

The refresh token is written to the `config.yaml` file to be used by subsequent invocations of `ksctl`.

**NOTE** The `config.yaml` file stores an additional information, `KSCTL_CLIENTID`. This is used by the `login` command and should not be deleted or modified.

## Refresh token lifetime

If you do not specify the lifetime for a refresh token, a default of 30 days is used.

### Example 1: Specifying a refresh token lifetime in days

Use the `--days` option to specify the number of days the token should be valid.

```
$ ksctl login --user admin --days 10
```

Returned information:

```
this login will expire in 10 days
```

### Example 2: Specifying a refresh token lifetime in hours

Use the `--hours` option to specify the number of hours the token should be valid.

```
$ ksctl login --user admin --hours 10
```

Returned information:

```
this login will expire in 10 hours
```

### Example 3: Specifying a refresh token lifetime in days and hours

Use both the `--days` and `--hours` options to specify the number of days and hours the token should be valid.

```
$ ksctl login --user admin --days 10 --hours 10
```

Returned information:

```
this login will expire in 10 days, 10 hours
```

## Summary of `login` command options

The following options can be provided in the `login` command. Each option used overrides the corresponding value in the configuration file.

- > `--user`: required only when there is no configuration file. If provided, the value is saved to the configuration file.
- > `--password`: never required, never saved to the configuration file.
- > `--url`: required only when there is no configuration file. If provided, the value is saved to the configuration file.
- > `--noSSLverify`: required when there is no configuration file. If provided, the value is saved to the configuration file.
- > `--verbose`: never required, used during invocation, and never saved to the configuration file.

If you get the following error, it indicates that your ssl verification is being done and that you did not specify `--noSSLverify` in the `login` command.

```
Post https://localhost/api/v1/auth/tokens: x509: certificate is not valid for any names, but
wanted to match localhost
```

### Revoke/refresh an existing token

If a token already exists in the configuration file when a `login` command is invoked, the existing token will be revoked/refreshed automatically.

## Logout Command

The `logout` command reads the configuration file and revokes the login token found in it. This command does not require any options.

```
$ ksctl logout
```

If the login token is successfully revoked, the following message is returned:

```
login revoked
```

If there is no login refresh token or client id present in the file, the following message is returned:

```
login token and client id must exist in ksctl configuration file
```

## Policies

Thales CipherTrust Manager uses an [Attribute-Based Access Control \(ABAC\)](#) model to authorize operations. ABAC essentially means that access-control policies can be expressed in terms of the attributes of the things involved in the request. The "things" involved are usually a user, a target (like a "key"), and the "environment" of the request, which includes things like the time, source IP, source geo-region, incoming port, etc.

**\*\*WARNING\*\*** Modifying the policy is an advanced operation that can result in lockouts and render a system unusable. It is strongly recommended that policy only be modified in consultation with Technical Support.

CipherTrust Manager ABAC system is inspired by [XACML](#) and [Amazon AWS policies](#).

An ABAC policy can use the attributes of any of these things to decide if the policy applies to the request. For example, an ABAC policy can say, "Allow reading Key XYZ, but only at night, if the Key is 'enabled', and if the user is the 'hr' department, and the user is an admin, and the request comes from the internal port".



ABAC is nice because it is a super-set of other commonly used, access control systems, such as Role-Based Access Control (RBAC). RBAC is essentially just ABAC policies defined in terms of the user's "role" attribute. CipherTrust Manager ABAC policies are expressed in terms of two resources: Policies and Policy Attachments.

## Policy Composition

A Policy is composed of:

- > a set of actions (i.e. operations)
- > a set of targets
- > a rule, either "allow" or "deny"
- > a set of conditions (optional)

The conditions are matching rules, which are tested against the request, to further refine the set of requests a Policy applies to. For example, a simple policy might be:

- allow read and write to key ABC and key XYZ

Conditions can further refine this policy:

- allow read and write to key ABC and key XYZ, under the following conditions:
  - the key is for document signing
  - the user is in the "Signers" group
- the request is from an internal IP address

## Policy Attachments

Policies on their own are not enforced. Policies have to be attached to a set of users first. This is done by creating Policy Attachments.

A Policy Attachment refers to a single policy, and set of users. The set of users is expressed as matching rule (sort of like policy conditions) which is tested against the user information contained the API token. For example, policy attachments can match a single user, or all users in a group, or users contain some flag in their metadata, or can match all users.

For example, say I named the Policy above "Internal HR Key Usage". I could enforce that policy by attaching it to users in the HR department. The Policy Attachment would look like:

- > Policy: Internal HR Key Usage
- > Attached To: Users in the HR department

This is a bit confusing, since the policy's conditions can refer to user attributes as well. For example, I could have moved the "Signers" group condition from the policy to the policy attachment, and have the same effect. This redundancy is intentional: it allows some added flexibility with the expression and application of policies.

## Default Policies

On launch, Thales CipherTrust Manager is pre-installed with a set of Policies and Policy Attachments, which define a default set of group based access controls. However, admins can add to, revise, or replace these Policies with their own, to implement almost any kind of access control regime.

**\*\*WARNING\*\*** Modifying the policy is an advanced operation that can result in lockouts and render a system unusable. It is strongly recommended that policy only be modified in consultation with Technical Support.

The default policies enforce the following access control rules:

- > The "admin" user can do anything
- > Members of the "admin" group can do anything
- > All users can create keys
- > Users may perform any operation on keys they created
- > Users can grant group permissions to the keys they created
- > Users in the "global" group can perform most operations to keys with the "global" flag set

Taken together, these policies implement an authorization model that mimics SafeNet KeySecure Classic:

- > All Users can create keys, which they "own"
- > Users have admin privileges to the keys they own
- > Key owners can grant user groups limited usage privileges to their keys

## Usage

Policies are currently somewhat dangerous primitives; it is possible to create policies that will lock out all access to Thales CipherTrust Manager. Plans are to provide some canned policy sets for common access control regimes (such as RBAC), and some safety measures to prevent users from applying policies that would lock out admins irreversibly.

## Records

Audit records are logs of auditable events in Thales CipherTrust Manager. To view these records in the GUI, go to **Key & Access Management > Records > Server Records**.

You can also view the records that are uploaded from the CipherTrust Manager Clients (for example, CTE). To view these records in the GUI, go to **Key & Access Management > Records > Client Records**.

Audit records can be fetched and searched using the CipherTrust Manager REST API, CLI, or the GUI.

**NOTE** To reduce cluster traffic and disk usage for deployments that support a large number of transactions, it is recommended to configure CipherTrust Manager to disable local database audit logging and enable logging using remote Syslog servers.

## Enabling NAE Crypto Audit Records

CipherTrust Manager can be configured to keep a record of various NAE cryptographic operations that are performed by CipherTrust Manager appliance.

To record NAE cryptographic operations and activities:

1. Go to **Admin Settings > System > Properties**.

- Under the Audit Record Settings, enable **NAE Crypto Audit Records** option.

CipherTrust Manager will now log all NAE related operations and activities. To view these logs, go to **Keys & Access Management > Records > Server Records**.

#### NOTE

- > Logging NAE Crypto activities is a memory-intensive task, as thousands of such cryptographic operations are performed every second. Thus, enabling this setting causes a significant drop in NAE performance and Crypto Operations.
- > NAE Crypto Audit Records are logged within 30 seconds after the occurrence of an operation activity.

## Disabling local database audit 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 CipherTrust Manager to disable logging to a local database and enable logging using a remote Syslog server. This significantly reduces cluster traffic and disk usage.

#### Example:

```
$ ksctl properties modify -n ENABLE_RECORDS_DB_STORE -p false
```

**NOTE** This is a cluster wide setting and only needs to be run against one node in the cluster.

**NOTE** To prevent the Audit Records database table from growing too large, it is automatically trimmed when it reaches 10gb in size. If a permanent record is required, syslog should be used and the audit records stored elsewhere.

#### To list all records:

```
$ ksctl records list
```

The response looks like:

```
{
 "skip": 0,
 "limit": 10,
 "total": 40,
 "Resources": [
 {
 "id": "18ef98f1-0704-4dec-a6f0-e87a581ce52f",
 "uri": "kylo:kylo:audit:records:18ef98f1-0704-4dec-a6f0-e87a581ce52f",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "system:system:admin:applications:system",
 "devAccount": "kylo:kylo:admin:accounts:kylo",
 "createdAt": "2018-05-22T18:34:05.239823Z",
 "message": "Create Token",
 "service": "kylo",
 "requestId": "45a8dcef-0b9e-44e2-b3e1-51b3d2ef99ca",
 "instanceId": "",
 "instanceTime": "0001-01-01T00:00:00Z",
 "instanceTz": "",
 "success": true,
```

```

 "username": ""
 },
 ...
}

```

There will be a number of records returned (10 shown in the example above). Most have been removed in the above example for space reasons. Note that the response indicates that there are 40 records available ("total": 40,). These records can be returned with options in the **ksctl records** commands. Type the following command to get more information:

```
$ ksctl records list --help
```

## Configuring remote Syslog server

Audit records can optionally be sent to one or more external Syslog server(s). By default, audit records are stored in the local database and continue to do so even if Syslog connections are configured. Each audit record will be sent to each configured Syslog connection.

A Syslog connection can either use UDP or TCP + TLS as the transport protocol. When TCP + TLS is used a trusted CA certificate in PEM format must also be provided.

All Syslog messages are generated with severity notice and facility local0.

An example entry looks as follows:

```

2019-08-12 06:22:16 keysecure1 KeySecure: <134>1 2019-08-12 06:22:16 keysecure KeySecure -
b7999852-dd64-46e9-b924-c3999eca9fad [msg="Update Syslog Connection" sev="6" details="'Update
Syslog Connection' succeeded"
({"createdAt":"2019-08-12T06:22:16.113081Z", "details":{"id":"e847e529-d331-45f9-a494-
83ee7ce6ab69"},"message":"Update Syslog
Connection", "service":"kylo", "success":true, "username":"admin", "severity":"info", "lineage":"","so
urce":""})]

```

The first time stamp is generated and added by Syslog and the second time stamp is the time of the actual audit record.

In a multi-node clustered environment, the Syslog connection configurations are automatically synchronized and each node is aware of all Syslog servers. The Syslog message is sent from the currently active node. This means that if an event that results in an audit record is performed on node 1, the Syslog message originates from node 1. Similarly, if an audit event is performed on node 2, the Syslog message originates from node 2.

**NOTE** The Syslog connection configuration can take up to five minutes to apply to all nodes in the cluster.

## Syslog Connections API

The Syslog connections API supports listing, adding, deleting, getting, and updating Syslog connections. Refer to the API guide, CLI Guide, or the GUI for more details.

Refer to ["Syslogs" on page 55](#) for instructions on configuring connection to a Syslog server on the CipherTrust Manager console.

## Configuring alarm triggers based on an audit record

The audit system can be configured to trigger one or more alarms based on the content of a record. To create an alarm configuration:

1. Log on to the CipherTrust Manager console as administrator.

2. Click **Keys & Access Management** to open the application.
3. In the left pane, click **Records**. The Records page is displayed on the right.
4. Click the **Alarm Configurations** tab.
5. Click **Configure Alarm Configuration**. The **Alarm Config - Create** dialog box is displayed. Specify the following details:
  - a. **Severity**: Defines the nature of the alarm. It can be Info, Error, Critical, or Warning.
  - b. **Name**: Name of the alarm.
  - c. **Error Threshold**: The maximum number of times the trigger condition is met in the specified time **Interval** before the alarm is raised.
  - d. **Interval**: The time duration in seconds, in which if the trigger condition is met for a defined number of times (**Error Threshold** value), the alarm is raised.
  - e. **Description**: A description for the alarm.
  - f. **Condition**: Conditional properties/events that will triggers an alarm.

**NOTE** Multiple alarms may be triggered using a single record.

#### Example:

```
$ ksctl records alarm-configs create --name "Weak RSA Key" --description "RSA key should be 2048 bits or higher" --severity critical --condition 'input.success; input.message == "Create Key"; input.details.algorithm == "RSA"; input.details.size <= 1024'
```

Response:

```
{
 "id": "4bf23269-3985-42f3-995c-8f3c8f8b6543",
 "uri": "kylo:kylo:audit:alarm-configs:4bf23269-3985-42f3-995c-8f3c8f8b6543",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2019-08-27T14:37:19.145946719Z",
 "name": "Weak RSA Key",
 "condition": "input.success; input.message == \"Create Key\"; input.details.algorithm == \"RSA\"; input.details.size <= 1024",
 "description": "RSA key should be 2048 bits or higher",
 "severity": "critical"
}
```

To trigger this alarm, create a weak RSA key:

```
$ ksctl keys create --name mysmallkey --alg RSA --size 1024
```

The alarm should be triggered and this can be confirmed by listing the alarms:

```
$ ksctl alarms list --state on
```

Here is an abbreviated response for brevity:

```
{
 ...
 "resources": [
 {
 ...
 "name": "Weak RSA Key",
 "state": "on",
 "description": "RSA key should be 2048 bits or higher",
 }
]
}
```

```

 "severity": "critical",
 "details": {
 ...
 "details": {
 "id":
"f2ec44f6977edf2d6ff7fa327b5499301732dc6b395284a71e47ac0346cf72d6",
 "uri": "kylo:kylo:vault:keys:mysmallkey-v0",
 "name": "mysmallkey",
 "size": 1024,
 "ownerId": "local|fa0e4dfa-e6fd-413b-a4a0-48c356cf0557",
 "algorithm": "RSA",
 "usageMask": 3,
 "objectType": "Private Key"
 },
 "message": "Create Key",
 "success": true,
 ...
 }
 }
}

```

### Creating Alarm Trigger for Certificate Expiration

If it is desired to create an Alarm Trigger (and receive Email Notifications) for these events, follow the steps mentioned in [Configuring alarm triggers based on an audit record](#) section, and specify the **Condition** action as `input.message == "Certificate Expiry Check"`.

**NOTE** The action **Certificate Expiry Check** can be created in root domain only.

After saving this Alarm Configuration, the [recepients in the email address](#) list will receive email notifications every time a certificate expiration related record is logged. The emails will contain the certificate IDs, expiration dates, and the domain name.

## Syslogs

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 to disable logging to a local database and enable logging using remote Syslog server(s). This significantly reduces the cluster traffic and disk usage. This section describes how to configure the connection to a remote Syslog server on the CipherTrust Manager.

The following table lists the parameters that are required when configuring the connection to a Syslog server on the CipherTrust Manager:

| Parameter              | Description                                         |
|------------------------|-----------------------------------------------------|
| Hostname or IP Address | Hostname or IP address of the Syslog server.        |
| Port                   | Port of the Syslog server. The default port is 514. |

| Parameter   | Description                                                                                                                                                                                                                                                                                                                               |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Log Format  | <p>Format in which the audit records are transferred to the Syslog server. The options are:</p> <ul style="list-style-type: none"> <li>&gt; CEF</li> <li>&gt; LEEF</li> <li>&gt; RFC5424</li> <li>&gt; Plain Message</li> </ul> <p>The default log format is RFC5424. This format adheres to the Syslog Protocol RFC 5424 guidelines.</p> |
| Transport   | <p>Transport protocol for the Syslog connection. The options are UDP and TLS. The default protocol is UDP.</p>                                                                                                                                                                                                                            |
| Certificate | <p>Trusted CA certificate in the PEM format. This field is available when the transport protocol is TLS.</p>                                                                                                                                                                                                                              |

## Configuring Connection to a Syslog Server

To configure the connection to a Syslog server:

1. Log on to the CipherTrust Manager console as administrator.
2. Click **Admin Settings** to open the application.
3. Click **Notifications > Syslog**. The **Syslog Settings** section is displayed on the right. This section displays the configured connections to Syslog servers.
4. Click **New Syslog Server**. The **Configure a connection to a Syslog Server** dialog box is displayed.
5. Specify the following details:
  - **Hostname or IP Address:** Specify the hostname or IP address of the Syslog server.
  - **Port:** Specify the port of the Syslog server.
  - **Log Format:** Select a log format from the drop-down list. The default log format is RFC5424.
  - **Transport:** Select a transport protocol from the drop-down list. The default protocol is UDP.
  - **Certificate:** Paste the content of the trusted CA certificate. This field appears when TLS is selected as the transport protocol.
6. Click **Save**. The connection to the Syslog server is configured. The connection appears under the Syslog Settings section.

## Modifying Connection to a Syslog Server

To modify the connection to a Syslog server:

1. Log on to the CipherTrust Manager console as administrator.
2. Click **Admin Settings** to open the application.
3. Click **Notifications > Syslog**. The **Syslog Settings** section is displayed on the right. This section displays the configured connections to Syslog servers.
4. Click the ellipsis icon corresponding to the desired connection and click **Edit**.

**NOTE** To delete a connection, click **Delete**.

5. Modify the fields as required.
6. Save the changes.

## Banners

There are two kind of banners, the pre-authentication banner is shown before the user logs in to the CipherTrust Manager server, and the post-authentication banner is shown after login. Both are optional and not set by default

The pre-authentication banner is useful for displaying disclaimer or legal security policy. You must accept the terms identified in this banner in order to login. Not accepting the terms will prevent you from logging in to the CipherTrust Manager server.

The post-authentication banner is useful for displaying about the system. This banner message can be used to provide general information about the server to the user such as upcoming maintenance. The user is not asked to accept this banner message.

Viewing and updating the banner pages can be done using the CLI, GUI and API.

Banners support the unicode standard and the Markdown markup language.

### Viewing the Banners using the CLI

Using the CLI, you can view the pre- and post-authentication banner by using the `banners get` command:

#### Example get commands:

```
$ ksctl banners get --name pre-auth
or
```

```
$ ksctl banners get --name post-auth
```

Default command response before updating the banner message, for each command is:

```
{
 "value": ""
}
```

Example response after updating the pre- or post authentication banner:

```
{
 "value": "Example text of the banner."
}
```

### Updating the Banners using the CLI

You can update the pre- and post-authentication banners using the `banners update` command:

There are two ways to provide a new banner message:

- > using the argument `--message <banner-message>` where `<banner-message>` is new banner message



- > using the argument `--file <banner-message-file>`, where `<banner-message-file>` is a path and filename in Markdown notation containing the new banner message

Example: `/path/banner-message.md`

#### Pre-auth Example update `--message` command:

```
$ ksctl banners update --name pre-auth --message "Before logging in to KeySecure, you must accept the following 'Terms of Use': ..."
```

Response:

```
{
 "value": "Before logging in to CipherTrust Manager, you must accept the following 'Terms of Use': ..."
}
```

#### Post-auth Example update `--message` command:

```
$ ksctl banners update --name post-auth --message "Welcome to **CipherTrust k170v Server**."
```

Response:

```
{
 "value": "Welcome to **CipherTrust k170v Server**."
}
```

#### Pre-auth Example update `--file` command:

```
$ ksctl banners update --name pre-auth --file </path/banner-message.md>
```

If the markdown file contains: "Before logging in to KeySecure, you must accept the following 'Terms of Use': ...", the response is:

```
{
 "value": "Before logging in to CipherTrust Manager, you must accept the following 'Terms of Use': ..."
}
```

#### Post-auth Example update `--file` command:

```
$ ksctl banners update --name post-auth --file </path/banner-message.md>
```

If the markdown file contains: "Welcome to \*\*CipherTrust k170v Server\*\*.", the response is:

```
{
 "value": "Welcome to **CipherTrust k170v Server**."
}
```

## Interfaces

Interfaces are services the Thales CipherTrust Manager hosts. Most interfaces are listening on a particular port, but may also represent other input channels, like local shell access or serial port access. To view the Interfaces page in the UI, go to **Admin Settings > System > Interfaces**.

The CipherTrust Manager currently supports five interfaces:

- > web: The HTTP server on port 80 and 443. This interface serves both the GUI and the REST API
- > ssh: The SSH server on port 22.
- > nae: The NAE-XML server on port 9000.
- > kmip: The KMIP server on port 5696.
- > SNMP: The SNMP Agent on port 161. For details on the SNMP interface, go to: ["SNMP" on page 69](#)

## To view the initial set of interfaces using ksctl:

```
$./ksctl interfaces list
```

### Example response:

```
{
 "skip": 0,
 "limit": 10,
 "total": 4,
 "resources": [
 {
 "id": "ee242373-2555-48c7-923b-86ed3e785504",
 "name": "kmip",
 "mode": "tls-cert-pw-opt",
 "cert_user_field": "CN",
 "auto_gen_ca_id": "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acfff84a",
 "trusted_cas": {
 "local": [
 "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acfff84a"
]
 }
 },
 {
 "createdAt": "2020-07-20T12:59:14.776939Z",
 "updatedAt": "2020-07-20T12:59:24.703806Z",
 "default_connection": "local_account",
 "port": 5696,
 "network_interface": "all",
 "interface_type": "kmip",
 "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
 },
 {
 "id": "a4db15b4-64fc-40d3-8465-9935866bbe09",
 "name": "nae",
 "mode": "no-tls-pw-req",
 "cert_user_field": "CN",
 "auto_gen_ca_id": "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acfff84a",
 "trusted_cas": {
 "local": [
 "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acfff84a"
]
 },
 "createdAt": "2020-07-20T12:59:14.770938Z",
 "updatedAt": "2020-07-20T16:21:43.427252Z",
 "default_connection": "local_account",
 "port": 9000,
 }
]
}
```

```

 "network_interface": "all",
 "interface_type": "nae",
 "local_auto_gen_attributes": {
 "cn": "nae.keysecure.local",
 "email_addresses": [
 "support@gemalto.com"
]
 },
 "names": [
 {
 "C": "US",
 "ST": "MD",
 "L": "Belcamp",
 "O": "Gemalto",
 "OU": ""
 }
],
 "generated": false
 },
 "enabled": true
},
{
 "id": "c8c6e4c1-30d7-4317-9d37-a9ec217ffb17",
 "name": "ssh",
 "trusted_cas": {},
 "createdAt": "2020-07-20T12:59:14.7787Z",
 "updatedAt": "2020-07-20T12:59:14.7787Z",
 "port": 22,
 "network_interface": "all",
 "interface_type": "ssh",
 "local_auto_gen_attributes": {
 "cn": "ssh.keysecure.local",
 "email_addresses": [
 "support@gemalto.com"
]
 },
 "names": [
 {
 "C": "US",
 "ST": "MD",
 "L": "Belcamp",
 "O": "Gemalto",
 "OU": ""
 }
],
 "generated": false
},
 "enabled": true
},
{
 "id": "e64d84b0-e952-4fa3-83c9-e1d1bc52a996",
 "name": "web",
 "mode": "unauth-tls-pw-req",
 "cert_user_field": "CN",
 "auto_gen_ca_id": "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acfff84a",
 "trusted_cas": {
 "local": [
 "kylo:kylo:naboo:localca:9d13d43a-381e-481a-80a1-9463acfff84a"
]
 },
 "createdAt": "2020-07-20T12:59:14.774618Z",
 "updatedAt": "2020-07-20T12:59:23.37516Z",
 "port": 443,
 "network_interface": "all",

```

```

 "interface_type": "web",
 "local_auto_gen_attributes": {
 "cn": "web.keysecure.local",
 "email_addresses": [
 "support@gemalto.com"
],
 "names": [
 {
 "C": "US",
 "ST": "MD",
 "L": "Belcamp",
 "O": "Gemalto",
 "OU": ""
 }
],
 "generated": false
 },
 "enabled": true
 }
]
}

```

**NOTE** Before interface changes can take effect, you must issue a Services Restart; see ["Services Restart" on page 31](#).

## Enabling/Disabling Interfaces

You can enable or disable the following interfaces:

- > **SSH**
- > **NAE**
- > **KMIP**

To enable/disable an interface using the GUI, click **Action Button > Enable/Disable**.

### NOTE

- > After the user has enabled/disabled an interface:
  - It remains in same state even after restarting the device.
  - It will get replicated if the device is part of a cluster.
- > After an interface has been disabled, the CipherTrust Manager drops all incoming and existing connections on that interface.

## Enable or Disable an interface using ksctl:

When using ksctl:

- > To enable an interface: `./ksctl interfaces enable --name <interface-name>`
- > To disable an interface: `./ksctl interfaces disable --name <interface-name>`

**NOTE** Only the SSH, KMIP, and NAE interfaces can be enabled or disabled. Replace `<interface-name>` in the above commands with `ssh`, `kmip`, or `nae` for these interfaces.

## Example: Disabling the SSH interface

```
$./ksctl interfaces disable --name ssh
{
 "id": "2e8d2344-c40b-466c-8202-d05d2cb6738a",
 "name": "ssh",
 "trusted_cas": {},
 "createdAt": "2020-08-13T10:22:32.792266Z",
 "updatedAt": "2020-08-14T11:11:28.564276Z",
 "port": 22,
 "network_interface": "all",
 "interface_type": "ssh",
 "local_auto_gen_attributes": {
 "cn": "ssh.keysecure.local",
 "email_addresses": [
 "support@gemalto.com"
]
 },
 "names": [
 {
 "C": "US",
 "ST": "MD",
 "L": "Belcamp",
 "O": "Gemalto",
 "OU": ""
 }
],
 "generated": false
},
{
 "enabled": false
}
```

## Interface Default Connection (NAE and KMIP)

NAE and KMIP interfaces may be configured to use a default connection other than `local\_account` for username/password authentication and client certificate authentication. Typically this setting is modified to simplify authentication because the username does not need to provide the connection name. For example, when a user presents their username for authentication without a connection, e.g. `jdoe`, then the CipherTrust Manager uses the default connection of the interface to authenticate the user. If the default connection is configured as `local\_account` then the user is authenticated as if `local\_account|jdoe` was presented. However, if configured as `myldapdomain` then the user is authenticated as if `myldapdomain|jdoe` was presented.

### Example - Interface configuration update for KMIP to use a new default connection:

```
$ ksctl interfaces modify -i kmip -d myldapdomain
```

Response:

```
{
 "id": "476618b4-7310-4783-819f-517c387d2281",
 "name": "kmip",
 "mode": "tls-cert-pw-opt",
 "cert_user_field": "CN",
 "auto_gen_ca_id": "kylo:kylo:naboo:localca:0d29d15d-0ddd-444d-8cc5-ffcc85d0928a",
 "trusted_cas": {
 "local": [
 "kylo:kylo:naboo:localca:0d29d15d-0ddd-444d-8cc5-ffcc85d0928a"
],
 "external": []
 }
}
```

```

 "createdAt": "2018-07-09T14:49:43.316319Z",
 "updatedAt": "2018-07-09T14:49:50.136975Z",
 "default_connection": "myldapdomain"
 }

```

## SSL/TLS Certificate (NAE, KMIP, and WEB)

During initial bootstrapping of a new server, a new local `KeySecure Root CA` is automatically generated. This CA is later used to issue a server certificate for the interfaces available in the system. All interfaces and services trust this CA by default, meaning that a client certificate issued from this initial `KeySecure Root CA` is automatically trusted by the system.

The CLI tool (`ksctl`) can be used to modify certificate configuration for a given interface. It can upload a certificate and private key or regenerate a self-signed certificate. Certificates (and the private key) can be uploaded in PEM or PKCS12 format. PKCS12 files are encrypted and require the use of `--private-key-password`.

**NOTE** We do not currently support encrypted PKCS8 keys embedded within a PEM file.

The common name (CN) in the certificate must have a valid value.

Here are CLI commands that generate a certificate and private key in PEM format, and then upload the combined certificate and key PEM file to the CipherTrust Manager.

**NOTE** The CipherTrust Manager must be restarted after modifying the interface's certificate.

### To create the certificate and key using openssl utility (version 0.9.8zh):

1. Generate a private key and certificate:

```
$ openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365
```

2. Combine the resulting `key.pem` with the `cert.pem` file and upload to the CipherTrust Manager using **ksctl**, (nae interface example).

```

$ cp key.pem certandkey.pem
$ cat cert.pem >> certandkey.pem
$ ksctl interfaces certificate modify --id=nae --format=pem --file=certandkey.pem --private-key-password password

```

### To create the certificate and key using openssl utility (version 1.0.2g):

Newer openssl versions (1.0.2g, for example) behave differently. They generate encrypted private keys.

1. With the newer versions, generate the encrypted key and certificate in PEM format:

```
$ openssl req -x509 -newkey rsa:4096 -keyout key1.pem -out cert.pem -days 365
```

2. Decrypt the encrypted key.

```
$ openssl rsa -in key1.pem -passin pass:password -out key.pem -passout pass:password
```

3. Combine the resulting `key.pem` with the `cert.pem` file and upload to the CipherTrust Manager using **ksctl**, (nae interface example).

```

$ cp key1.pem certandkey.pem
$ cat cert.pem >> certandkey.pem

```

```
$ ksctl interfaces certificate modify --id=nae --format=pem --file=certandkey.pem --private-key-password password
```

## Interface Mode (NAE, KMIP, and WEB)

The "mode interface" configuration parameter specifies the Interface mode, and must be one of the following:

- > no-tls-pw-opt: No TLS, allow anonymous logins.
- > no-tls-pw-req: No TLS, user must supply password.
- > unauth-tls-pw-opt: TLS, allow anonymous logins, ignore client cert.
- > unauth-tls-pw-req: TLS, user must supply password, ignore client cert.
- > tls-pw-opt: TLS, allow anonymous logins, verify client cert.
- > tls-pw-req: TLS, user must supply password, verify client cert.
- > tls-cert-pw-opt: Verify client cert, user name taken from client cert, auth request is optional.
- > tls-cert-and-pw: Verify client cert, password is needed, user name in cert must match user name in authentication request.

The following interface modes verify that the TLS client certificate is signed by one of the trusted CAs:

- > tls-pw-opt
- > tls-pw-req
- > tls-cert-pw-opt
- > tls-cert-and-pw-req

These interface modes extract the user name from the certificate:

- > tls-cert-pw-opt
- > tls-cert-and-pw-req

The interface mode can be changed using the API or the CLI "ksctl interfaces modify" command.

### Interface restrictions on the interface mode

Each interface has the following restrictions on the interface mode:

- > nae: All modes are allowed. Default is unauth-tls-pw-opt.
- > web: The only allowed mode is unauth-tls-pw-req.
- > kmip: Only following modes are allowed:
  - tls-cert-pw-opt
  - tls-cert-and-pw
  - tls-pw-opt

## certUserField Interface (NAE, KMIP, and WEB)

The certUserField Interface configuration parameter specifies how the user name is extracted from the client certificate. The default value is CN. This parameter must be one of the following:

- > CN: common name

- > OU: organizational unit
- > SN: surname
- > E: Email address
- > E\_ND: email without domain; the data to the left of the @ sign in the email address is taken as the user name.

**NOTE** For KMIP, only CN is supported.

## Auto-generation of Server Certificate (NAE, KMIP, and WEB)

By default, the CipherTrust Manager is configured to auto-generate a server certificate for each interface by using the initial Local CA. If the current certificate is issued by a different CA (other than the **selected** Local CA), auto-generation of the certificate is triggered every time the CipherTrust Manager service restarts. The current certificate becomes the certificate issued by the Local CA. As a result, this point onward, certificate auto-generation does not happen most of the times.

This feature is, especially, useful when a new node joins the cluster. The new node's existing Local CA is overwritten by the Local CAs from the cluster. In this case, the new node's current certificate becomes issued by a Local CA unknown to the cluster. As a result, auto-generation happens on the first restart of the new node after it joins the cluster.

## Changing the Local CA Used for Auto-generation

If needed, the Local CA used for auto-generation of the server certificate can be changed. Run the following command:

```
$./ksctl interfaces modify --id web --auto-gen-ca-id kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a
```

Response:

```
{
 "id": "3b85c5cf-6075-4555-a17c-2a12479081d4",
 "name": "web",
 "mode": "unauth-tls-pw-req",
 "cert_user_field": "CN",
 "auto_gen_ca_id": "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a",
 "trusted_cas": {
 "local": [
 "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a"
],
 "external": []
 },
 "createdAt": "2018-07-06T16:10:23.705291Z",
 "updatedAt": "2018-07-06T17:15:38.285859Z"
}
```

## Disabling Auto-generation of Server Certificate

To disable the auto-generation feature, run the following command:

```
$./ksctl interfaces modify --id web --auto-gen-ca-id ""
```

Response:



```
{
 "id": "3b85c5cf-6075-4555-a17c-2a12479081d4",
 "name": "web",
 "mode": "unauth-tls-pw-req",
 "cert_user_field": "CN",
 "auto_gen_ca_id": "",
 "trusted_cas": {
 "local": [
 "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a"
],
 "external": []
 },
 "createdAt": "2018-07-06T16:10:23.705291Z",
 "updatedAt": "2018-07-06T17:16:41.119686Z"
}
```

**NOTE** The auto-generation feature is automatically disabled if an external CA chain and server certificate are uploaded to the interface.

### Customizing Auto-generation of Interface Certificate

To supply custom fields for auto-generation of an interface certificate, specify `--local-auto-gen-attributes-file` in the `ksctl interfaces modify` command. For example, run:

```
$./ksctl interfaces modify --local-auto-gen-attributes-file <local-auto-gen-attributes-file-name>.json
```

The `--local-auto-gen-attributes-file` option specifies a file that contains attributes for auto-generation of a given interface. This provides an option to supply custom fields for automatic interface certification generation.

These parameters are for the local node only and have no effect on other nodes in a cluster. Without customization, the system default values are used, or taken from cloud-init, if applicable.

Content of a sample local auto-generation attributes file is:

```
{
 "cn": "KeySecure Server on ks.eng.thesgroup.com",
 "dns_names": [
 "ks.eng.thesgroup.com",
 "ks-ha.eng.thesgroup.com"
],
 "email_addresses": [
 "admin1@eng.thesgroup.com",
 "admin2@eng.thesgroup.com"
],
 "ip_addresses": [
 "10.3.111.11",
 "10.3.111.22"
],
 "names": [
 {
 "C": "US",
 "ST": "CA",
 "L": "San Jose",
 "O": "Thales CPL",
 "OU": "Engineering"
 }
]
}
```

```

 }
]
}

```

## Trusted CAs (NAE, KMIP, and WEB)

Each interface can have a set of either local and/or external trusted CAs for client authentication. By default each interface trusts the initial Local CA for client authentication but it can be modified by updating the interface. Please note that the concept of trusted CAs is only relevant when client authentication is used by updating the interface mode.

### Example - configuration update for web interface to trust the set of two Local CAs:

```
$./ksctl interfaces modify --id web --trusted-local-cas kylo:kylo:naboo:localca:d21584bd-7d8e-44a0-9eac-693f2fa32504,kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a
```

Response:

```

{
 "id": "3b85c5cf-6075-4555-a17c-2a12479081d4",
 "name": "web",
 "mode": "unauth-tls-pw-req",
 "cert_user_field": "CN",
 "auto_gen_ca_id": "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a",
 "trusted_cas": {
 "local": [
 "kylo:kylo:naboo:localca:d21584bd-7d8e-44a0-9eac-693f2fa32504",
 "kylo:kylo:naboo:localca:cf648c6d-2a0b-4427-a0b6-8f0e9157d15a"
],
 "external": []
 },
 "createdAt": "2018-07-06T16:10:23.705291Z",
 "updatedAt": "2018-07-06T17:35:29.053809Z"
}

```

## Adding, Removing, and Modifying Interfaces (NAE and KMIP)

You can add and remove interfaces other than default interfaces (NAE, KMIP, and WEB). Currently, the Create and Delete commands are supported only for the NAE and KMIP interfaces.

### To create a new NAE interface:

**Example:**

```
$ ksctl interfaces create -o 9009 -y nae
```

Response:

```

{
 "id": "456eb374-ec5c-40e8-bc89-4ab485c20c6c",
 "name": "nae_all_9009",
 "mode": "unauth-tls-pw-opt",
 "cert_user_field": "CN",
 "auto_gen_ca_id": "kylo:kylo:naboo:localca:61476734-4778-40ec-a3be-06654d123513",
 "trusted_cas": {
 "local": [
 "kylo:kylo:naboo:localca:61476734-4778-40ec-a3be-06654d123513"
],
 "external": []
 }
}

```

```

 },
 "createdAt": "2019-01-21T05:52:57.657447Z",
 "updatedAt": "2019-01-21T05:52:57.657447Z",
 "default_connection": "local_account",
 "custom_uid_size": 0,
 "port": 9009,
 "network_interface": "all",
 "interface_type": "nae"
 }
}

```

### 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"
}

```

### To modify TLS version of an interface:

**NOTE** The CipherTrust Manager server supports TLS version 1.0, TLS version 1.1, and TLS version 1.2. For the NAE interface, the default TLS version is 1.1, and for the KMIP interface, the default TLS version is 1.2.

```
$ ksctl interfaces modify -n <interface name> -e <minimum tls version>
```

Here:

- <interface name> can be NAE or 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"
}

```

### To delete an NAE interface:

#### Example:

```
$ ksctl interfaces delete -n nae_all_9009
```

## SNMP

### SNMP Overview

Simple Network Management Protocol (SNMP) is a request-response protocol used to communicate management information. SNMP enables network and system administrators to remotely monitor devices on the network, such as switches, routers, proxies, and hubs.

This protocol relies on three main concepts: Network Management Station (NMS), agent, and Management Information Base (MIB). The NMS is configured on a network node and runs SNMP management software; agents run on network devices that are being monitored by the NMS; and the MIB defines what kind of information can be exchanged between the agent and the NMS.

Management information is communicated between an Network Management Station (NMS) and an agent. SNMP notifications (traps and informs), sent from agents to managers, might indicate a warning or error condition or otherwise notify the manager about the agent's state.

There are three versions of SNMP: SNMPv1, SNMPv2c and SNMPv3. SNMPv1 and SNMPv2c use community-based security model and SNMPv3 uses User-based Security Model (USM).

### SNMP on CipherTrust Manager

CipherTrust Manager supports SNMP versions v1, v2c and v3.

#### SNMP v1/v2c

SNMPv1/v2c relies on the community name to provide a low level of security for communications between the NMS and agent. A pairing of an SNMP agent with some arbitrary set of SNMP application entities is called an SNMP community. Each SNMP community is named by a string of octets that is called the community name for

said community. For an agent, a subset of objects in the MIB that pertain to that element is called a SNMP MIB view. Community names are associated with a certain access level (read-only, read-write) to SNMP MIB view. When the CipherTrust Manager receives a SNMPv1/v2c request, the agent verifies that the community name in the request and can optionally validate the source IP of the request matches the community name. The request will be processed if a match is found, otherwise it is dropped. To configure the device for SNMP v1/v2c, see ["Configuring SNMPv1/v2c" on the next page](#)

## SNMP v3

SNMPv3 incorporated all the capabilities of SNMPv1/v2c, and introduced the concept of a User-based Security Model (USM). This security model supports data integrity, data origin authentication, data confidentiality and message timeliness and limited replay protection. Additionally, SNMPv3 enhanced the existing access control with View-based Access Control Model (VACM). CipherTrust Manager supports MD5, SHA, SHA-224, SHA-256, SHA-384, and SHA-512 for authentication protocols; and DES, AES, AES-192, and AES-256 for privacy protocols. To configure the device for SNMP v3, see ["Configuring SNMPv3" on page 72](#)

## MIBs

CipherTrust Manager implements the following MIBs:

- > Standard MIBs (SNMP v1, v2c, v3)
- > Internet Standard MIBs (RFC-1213)
- > Host Resources MIB (HOST-RESOURCES-MIB)
- > Distributed Management MIBs (RFC-2981, RFC-2982)

## Access Control

Structure of Management Information (SMI & SMIv2) defines ACCESS(SMIv1)/MAX-ACCESS(SMIv2) clause to allow which SNMP operations (GET, SET or Notification) are allowed for a MIB object. An ACCESS value of 'read-write' means that the object can be both read with GET or GET-NEXT request and written with a SET request, while an ACCESS value of 'not-accessible' means that the object can neither be read nor written to. In addition, access control in the CipherTrust Manager SNMP implementation makes it possible for agents to provide different levels of MIB group access to different managers. Access levels 'read-only' or 'read-write' can be provided to Standard and Enterprise MIB access groups. You can restrict access by allowing one NMS to read only standard MIBs and another NMS to read and write both standard MIBs and enterprise MIBs.

## Enterprise MIBs

The Enterprise MIB objects can be accessed using the community names/users configured with Enterprise MIB access group. The Enterprise MIB access group is able to retrieve UCD-SNMP-MIB objects. UCD-SNMP-MIB defines system monitoring objects; e.g. disk, cpu load, memory, processes.

- > Hardware Objects
  - Processes, Memory, Disk, CPU Load Average objects (UCD-SNMP-MIB)

**NOTE** Currently, there are no specific enterprise MIB objects defined for the CipherTrust Manager.

## Notifications

The CipherTrust Manager can generate and send notifications (traps, informs or both) to configured management stations when an important system events occur such as Authentication failure, Interface link up/down, and DisMan events.

- > Trap: Traps are unreliable notification message sent from a device to management station.
- > Inform: Informs are reliable notification message sent from a device to management station that also requires an acknowledgment from the management station.

| Traps/Notifications                  | Description                                                                     |
|--------------------------------------|---------------------------------------------------------------------------------|
| Authentication Failure               | SNMP requests (GET, GET-NEXT, GET-BULK, SET) authentication failed              |
| Link Up/Down Notifications           | Communication link state change                                                 |
| Temperature Notifications            | Temperature sensors exceeds 68 degrees Centigrade                               |
| Free Memory Notifications            | Total memory available is less than 1 GB                                        |
| Processor Load Average Notifications | Load average exceeds: 30.00 in 1 minute, 10.00 in 5 minutes, 5.00 in 15 minutes |
| Disk Utilization Notifications       | Disk usage exceeds 80%                                                          |

To receive SNMP notifications (traps and informs) from CipherTrust Manager, the management station needs to be configured.

## Configuring SNMPv1/v2c

To monitor and control the CipherTrust Manager using SNMPv1/v2c, community names must be configured. Both SNMPv1 and SNMPv2c are enabled.

### Configure Communities:

Configure (add, read, update and delete) community names with management station host(s) filtering for SNMP v1 and SNMPv2c.

#### To add a community configuration:

Add a SNMP community configuration with management station host(s) filtering and mib access control.

```
$ ksctl snmp communities add -community <community-name> --source <mgmt.-station-ip-or-cidr> --mib-accesses <standard,enterprise>
```

#### Example 1:

Allow any management station to access standard MIB view using the community name 'public'

```
$ ksctl snmp communities add -community public
```

#### Example 2:

Allow a management station with IP '10.10.10.10' to access standard and enterprise MIB views using the community name 'private'.

```
$ ksctl snmp communities add -community private -source 10.10.10.10 -mib-accesses
"standard,enterprise"
```

### Example 3:

Allow a group of management stations using CIDR notation to access enterprise MIB view using the community name 'secret'

```
$ ksctl snmp communities add -community secret -source 10.10.10.0/24 -mib-accesses enterprise
```

### To read a community configuration:

To read a SNMP community configuration using its ID.

```
$ ksctl snmp communities get --id <community-id>
```

### To list community configurations:

To list all the SNMP community configurations.

```
$ ksctl snmp communities list
```

### To delete community configuration:

To delete a SNMP community configuration using its ID.

```
$ ksctl snmp communities delete --id <community-id>
```

## Configuring SNMPv3

To monitor and control the CipherTrust Manager and to receive notifications using SNMPv3, users must be configured.

### Configuring SNMP USM Users:

Configure (add, read, update and delete) SNMP USM users for management and to use with SNMP management station (notifications).

### To add a SNMP USM user:

#### Example 1:

To add a SNMP USM user configuration with no authentication and no privacy, and with read-only access to standard MIB view. No authentication and no privacy configuration is not recommended.

```
$ ksctl snmp users add --user-name rob --security-level noAuthNoPriv -mib-accesses standard
```

#### Example 2:

To add a SNMP USM user configuration with authentication but no privacy. It gives read-write access to standard and enterprise MIBs views.

```
$ ksctl snmp users add --user-name bob --security-level authNoPriv --auth-protocol SHA --auth-
password --mib-accesses "standard,enterprise" -read-write
```

#### Example 3:

To add a SNMP USM user configuration for SNMPv3 notification with authentication and privacy. There is no MIB access provided for this user.

```
$ ksctl snmp users add --user-name trapuser --security-level authPriv --auth-protocol SHA --auth-password authpass --priv-protocol AES --priv-password privpass
```

#### Example 4:

To add a SNMP USM user configuration for inform notification with authentication and privacy. There is no MIB access provided for this user. EngineID of the notification receiver is required.

```
$ ksctl snmp users add --user-name informuser --engine-id 0x0102030405 --security-level authPriv --auth-protocol SHA --auth-password authpass --priv-protocol AES --priv-password privpass
```

### To add an SNMP management station:

Add an SNMP management station configuration to send SNMP notifications to it.

#### Example 1:

To send SNMPv1 trap using community name 'private'

```
$ ksctl snmp mgmt-stations add --host 1.1.1.1 --version 1 --security-name private
```

#### Example 2:

To send SNMPv2c inform using community name 'secret'

```
$ ksctl snmp mgmt-stations add --notification-type inform --host 1.1.1.2 --version 2c --security-name secret
```

#### Example 3:

To send SNMPv3 trap using USM user name 'trapuser' that is configured through 'ksctl snmp users' command.

```
$ ksctl snmp mgmt-stations add --notification-type trap --host 1.1.1.3 --port 1162 --version 3 --security-name trapuser
```

## SMTP

SMTP stands for Simple Mail Transfer Protocol. This protocol facilitates the flow of emails over the internet. You can configure CipherTrust Manager to automatically send email notifications for system alarms to a set of email addresses, by configuring a SMTP server.

### Adding a New SMTP Server

To add a new SMTP server:

1. Log on to the CipherTrust Manager console as administrator.
2. Click **Admin Settings**.
3. Go to the **Notifications > SMTP** section.
4. Click **New SMTP Server**. The **Configure a SMTP Server** dialog box is displayed.
5. Specify the following details:
  - **Hostname or IP Address:** Specify the hostname or IP address of the SMTP server.



**NOTE** If using a **Hostname**, make sure that it is added to the CipherTrust Manager. To read more, go to ["Configuring DNS Hosts" on page 107](#).

- **Port:** Specify the port of the SMTP server.
  - **Username:** Specify the Username for the SMTP server.
  - **Sender's Email:** Specify the email address from where the email is to be sent.
  - **Password:** Specify the Password for the SMTP server's Username.
6. Click **Save**. The SMTP server is configured. The server details will appear in the SMTP Server Settings section.

**NOTE** You can configure only one SMTP server at a time. If you add a new SMTP server, the new server will replace the older SMTP server configuration.

**TIP** After configuring the SMTP server, click **Test SMTP Connection** to test the connection with the SMTP server. If the server is configured correctly, the Sender's email address will receive a test email.

## Adding/Deleting Emails in the Notification Address List

To view, add, or delete the email addresses that will receive the email notifications:

1. Log on to the CipherTrust Manager console as administrator.
2. Click **Admin Settings**.
3. Go to the **Notifications > SMTP** section.  
The list of notification email address will be displayed under **Email Notification Settings** section.
  - a. To add a new email address, click **Add Email**. Specify the desired email address and click **Save**.
  - b. To delete an existing email address, click **Delete**.

## Backups

Backups are generated on the appliance and are stored there until deleted. They can be downloaded from the appliance and uploaded back to the same or to another appliance. When a backup is created, it is always encrypted with a backup key. If a backup key is not specified, the default backup key is used to encrypt the backup. This backup key is required to restore the backup file.

Backups and backup keys are not automatically copied to other nodes in a cluster. They only exist on a node where they are created or copied to. Backups are treated as resources and are controlled through the API, ksctl, and GUI.

CipherTrust Manager supports the following types of backups:

- > ["System Backups" on the next page](#)
- > ["Domain Scoped Backups" on page 76](#)

**NOTE** Backup jobs can also be scheduled to run periodically in the background. Refer to ["Scheduling Backups" on page 86](#) for details.

## System Backups

System backups are a snapshot of the database in an appliance at a point in time. They are created on the appliance, and after creation, they can be used to:

- > Restore the appliance to a previous state
- > Transfer the backup to another appliance to restore on that appliance
- > Create a new appliance with the backup

A backup can optionally be tied to an HSM for increased protection of the data. This means that the backup can only be restored to an appliance that contains the same HSM root key as was used when the backup was taken. Unless the appliances share the same HSM partition and therefore also the HSM root key, a backup tied to the HSM cannot be restored to any other appliance. Also, if the HSM is tampered and the HSM root key is deleted, a backup tied to that HSM cannot be restored. This feature improves the protection of backup files, but requires careful management of the HSM root key used for the backup, otherwise, the data loss is possible.

CipherTrust Manager users (such as Application Administrators), who are authorized to perform backup, can perform system backups and restores. Domain administrators are not authorized to perform system backups.

**NOTE** System backups cannot be restored on a cluster node. The cluster Remove/Rebuild method should be used to restore a system backup.

### Backup contents

A system backup includes the following:

- > All domain resources
- > Keys, key attributes, and key links
- > Users, groups, and their relationships
- > Local CAs
- > External CAs
- > Certificates
- > Policies
- > Cluster information
- > Audit records
- > Syslog connections
- > Resources of all supported applications: Keys & Access Management, Admin Settings, Data Discovery, ProtectDB, ProtectFile, ProtectV, ProtectApp, and KMIP

## Release compatibility

CipherTrust Manager backups are forward compatible to the next *three* versions of backups. For example, a backup of release 1.6.1 can be restored to 1.7, 1.8, and 1.9.1 instances. Backward compatibility is not supported. For example, a backup of release 1.6.1 can be restored to a 1.7 instance, however, a backup of release 1.7 cannot be restored to a 1.6.1 instance.

## Domain Scoped Backups

Domain scoped backup and restore is the ability to back up user-specified resources partially and incrementally from a domain. They are created on the appliance, and after creation, can be used to:

- > Restore the resources on the same domain to its previous state
- > Restore on a different domain in the same appliance
- > Transfer to another appliance to restore on a different domain

CipherTrust Manager users (such as Domain Administrators), who are authorized to perform backup, can perform domain scoped backup and restore. Application Administrators can perform backup and restore operations of domain scoped backup of root domain only, not other sub-domains.

**NOTE** Domain scoped backups can be restored on any clustered node.

## Backup contents

Domain scoped backups include the following:

- > Keys, key attributes, and key links

## Release compatibility

Domain scoped backups can be restored to the same or higher versions of the CipherTrust Manager.

**NOTE** Domain scoped backups cannot be tied to an HSM.

## Typical Backup and Restore Operations

The following are typical backup operations using the **ksctl**. If you are using the API, refer to the API playground for details.

1. ["Create the backup key" on the next page](#)
2. ["Create the backup" on page 78](#)
3. ["Check the backup status" on page 81](#)
4. ["Download the backup key" on page 82](#)
5. ["Download the backup" on page 82](#)
6. ["Inspect the backup key file" on page 82](#)
7. ["Upload the backup key" on page 83](#)
8. ["Inspect the backup file" on page 84](#)
9. ["Upload the backup" on page 84](#)

10. "Restore the backup" on page 85

11. "Check the restore status" on page 86

### Create the backup key

Backups are automatically encrypted by a backup key at the time of their creation. If no key is specified, the default backup key is used. If needed, a new backup key can be created and optionally set as the default backup key.

Backup keys can be of two types:

- > System scoped keys to encrypt only system backups
- > Domain scoped keys to encrypt only domain scoped backups

### To create a system backup key

To create a system backup key, run `ksctl backupkeys create`. If not specified, the default scope is `system`.

#### Example:

```
$ ksctl backupkeys create
{
 "scope": "system",
 "account": "kylo:kylo:admin:accounts:kylo",
 "version": "2",
 "id": "5dac1d33-9c2d-4fc1-9793-961052d73a29",
 "createdAt": "2020-04-07T08:17:54.600032599Z",
 "isDefault": false
}
```

To create a system backup key and set it as the *default* key, run `ksctl backupkeys create --scope system --default`.

#### Example:

```
$ ksctl backupkeys create --scope system --default
{
 "scope": "system",
 "account": "kylo:kylo:admin:accounts:kylo",
 "version": "2",
 "id": "c05b638d-2eea-4651-9937-7bee49a95f6b",
 "createdAt": "2020-04-07T08:21:59.139786694Z",
 "isDefault": true
}
```

### To create a domain scoped backup key

To create a domain scoped backup key, run `ksctl backupkeys create --scope domain`. Here, `--scope domain` specifies that the key is domain scoped. The scope flag must be used for creating domain scoped backup keys.

#### Example:

```
$ ksctl backupkeys create --scope domain
{
 "scope": "domain",
 "account": "kylo:kylo:admin:accounts:kylo",
 "domainID": "00000000-0000-0000-0000-000000000000",
}
```

```

 "version": "2",
 "id": "88f47a71-ald7-41ae-af49-736862f4f625",
 "createdAt": "2020-04-07T08:31:29.314569058Z",
 "isDefault": false
 }

```

To create a domain scoped backup key and set it as the *default* key, run `ksctl backupkeys create --scope domain --default`.

#### Example:

```

$ ksctl backupkeys create --scope domain --default
{
 "scope": "domain",
 "account": "kylo:kylo:admin:accounts:kylo",
 "domainID": "00000000-0000-0000-0000-000000000000",
 "version": "2",
 "id": "2f91020c-5fb7-4709-aca4-f4celba9a54a",
 "createdAt": "2020-04-07T08:24:13.648719582Z",
 "isDefault": true
}

```

### To view the list of available backup keys

Run the command `ksctl backupkeys list`.

#### Example:

```

$ ksctl backupkeys list
{
 "skip": 0,
 "limit": 10,
 "total": 2,
 "resources": [
 {
 "scope": "system",
 "account": "kylo:kylo:admin:accounts:kylo",
 "version": "2",
 "id": "c05b638d-2eea-4651-9937-7bee49a95f6b",
 "createdAt": "2020-04-07T08:21:59.139786694Z",
 "isDefault": true
 },
 {
 "scope": "domain",
 "account": "kylo:kylo:admin:accounts:kylo",
 "domainID": "00000000-0000-0000-0000-000000000000",
 "version": "2",
 "id": "2f91020c-5fb7-4709-aca4-f4celba9a54a",
 "createdAt": "2020-04-07T08:24:13.648719582Z",
 "isDefault": true
 }
]
}

```

### Create the backup

Create the system and domain scoped backups using the `ksctl`.

## To create a system backup

To create a system backup, run `ksctl backup create`. If an encryption key is not specified when creating a backup, the backup is encrypted by the default backup key. You can check the status of the backup by running the `ksctl backup status` command. Refer to ["Check the backup status" on page 81](#) for details.

### Example:

```
$ ksctl backup create
{
 "scope": "system",
 "account": "kylo:kylo:admin:accounts:kylo",
 "version": "v1.1.4",
 "id": "245d3336-21fa-4a34-9498-c0b64fa4d322",
 "createdAt": "2020-04-07T08:33:44.687970269Z",
 "status": "In Progress",
 "tiedToHSM": false,
 "backupKey": "c05b638d-2eea-4651-9937-7bee49a95f6b",
 "productVersion": "v1.10.0-beta4.4388"
}
```

When a backup is initiated, its status becomes "In Progress". It may take a while until the status becomes "Completed". When a backup operation is "In Progress", another backup or restore operation cannot be initiated.

To create a system backup by specifying the scope as *system*, run `ksctl backup create --scope system`.

### Example:

```
$ ksctl backup create --scope system
{
 "scope": "system",
 "account": "kylo:kylo:admin:accounts:kylo",
 "version": "v1.1.4",
 "id": "06c7d57d-2749-4285-8c60-c2778c911cd2",
 "createdAt": "2020-04-07T08:34:49.914616905Z",
 "status": "In Progress",
 "tiedToHSM": false,
 "backupKey": "c05b638d-2eea-4651-9937-7bee49a95f6b",
 "productVersion": "v1.10.0-beta4.4388"
}
```

**NOTE** A system level backup can optionally be tied to the HSM during creation by passing the flag `--tied-to-hsm`. The backup can only be restored to instances that use the same HSM partition. This option is invalid for domain scoped backups.

To initiate a system level backup tied to the HSM, run the command:

```
$ ksctl backup create --scope system --tied-to-hsm --description
<description>
```

## To create a domain scoped backup

When creating a domain scoped backup, specify the scope as `domain`. To create a domain scoped backup, run `ksctl backup create --scope domain`. You can check the status of the backup by running the `ksctl backup status` command. Refer to ["Check the backup status" on page 81](#) for details.

Here, `--scope domain` specifies that the backup is domain scoped. The scope flag must be used for creating domain scoped backups.

**NOTE** To perform the backup operation in a sub-domain, you need to log on to the domain or use the `--domain` flag.

#### Example:

```
$ ksctl backup create --scope domain
{
 "scope": "domain",
 "account": "kylo:kylo:admin:accounts:kylo",
 "domainID": "00000000-0000-0000-0000-000000000000",
 "version": "v1.1.4",
 "id": "762d15f3-4eba-4cd7-960c-c50a84a52b48",
 "createdAt": "2020-04-07T08:36:34.667280981Z",
 "status": "In Progress",
 "tiedToHSM": false,
 "backupKey": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
 "productVersion": "v1.10.0-beta4.4388"
}
```

**NOTE** When creating a **domain** scoped backup, use filters to select resources. Filters specify selection criteria to include only certain resources in the backup from the requester's domain. Filter criteria include the resource type and its matching attribute values in JSON format for querying the resources.

> Filters apply to domain scoped backups only.

To use filters, run either of the following commands:

- > `ksctl backup create --scope domain --filters <filters-json>`
- > `ksctl backup create --scope domain --filters-jsonfile <file-name-containing-filters-json>`

Refer to the API playground for details on filters.

### To view the list of available backups

To view the list of available backups, run `ksctl backup list`.

#### Example:

```
$ ksctl backup list
{
 "skip": 0,
 "limit": 10,
 "total": 2,
 "resources": [
 {
 "scope": "system",
 "account": "kylo:kylo:admin:accounts:kylo",
 "version": "v1.1.4",
 "id": "06c7d57d-2749-4285-8c60-c2778c911cd2",
 "createdAt": "2020-04-07T08:34:49.914616905Z",
 "status": "Completed",
 "tiedToHSM": false,
```

```

 "backupKey": "c05b638d-2eea-4651-9937-7bee49a95f6b",
 "productVersion": "v1.10.0-beta4.4388"
 },
 {
 "scope": "domain",
 "account": "kylo:kylo:admin:accounts:kylo",
 "domainID": "00000000-0000-0000-0000-000000000000",
 "version": "v1.1.4",
 "id": "762d15f3-4eba-4cd7-960c-c50a84a52b48",
 "createdAt": "2020-04-07T08:36:34.667280981Z",
 "status": "Completed",
 "tiedToHSM": false,
 "backupKey": "2f91020c-5fb7-4709-aca4-f4celba9a54a",
 "productVersion": "v1.10.0-beta4.4388"
 }
]
}

```

## To view the details of a backup

To view the details of a backup, run `ksctl backup get --id "backupID"`. The status of the backup is given in the response. A status of "Completed" indicates that the backup is successful.

### Example:

```

$ ksctl backup get --id "762d15f3-4eba-4cd7-960c-c50a84a52b48"
{
 "scope": "domain",
 "account": "kylo:kylo:admin:accounts:kylo",
 "domainID": "00000000-0000-0000-0000-000000000000",
 "version": "v1.1.4",
 "id": "762d15f3-4eba-4cd7-960c-c50a84a52b48",
 "createdAt": "2020-04-07T08:36:34.667280981Z",
 "status": "Completed",
 "tiedToHSM": false,
 "backupKey": "2f91020c-5fb7-4709-aca4-f4celba9a54a",
 "productVersion": "v1.10.0-beta4.4388"
}

```

## Check the backup status

Initiating a backup begins an asynchronous operation on the appliance. Only one backup or restore operation can be performed at a time. If you try to perform another operation until the previous is complete, the operation will fail.

To view the status of the last backup, run the command `ksctl backup status`. The status of the backup is given in the response.

### Example:

```

$ ksctl backup status
{
 "scope": "system",
 "account": "kylo:kylo:admin:accounts:kylo",
 "operation": "Backup create",
 "id": "f6b55640-ddec-49d0-893b-4f33f0809bf0",
 "started": "2020-04-10T00:27:01.078766633Z",
 "status": "In Progress"
}

```



In the output above, the status of the backup is `In Progress`.

### Download the backup key

The key with which the backup is encrypted must be transferred to the appliance where the backup will be restored. The backup key is required for the restore. To transfer a backup key, it must be downloaded using the command `ksctl backupkeys download`. After the key is downloaded, it can be uploaded to the appliance to restore the backup.

#### To download the backup key

Run the command `ksctl backupkeys download --id "<backupkeyID>" --file "<file-to-store-key>"`. The same command is used to the download system scoped and domain scoped backup keys.

Here:

- > "`<backupkeyID>`": ID of the backup key to download.
- > "`<file-to-store-key>`": Name of the file to store the backup key.

Specify a password for the backup key when prompted.

#### Example - download system backup key:

```
$ ksctl backupkeys download --id "c05b638d-2eea-4651-9937-7bee49a95f6b" --file "sysbkkey.txt"
Password is required for backupkey.
Please provide the backupkey password:
```

### Download the backup

The backup must be transferred to the appliance where it will be restored. To transfer a backup, it must be downloaded by using the command `ksctl backup download`. After the backup is downloaded, it can be uploaded to the appliance for restore.

#### To download the backup

To download a backup, run the command `ksctl backup download --id "<backupID>" --file "<backup-file>"`.

Here,

- > "`<backupID>`": ID of the backup to download.
- > "`<backup-file>`": Name of the file to store the backup.

#### Example - download system backup:

```
$ ksctl backup download --id "245d3336-21fa-4a34-9498-c0b64fa4d322" --file "backup-file.txt"
```

### Inspect the backup key file

Before uploading the backup key file, inspect it by reviewing its metadata. To inspect the backup key file, run the command `ksctl backupkeys inspect --file "<backup-key-file>"`.

#### Example - system backup key file:

```
$ backupkeys inspect --file "sysbkkey.txt"
{
 "version": 2,
```

```

 "keyid": "c05b638d-2eea-4651-9937-7bee49a95f6b",
 "createdAt": "2020-04-07T08:21:59.139786694Z"
 }

```

### Example - domain scoped backup key file:

```

$ backupkeys inspect --file "domainbkkey.txt"
{
 "version": 2,
 "keyid": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
 "createdAt": "2020-04-07T08:24:13.648719582Z"
}

```

## Upload the backup key

Upload the backup key to the appliance where the backup will be restored.

### To upload the system backup key

Run the command `ksctl backupkeys upload --file "<backup-key-file>". Here, <backup-key-file> represents the name of the system backup key file.`

Specify the password of the backup key when prompted.

#### Example:

```

$ ksctl backupkeys upload --file "sysbkkey.txt"
Please provide the backupkey password:
{
 "scope": "system",
 "account": "kylo:kylo:admin:accounts:kylo",
 "version": "2",
 "id": "c05b638d-2eea-4651-9937-7bee49a95f6b",
 "createdAt": "2020-04-07T08:21:59.139786694Z",
 "isDefault": false
}

```

### To upload the domain scoped backup key

Run the command `ksctl backupkeys upload --scope domain --file "<backup-key-file>".`

Here,

- > `--scope domain`: Scope of the backup key is domain. The scope flag must be used for uploading domain scoped backup keys.
- > `<backup-key-file>`: Name of the domain scoped backup key file.

Specify the password of the domain scoped backup key when prompted.

#### Example:

```

$ ksctl backupkeys upload --scope domain --file "domainbkkey.txt"
Please provide the backupkey password:
{
 "scope": "domain",
 "account": "kylo:kylo:admin:accounts:kylo",
 "domainID": "00000000-0000-0000-0000-000000000000",
 "version": "2",
 "id": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
}

```

```

 "createdAt": "2020-04-07T08:24:13.648719582Z",
 "isDefault": false
 }

```

## Inspect the backup file

Before uploading the backup file, inspect the file by reviewing its metadata. To inspect the backup file, run the command `ksctl backup inspect --file "<backup-file>"`.

### Example - system backup file:

```

$ ksctl backup inspect --file "backup-file.txt"
{
 "version": "v1.1.4",
 "createdAt": "2020-04-07T08:33:44.687970269Z",
 "scope": "system",
 "domainID": "",
 "backupKeyId": "c05b638d-2eea-4651-9937-7bee49a95f6b",
 "isEncrypted": true,
 "creatorAccount": "kylo:kylo:admin:accounts:kylo",
 "isHSMBacked": false,
 "description": "",
 "checksum": "9f19ab202f860cfa5de7fcbe70872a0aea1ee91af9092c80cb89939c35aea994",
 "mkekChecksum": "dlfe06afe54bfe5f242323b42c640b218a76fae91e7bb0baf10720b47a39be20",
 "productVersion": "v1.10.0-beta4.4388",
 "id": "245d3336-21fa-4a34-9498-c0b64fa4d322"
}

```

### Example - domain scoped backup file:

```

$ ksctl backup inspect --file "domain-backup-file.txt"
{
 "version": "v1.1.4",
 "createdAt": "2020-04-07T08:36:34.667280981Z",
 "scope": "domain",
 "domainID": "00000000-0000-0000-0000-000000000000",
 "backupKeyId": "2f91020c-5fb7-4709-aca4-f4ce1ba9a54a",
 "isEncrypted": true,
 "creatorAccount": "kylo:kylo:admin:accounts:kylo",
 "isHSMBacked": false,
 "description": "",
 "checksum": "1c1a58b2ead4f09fceb67297603dbf01e40a8bec6d812e78a64287afa9619f8f",
 "mkekChecksum": "12f700ed4f872bd93864cfd7f9d162574025d3ee30ec2c08ccf13362c899a837",
 "productVersion": "v1.10.0-beta4.4388",
 "id": "762d15f3-4eba-4cd7-960c-c50a84a52b48"
}

```

## Upload the backup

Upload the backup to the appliance where it will be restored.

### To upload the system backup

Run the command `ksctl backup upload --file "<backup-file>"`. Here, `<backup-file>` represents the name of the system backup file.

### Example:

```

$ ksctl backup upload --file "backup-file.txt"
{
 "scope": "system",

```

```

 "account": "kylo:kylo:admin:accounts:kylo",
 "version": "v1.1.4",
 "id": "245d3336-21fa-4a34-9498-c0b64fa4d322",
 "createdAt": "2020-04-07T08:33:44.687970269Z",
 "status": "Completed",
 "tiedToHSM": false,
 "backupKey": "c05b638d-2eea-4651-9937-7bee49a95f6b",
 "productVersion": "v1.10.0-beta4.4388"
 }

```

## To upload the domain scoped backup

Run the command `ksctl backup upload --scope domain --file "<domain-backup-file>"`.

Here,

- > `--scope domain`: Scope of the backup is domain. The scope flag must be used for uploading domain scoped backups.
- > `<domain-backup-file>`: Name of the domain scoped backup file.

### Example:

```

$ ksctl backup upload --scope domain --file "domain-backup-file.txt"
{
 "scope": "domain",
 "account": "kylo:kylo:admin:accounts:kylo",
 "domainID": "00000000-0000-0000-0000-000000000000",
 "version": "2",
 "id": "2f91020c-5fb7-4709-aca4-f4celba9a54a",
 "createdAt": "2020-04-07T08:24:13.648719582Z",
 "isDefault": false
}

```

## Restore the backup

After the backup and the backup key are uploaded to the appliance, you can restore the backup on the appliance. Check the restore status by running the `ksctl backup status` command. Refer to ["Check the restore status" on the next page](#).

## To restore the backup

Run the command `ksctl backup restore --id "<backupID>"`. Here, `<backupID>` specifies the ID of the backup to restore.

System services will restart after the system backup is restored. Restore of the domain scoped backup does not restart the system services.

### Example:

```

$ ksctl backup restore --id "245d3336-21fa-4a34-9498-c0b64fa4d322"
{
 "version": "v1.1.4",
 "id": "245d3336-21fa-4a34-9498-c0b64fa4d322",
 "createdAt": "2020-04-07T08:33:44.687970269Z",
 "status": "Completed",
 "tiedToHSM": false
}

```

## Check the restore status

After performing the backup restore, you can check the status of the operation. To view the status of the restore, run the command `ksctl backup status`. The status of the restore is given in the response.

### Example:

```
$ ksctl backup status
{
 "account": "kylo:kylo:admin:accounts:kylo",
 "id": "f6b55640-ddec-49d0-893b-4f33f0809bf0",
 "createdAt": "0001-01-01T00:00:00Z",
 "status": "In Progress",
 "tiedToHSM": false
}
```

In the output above, the status of the restore is `In Progress`.

**NOTE** It is normal to get an error like the following for status when the system services are restarting after the system backup is restored:

```
Error: Server error. HTTP status code: 500
```

## Scheduling Backups

The CipherTrust Manager can periodically run database backup jobs in the background. This can be achieved by creating scheduler configurations for database backups. Scheduler configurations are supported for both system and domain scoped backups. Refer to ["Backups" on page 74](#) for details on backups and their types.

This section provides examples of backup scheduler configurations using the `ksctl` utility. Refer to the `ksctl` or API documentation for details.

**NOTE** Scheduler itself is a generic service, so it provides granularity of a minute. A job configuration should take the nature of operation information into consideration. For example, backing up a database every minute is a highly unlikely use case.

### Scheduling System Scoped Backup Jobs

To schedule a system scoped backup job, run the command:

```
ksctl scheduler configs create backup --name <config-name> --run-at "<cron-expression-format>" --description "<config-description>" --backup-key-id "<backupkeyID>"
```

Here,

- > `<config-name>`: Name for the job configuration.
- > `<cron-expression-format>`: Time when the job runs. The format must have *five* fields. If the number of fields is not equal to five, the format becomes invalid. Refer to ["Time Specification" on page 89](#) for details.
- > `<config-description>`: Description of the job configuration.
- > `<backupkeyID>`: ID of the backup key. If a key ID is not specified, the default backup key will be used.

**NOTE** Refer to the `ksctl` or API documentation for complete list of parameters needed for backup operations.

**Example:**

The following command schedules a system scoped database backup job using the specified backup key everyday at 9 AM.

```
$ ksctl scheduler configs create backup --name daily_backup --run-at "0 9 * * *" --
description "Daily system backup" --backup-key-id "e817b3ba-1787-496a-b5ac-
27756e9efb1b"
```

**Scheduling Domain Scoped Backup Jobs**

To schedule a domain scoped backup job, run the command:

```
ksctl scheduler configs create backup --scope "domain" --name <config-name> --run-at
"<cron-expression-format>" --description "<config-description>"
```

Here,

- > `--scope domain`: Scope of the backup is domain. The scope flag must be used for scheduling domain scoped backup jobs.
- > `<config-name>`: Name for the job configuration.
- > `<cron-expression-format>`: Time when the job runs. The format must have *five* fields. If the number of fields is not equal to five, the format becomes invalid. Refer to ["Time Specification" on page 89](#) for details.
- > `<config-description>`: Description of the job configuration.

**Example:**

The following command schedules a domain scoped database backup job every Monday at 9 AM.

```
$ ksctl scheduler configs create backup --scope "domain" --name daily_backup --run-
at "0 9 * * 1" --description "Weekly domain backup"
```

## Disk Encryption

For added security, the disk of the Thales CipherTrust Manager appliance can be fully encrypted. Encryption can either be initiated when an instance is first launched, or on an already launched instance. Because installation specific secrets are generated the first time a CipherTrust Manager appliance is run, it is recommended that the instance be encrypted at launch time to ensure these secrets are never exposed.

**\*\*WARNING\*\*** After encrypting the disk, you will need to unlock the encrypted instance on every boot using the 'ksctl diskenc secureboot' command. See ["To unlock an encrypted instance" on page 89](#).

To configure the instance to encrypt at launch, use the cloud-init settings below.

```
#cloud-config
diskenc:
 encrypt: true
```

**NOTE** Cloud-init configuration (user-data) files uses YAML syntax; indentation is important and tabs can not be used.

This file must be passed to the CipherTrust Manager instance, which uses cloud-init to enable the encryption. How this file is passed to the CipherTrust Manager appliance depends on the environment in which it is deployed. For other examples of passing cloud-init config.dat file in supported environments, go to [Plan Configuration Settings for Cloud Init](#).

## To encrypt a disk image

When using the disk image, how the cloud-init data is passed to Thales CipherTrust Manager depends on the virtualization platform.

For example, when launching a virtual machine with the Qcow2 image using 'libvirt', the cloud-init data has to be passed in as an ISO file. The ISO can be generated as follows:

### 1. Prepare the **user-data** file as follows:

- a. Rename the file **config.data** to **user-data**.
- b. Since, user's ssh key is used for wrapping a layer of encryption keys, it must be added to the cloud-init config. So, the 'user-data' file should look like:

```
#cloud-config
diskenc:
 encrypt: true
ssh_authorized_keys:
 - <replace with user's ssh public key>
```

**NOTE** 'ssh\_authorized\_keys' can be configured with multiple ssh public keys.

### 2. Create a **meta-data** file and provide instance parameters, for example:

```
$ instance-id: <some instance id>
$ local-hostname: <host name for the instance>
```

### 3. Create an ISO image file:

- a. Make sure **genisoimage** utility is installed.
- b. Create the ISO file.

```
$ genisoimage -o config.iso -volid cidata -joliet -rock user-data meta-data
```

### 4. Launch instance using **virt-install**, for example:

```
$ virt-install --virt-type kvm --name <virtual image name> --ram 2048
--disk path=<path to ciphertrust qcow2 image>,size=16,format=qcow2
--disk path=<path to config.iso> --network network=default --graphics vnc,listen=0.0.0.0 --
noautoconsole --os-type=linux
--os-variant=ubuntu16.04 --import
```

## To check encryption status

To check the encryption progress, you can run following CLI command:

```
$ ksctl diskenc status -p
```

### Example:

```
$ ksctl diskenc status -p
```

This returns the following response:

```
Encrypting...
```

14.81 GiB / 15.52 GiB [=====>-----] 95.44% 11s

## To unlock an encrypted instance

The instance will start up after the encryption has finished. To unlock the instance, you must run following CLI command:

```
$ ksctl diskenc secureboot -i <ssh key used to launch the instance> -u https://<instance dns name>
```

Every time an encrypted instance boots, the above CLI command must be executed to unlock the instance.

## Scheduler

The CipherTrust Manager can periodically run key rotation and database backup jobs in the background. These jobs can be:

- > Configured using the scheduler
- > Scheduled by creating a job configuration

A job configuration defines what job is run and when. Job configurations can be read, listed, updated, and deleted. Jobs are run after job configurations are created. The jobs that were run can also be listed.

This section provides examples of scheduler configurations using the ksctl utility. Refer to the ksctl or API documentation for details.

## Time Specification

The time that a job runs is described in the `run-at` parameter using the cron expression format:

```
" * * * * * "
```

These five values indicate when the job should be executed. They are in order of minute, hour, day of month, month, and day of week.

The following table lists the accepted values:

| Field Name   | Mandatory? | Allowed Values  | Allowed Special Characters |
|--------------|------------|-----------------|----------------------------|
| Minute       | Yes        | 0-59            | * / , -                    |
| Hour         | Yes        | 0-23            | * / , -                    |
| Day of month | Yes        | 1-31            | * / , - ?                  |
| Month        | Yes        | 1-12 or JAN-DEC | * / , -                    |
| Day of week  | Yes        | 0-6 or SUN-SAT  | * / , - ?                  |

**NOTE** Month and Day of week field values are case *insensitive*. For example, "SUN", "Sun", and "sun" are equally accepted.



**NOTE** Scheduler itself is a generic service, so it provides granularity of a minute. A job configuration should take the nature of operation information into consideration. For example, rotating the same set of multiple keys and backing up a database every minute are highly unlikely use cases.

## Allowed Special Characters

The following table describes the use of special characters in the cron expression format:

| Special Character   | Description                                                                                                                                                                                                                             |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Asterisk ( * )      | Indicate that the cron expression will match for all values of the field. For example, using an asterisk in the 5th field (month) would indicate every month.                                                                           |
| Slash ( / )         | Describe increments of ranges. For example, 3-59/15 in the 1st field (minutes) would indicate the 3rd minute of the hour and every 15 minutes thereafter.                                                                               |
| Comma ( , )         | Separate items of a list. For example, using "MON,WED,FRI" in the 5th field (day of week) would mean Mondays, Wednesdays, and Fridays.                                                                                                  |
| Hyphen ( - )        | Define ranges. For example, 9-17 would indicate every hour between 9am and 5pm inclusive, MON-WED would indicate Monday through Wednesday, and JAN-JUN would indicate every month from January through June.                            |
| Question mark ( ? ) | Use instead of (*) for leaving either day-of-month or day-of-week blank. ( ? ) means no specific value. For example, to run the job every 10th day of the month and not limit the job to a particular day of the week: " * * 10 * ? *". |

## Creating Job Configurations

To create a job configuration, run the command `ksctl scheduler configs create <operation> --name <config-name> --run-at "<cron-expression-format>"`.

Here,

- > `<operation>`: Operation to schedule. The options are:
  - `backup`: Create a job configuration for backup operation. Refer for ["Scheduling Backups" on page 86](#) more details.
  - `key-rotation`: Create a job configuration for key rotation. Refer for ["Scheduling Key Rotation" on page 126](#) more details.
- > `<config-name>`: Name for the job configuration.
- > `<cron-expression-format>`: Time when the job runs. The format must have *five* fields. If the number of fields is not equal to five, the format becomes invalid. Refer to ["Time Specification" on the previous page](#) for details.

### Example - backup scheduler:

```
$ ksctl scheduler configs create backup --name ab-backup-config --run-at "0 12 * * 0"
{
 "id": "fd0c27b8-a502-47cc-b6bc-4c1c6fbbbe89a",
```

```

 "uri": "kylo:kylo:scheduler:job_configs:fd0c27b8-a502-47cc-b6bc-4c1c6fbbe89a",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2020-04-12T06:06:15.110530931Z",
 "name": "ab-backup-config",
 "updatedAt": "2020-04-12T06:06:15.110530931Z",
 "description": "",
 "operation": "database_backup",
 "run_at": "0 12 * * 0",
 "run_on": "any",
 "disabled": false
 }
}

```

### Example - key rotation scheduler:

```

$ ksctl scheduler configs create key-rotation --name ab-key-rot-config --run-at "45 11 * * 0"
{
 "id": "20c5263d-e296-4994-9129-b71aff49f4b3",
 "uri": "kylo:kylo:scheduler:job_configs:20c5263d-e296-4994-9129-b71aff49f4b3",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2020-04-12T06:07:40.05471195Z",
 "name": "ab-key-rot-config",
 "updatedAt": "2020-04-12T06:07:40.05471195Z",
 "description": "",
 "operation": "key_rotation",
 "run_at": "45 11 * * 0",
 "run_on": "any",
 "disabled": false
}

```

## Viewing Existing Job Configurations

To view the list of job configurations, run the command `ksctl scheduler configs list`.

### Example:

```

$ ksctl scheduler configs list
{
 "skip": 0,
 "limit": 10,
 "total": 2,
 "resources": [
 {
 "id": "20c5263d-e296-4994-9129-b71aff49f4b3",
 "uri": "kylo:kylo:scheduler:job_configs:20c5263d-e296-4994-9129-b71aff49f4b3",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2020-04-12T06:07:40.054712Z",
 "name": "ab-key-rot-config",
 "updatedAt": "2020-04-12T06:07:40.054712Z",
 "description": "",
 "operation": "key_rotation",
 "run_at": "45 11 * * 0",
 "run_on": "any",
 "disabled": false
 }
]
}

```

```

 },
 {
 "id": "fd0c27b8-a502-47cc-b6bc-4c1c6fbb89a",
 "uri": "kylo:kylo:scheduler:job_configs:fd0c27b8-a502-47cc-b6bc-4c1c6fbb89a",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2020-04-12T06:06:15.110531Z",
 "name": "ab-backup-config",
 "updatedAt": "2020-04-12T06:06:15.110531Z",
 "description": "",
 "operation": "database_backup",
 "run_at": "0 12 * * 0",
 "run_on": "any",
 "disabled": false
 }
]
}

```

The output shows existing job configurations.

## Viewing Details of Job Configurations

To view the details of a job configuration, run the command `ksctl scheduler configs get --id "configID"`. Here, `<configID>` represents the ID of the configuration.

### Example:

```

$ ksctl scheduler configs get --id "fd0c27b8-a502-47cc-b6bc-4c1c6fbb89a"
{
 "id": "fd0c27b8-a502-47cc-b6bc-4c1c6fbb89a",
 "uri": "kylo:kylo:scheduler:job_configs:fd0c27b8-a502-47cc-b6bc-4c1c6fbb89a",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2020-04-12T06:06:15.110531Z",
 "name": "ab-backup-config",
 "updatedAt": "2020-04-12T06:06:15.110531Z",
 "description": "",
 "operation": "database_backup",
 "run_at": "0 12 * * 0",
 "run_on": "any",
 "disabled": false
}

```

## Modifying Job Configurations

Existing job configurations can be modified when needed. To modify a job configuration, run the command:

```
ksctl scheduler configs modify <operation> --id "<configID>" <args>
```

Here,

- > `<operation>`: Operation can be backup or key-rotation.
- > `<configID>`: ID of the job configuration to be modified.
- > `<args>`: Parameters to modify.

### Example:

```
$ ksctl scheduler configs modify key-rotation --id "20c5263d-e296-4994-9129-b71aff49f4b3" --
run-at "50 11 * * 0"
{
 "id": "20c5263d-e296-4994-9129-b71aff49f4b3",
 "uri": "kylo:kylo:scheduler:job_configs:20c5263d-e296-4994-9129-b71aff49f4b3",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2020-04-12T06:07:40.054712Z",
 "name": "ab-key-rot-config",
 "updatedAt": "2020-04-12T06:14:57.23658004Z",
 "description": "",
 "operation": "key_rotation",
 "run_at": "50 11 * * 0",
 "run_on": "any",
 "disabled": false
}
```

## Manually Running Job Configurations

A scheduled configuration can be run manually when needed. To manually run a job configuration, run the command:

```
ksctl scheduler configs run-now --id "<configID>"
```

Here, `<configID>` represents the ID of the configuration. The command creates a new job and waits for the job to be posted on the job queue. The `"job_id"` is returned if the job is posted within about 30 seconds.

### Example:

```
$ ksctl scheduler configs run-now --id "e90cb2df-7df5-487a-ab5e-2c0ead7405c4"
{
 "job_id": "1e05dfb0-a5f9-48f6-afdf-406c368d2847",
 "job_config_id": "e90cb2df-7df5-487a-ab5e-2c0ead7405c4"
}
```

## Deleting Job Configurations

To delete a job configuration, run the command `ksctl scheduler configs delete --id "<configID>"`. Here, `<configID>` is the ID of the job configuration to be deleted. The specified job configuration is deleted without any message.

After a configuration is deleted, it will no longer run jobs automatically.

## Job Runs

A job is created and run automatically when a scheduler configuration is run manually (using `run-now`) or automatically at a schedule defined in the job configuration. You can view the results of jobs executed in the past, including status and error count.

## Viewing Job Runs

To view the list of already run jobs for the logged in account, run the command `ksctl scheduler jobs list`.

### Example:

```

$ ksctl scheduler jobs list
{
 "skip": 0,
 "limit": 10,
 "total": 2,
 "resources": [
 {
 "id": "5ce37ebd-f43f-4a97-bbaf-d6a7416074a7",
 "uri": "kylo:kylo:scheduler:jobs:5ce37ebd-f43f-4a97-bbaf-d6a7416074a7",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2020-04-15T02:25:01.749924Z",
 "name": "rotate AES keys weekly",
 "updatedAt": "2020-04-15T02:25:14.003393Z",
 "job_config_id": "cbf85c25-08e9-4932-93da-a1042aafafdd",
 "description": "This is to rotate AES keys every Sunday, at 02:00",
 "operation": "key_rotation",
 "job_config_params": {
 "meta": {
 "naeState": "Restricted"
 },
 "query": {
 "algorithm": "AES"
 },
 "deactivate_replaced_key": 30
 },
 "enqueued_at": "2020-04-15T02:25:01.749791Z",
 "status": "completed",
 "processing_node": "local-node",
 "details": {
 "job_id": "5ce37ebd-f43f-4a97-bbaf-d6a7416074a7",
 "job_config_id": "cbf85c25-08e9-4932-93da-a1042aafafdd",
 "name": "rotate AES keys weekly",
 "operation": "key_rotation",
 "status": "completed",
 "percent_done": 100,
 "details": {
 "num_rotated": 1
 }
 }
 },
 {
 "id": "5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",
 "uri": "kylo:kylo:scheduler:jobs:5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2020-04-15T02:09:02.740656Z",
 "name": "rotate AES keys",
 "updatedAt": "2020-04-15T02:09:13.978049Z",
 "job_config_id": "61bf1c6f-f1b7-4ef9-b097-af8304647ecf",
 "description": "This is to rotate AES keys",
 "operation": "key_rotation",
 "job_config_params": {
 "meta": {
 "naeState": "Restricted"
 }
 },
 }
]
}

```

```

 "query": {
 "algorithm": "AES"
 },
 "deactivate_replaced_key": 30
 },
 "enqueued_at": "2020-04-15T02:09:02.740554Z",
 "status": "completed",
 "processing_node": "local-node",
 "details": {
 "job_id": "5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",
 "job_config_id": "61bf1c6f-flb7-4ef9-b097-af8304647ecf",
 "name": "rotate AES keys",
 "operation": "key_rotation",
 "status": "completed",
 "percent_done": 100,
 "details": {
 "num_rotated": 1
 }
 }
}
]
}

```

The output shows that two jobs have run successfully.

## Viewing Details of a Particular Job Run

To view the details of a particular job run for the logged in account, run the command `ksctl scheduler jobs get --id`. Here, `--id` is the ID of the job run you are searching for.

### Example:

```

$ ksctl scheduler jobs get --id "5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1"
{
 "id": "5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",
 "uri": "kylo:kylo:scheduler:jobs:5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2020-04-15T02:09:02.740656Z",
 "name": "rotate AES keys",
 "updatedAt": "2020-04-15T02:09:13.978049Z",
 "job_config_id": "61bf1c6f-flb7-4ef9-b097-af8304647ecf",
 "description": "This is to rotate AES keys",
 "operation": "key_rotation",
 "job_config_params": {
 "meta": {
 "naeState": "Restricted"
 },
 "query": {
 "algorithm": "AES"
 },
 "deactivate_replaced_key": 30
 },
 "enqueued_at": "2020-04-15T02:09:02.740554Z",
 "status": "completed",
 "processing_node": "local-node",
 "details": {
 "job_id": "5f2cf5fb-94ef-419a-9cf6-4f4045d43dc1",

```

```

 "job_config_id": "61bf1c6f-flb7-4ef9-b097-af8304647ecf",
 "name": "rotate AES keys",
 "operation": "key_rotation",
 "status": "completed",
 "percent_done": 100,
 "details": {
 "num_rotated": 1
 }
 }
}

```

## Deleting a Job

To delete a job, run the command `ksctl scheduler jobs delete --id`. This command stops the job (if it is running) and deletes it without any confirmation message.

## Licensing

CipherTrust Manager (available as virtual and physical appliances) has transitioned to the Sentinel Entitlement Management System (EMS). The Sentinel platform allows you to activate new features and manage existing licenses for the CipherTrust Manager appliance and Connectors. Use the following self-service License Portal to activate these licenses:

<https://dplicense.prod.sentinelcloud.com/ecp>

## Lock Codes

The Connector Lock Code and Key Manager Lock Code are used for server licenses. The Connector Lock Code is applicable to all nodes of a cluster to enforce Connector licenses across the cluster nodes. Activation is required for each individual CipherTrust Manager appliance using the Key Manager Lock Code.

To activate the purchased Connector license, you need to activate the license for the Connector Lock Code and add it to one of the nodes in the cluster. The License will be replicated to all the nodes in the cluster.

## CipherTrust Manager Licenses

Physical and virtual CipherTrust Manager appliances follow the same licensing model that is explained in this section.

- > CipherTrust Manager version 1.9.0 onward, physical appliances no longer come with default 90-days trial licenses. They are licensed "out of the box." Separate connectors licenses are required on these appliances.
- > For CipherTrust Manager versions lower than 1.9.0, physical appliances come with default 90-days trial licenses, which is replaced by a perpetual license as part of the appliance sale.

### NOTE

- > CipherTrust Manager licenses are node locked, that is, each node in a CipherTrust Manager cluster requires a separate license.
- > Every clone of a CipherTrust Manager appliance requires a separate license.

## Installing a CipherTrust Manager License

You need a valid and active license on your CipherTrust Manager appliance. Installing a CipherTrust Manager license requires a license string for the CipherTrust Manager. This string is generated when a license is activated on the Sentinel EMS License Portal.

To install a CipherTrust Manager license:

1. Log on to the CipherTrust Manager console as administrator.

2. Click **Admin Settings > Licensing**.

A list of licenses is displayed on the right.

3. Click the **Lock codes** tab and copy the **Key Manager Lock Code**.

Every CipherTrust Manager appliance has a unique key manager lock code. This lock code is needed to activate a CipherTrust Manager license.

4. Activate the CipherTrust Manager 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 CipherTrust Manager license:

- a. Log on to the [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 CipherTrust Manager 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 [Key Manager Lock Code](#) copied earlier.
- g. Click **Activate**.

### NOTE

- > Based on the requirements, a CipherTrust Manager 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 CipherTrust Manager license on the CipherTrust Manager.

5. Return to the CipherTrust Manager console.

6. Under the **Add license** section, in the **License String** field, paste the copied [License String](#).

7. Click **Add**. The license is now installed. The **Virtual\_KeySecure** license is displayed in the installed licenses table.



## Connector Licenses

CipherTrust Manager supports enforced Connector licenses through self-service License Portal on the Sentinel platform. Registering new clients, running new scans and generating new reports, and enabling KMIP client communications require active Connector licenses on the CipherTrust Manager appliance.

**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 Connector license is activated for one CipherTrust Manager appliance, the license is replicated on all nodes of the cluster.

Refer to the "Licensing" sections in Connector documentation for information on their licensing models and instructions to install the license.

## License Enforcement

This section describes license enforcement on Connectors and on KMIP client communication.

### ProtectApp, ProtectFile, and ProtectV

- > **CipherTrust Manager appliance has activated Connector licenses:** When Connector licenses are activated and uploaded to a CipherTrust Manager, you can register clients to license capacity. The number of clients that you can register cannot exceed the Connector license count.
- > **Reaching license capacity:** If you attempt to register additional clients, registration fails because the license count has been exhausted. In this case, delete currently configured clients or buy additional licenses to register new clients.
- > **License expires:** The CipherTrust Manager GUI displays a red banner to inform the administrator of expired licenses. At this time, no new client registration is allowed. Currently registered clients can still be managed for 90 days from license expiry. After 90 days, changes on currently registered clients are restricted and only decryption of data is allowed.

### Data Discovery and Classification (DDC)

- > **CipherTrust Manager appliance has trial license activated:** 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 30 days and up to the 2 TB of data allowance.
- > **Data allowance is used up:** You can continue scanning but cannot generate reports. However, the data from scans is stored so after you install a new license, you can access the data and generate reports.
- > **License expires:** The DDC configuration on CipherTrust Manager becomes read-only. While you still have access to your old reports, you cannot generate new reports, add new targets, or create new scans. The data collected so far is not deleted, so you can access it when you install a new license.

## KMIP

- > **CipherTrust Manager appliance has activated KMIP license:** Valid KMIP license enables the KMIP feature on the CipherTrust Manager. KMIP clients can communicate with the CipherTrust Manager.

- > **License expires:** A red Banner appears on the CipherTrust Manager GUI to inform the administrator of expired licenses. KMIP feature (from API) will show as expired. All KMIP communications fail.

**NOTE** KMIP client license count is not yet enabled. An active KMIP license on the CipherTrust Manager appliance will enable the feature and any KMIP client can communicate with the CipherTrust Manager appliance. As soon as the license expires, all KMIP communication will fail.

- 30 days before a license expires, an orange banner appears on the CipherTrust Manager GUI, as a system message on every page to inform the administrator of the license status.
- A red banner is displayed, when one or more licenses are expired. When an administrator navigates through the CipherTrust Manager GUI, the red banner appears as a system message at the top of every page.

## Viewing Installed Licenses

The installed licenses table displays the complete list of additional features running on the CipherTrust Manager. The following information is displayed:

- > **Feature:** Name of the licensed Connector or appliance. The supported features include Virtual\_KeySecure, DDC\_DATA\_ALLOWANCE, KMIP, LiveDataTransformation, ProtectApp, ProtectFile, ProtectV, and TransparentEncryption.
- > **State:** State of the license.
- > **Expiration:** License expiration date.

## Deleting Installed Licenses

To delete an installed license:

1. Navigate to the installed licenses table (**Admin Settings > Licensing**).
2. Click the ellipsis icon corresponding to the license that you want to delete and click **View/Edit**.
3. In the **LICENCES** screen that is displayed for the selected license, click the **Delete** active link to delete the license.

## System Logs

The CipherTrust Manager maintains a variety of logs to record administrative actions, network activity, cryptography requests, and more.

The following types of logs are recorded:

- > Connection request received
- > Configuration changes
- > Client requests
- > Service starts, stops, and restarts
- > System and user input errors

### > Successful and failed operations

These logs can be useful for debugging, error handling, troubleshooting, and for chronologically tracing failures and system events. The logs can be downloaded using the CipherTrust Manager GUI. The downloaded file contains following files in *.gzip* format:

- > hostd.log
- > cloud-init.log
- > keysecuresystem.log

If it is desired to download the rotated logs as well, enable the **Download all logs** option. The downloaded log file will additionally contain rotated logs that are up to 4 weeks old.

## Downloading Logs

Perform the following steps to download these logs using the GUI:

1. Navigate to **Admin Settings > Logs**.
2. Select a CA for signing the logs.
3. Download the logs:
  - To download the **System Logs**, click **Download**.
  - To download the **rotated logs as well**, enable the **Download all logs** option and then click **Download**.

**NOTE** Only the users who are part of **System Admins** and **Admin** groups can download the logs from the root domain.

## Preserving integrity of downloaded logs

CipherTrust Manager preserves integrity of downloaded logs by performing following steps:

- > SHA256 hash of downloaded zip file is calculated, which is then signed by a dynamically generated asymmetric key pair.
- > Certificate is issued by the CipherTrust Manager CA. This CA is selected while downloading the logs, as explained [here](#).
- > Certificate to verify the signed hash is contained in the downloaded content.

## Verifying and viewing the downloaded logs

You must have installed OpenSSL on your machine to verify the log file. You can use the procedure in both Windows and UNIX/Linux environments.

Perform the following steps on the downloaded log file to verify its integrity:

1. Unzip (extract) the log file.
2. Use the following commands:
  - a. `openssl x509 -pubkey -noout -in keySecure.cert > pubkey.pem`
  - b. `openssl dgst -sha256 -verify pubkey.pem -keyform PEM -sha256 -signature keySecure.signature keySecureLogs.tar.gz`

3. Depending on your operating system, use following steps to view the downloaded logs:

- **For Windows users**

**NOTE** Do not use WinRAR for extracting the log files. Use of 7-Zip is recommended.

- Extract the `keySecureLogs.tar.gz` file using 7-Zip.  
You will get three logs file: `cloud-init.log`, `hostd.log`, `keyecure.system.log`.
- Open these files in a reader to view the logs.

- **For UNIX/Linux users**

Use the following commands in the Terminal:

- Decompress the `keySecureLogs.tar.gz` using this command:  

```
gzip -d keySecureLogs.tar.gz
```

  
You will get `keySecureLogs.tar`.
- Extract the log files using this command:  

```
tar xvf keySecureLogs.tar
```

  
You will get three logs file: `cloud-init.log`, `hostd.log`, `keyecure.system.log`.

## System Upgrade/Downgrade

You can upgrade/downgrade your Thales CipherTrust Manager by securely downloading and applying a new/older system archive file.

**NOTE** Refer to "[Cluster Upgrade](#)" on page 43 for details on cluster upgrades.

### System Upgrade

#### To download a system upgrade file

System upgrades are supplied in the form of a signed archive file available from the Support Portal.

1. Download the archive file to CipherTrust Manager:

```
$ scp -i <ssh private key path> <update file name> ksadmin@<ip>:.
```

**NOTE** The signature of the archive file is verified before the upgrade/downgrade is performed.

#### To upgrade your CipherTrust Manager

**NOTE** Upgrades keep all the data and may migrate the data and configuration. Therefore, as a precaution, it is recommended to take a backup before upgrading.

1. SSH into the CipherTrust Manager as "ksadmin".
2. Upgrade the CipherTrust Manager:

```
$ sudo /opt/keysecure/ks_upgrade.sh -f <~/filename>
```

Here, <~/filename> specifies path to the signed CipherTrust Manager installer file.

## System Downgrade

CipherTrust Manager 2.0.0 can be downgraded to 1.10.0. For release-specific upgrade/downgrade information, refer to the release notes for your release.

### To downgrade your CipherTrust Manager

**NOTE** Downgrades perform a CipherTrust Manager reset, which wipes all data except the backup files that already exist.

1. SSH into the CipherTrust Manager as "ksadmin".

2. Downgrade the CipherTrust Manager:

```
$ sudo /opt/keysecure/ks_downgrade.sh -f <~/filename> -y
```

Usage: ks\_downgrade.sh -f <FILE> [-o] [-y]

- -f: Path to the signed CipherTrust Manager installer file.
- -o: Clustered node cannot be downgraded. Use this flag to override this behavior.
- -y: Skip the confirmation prompt for CipherTrust Manager reset.

## Client Management

A client represents a machine where a supported product such as ProtectFile, ProtectV, and CTE is installed. A client needs to be registered with the Thales CipherTrust Manager to retrieve and store encryption keys to encrypt and decrypt data. When successfully registered, the client is automatically added to the CipherTrust Manager.

A client can be registered with the CipherTrust Manager by using a registration token generated on the CipherTrust Manager. This token is called the client registration token. The fingerprint of the server's web interface certificate is also required for registering clients with the CipherTrust Manager.

The CipherTrust Manager provides options to manage registration tokens and clients.

### Tokens

A token is a string that is used to register clients having ProtectFile, ProtectV, CTE, and CTE UserSpace installed with the CipherTrust Manager. The CipherTrust Manager provides options to create tokens, view existing tokens, view and modify their details, and delete them when they are no longer required.

The following table lists the parameters that are required when creating or managing a registration token on the CipherTrust Manager:

| Parameter                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Certificate Authority ID | <p>ID of the trusted CA that will be used to sign a client certificate during the registration process. By default, a local CA will be used to issue certificates.</p> <p>A signed Local CA certificate is used to secure communication between the CipherTrust Manager and clients. To use an external CA certificate, specify the ID of the Local CA signed by the external CA.</p>                                                                                                                                                                                                                                                                                                                                          |
| Lifetime                 | <p>Duration (in minutes, hours, or days) for which this token can be used for registering clients. Specify <code>m</code> for minutes, <code>h</code> for hours, and <code>d</code> for days. For example, specify the lifetime as:</p> <ul style="list-style-type: none"> <li>&gt; <code>xm</code> for <code>x</code> minutes</li> <li>&gt; <code>xh</code> for <code>x</code> hours</li> <li>&gt; <code>xd</code> for <code>x</code> days</li> </ul> <p>By default, there is no time limit; the token can be used forever.</p>                                                                                                                                                                                               |
| Certificate Duration     | Duration (in days) for which the CipherTrust Manager client certificate is valid. The default duration is 730 days.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Maximum Clients          | <p>Maximum number of clients that can be registered using this registration token.</p> <p>By default, there is no limit; any number of clients can be registered using this token.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Name Prefix              | <p>Prefix for the client name. This prefix will be used to construct names for clients whose names will not be specified during registration with the CipherTrust Manager using this token.</p> <ul style="list-style-type: none"> <li>&gt; If the name prefix is specified as <code>ks_client</code>, client names will be constructed as <code>ks_client#</code>; for example, <code>ks_client1</code>, <code>ks_client2</code>, <code>ks_client3</code>, and so on.</li> <li>&gt; If the name prefix is not specified, the CipherTrust Manager will construct a random name for clients.</li> </ul> <p>However, if a client's name is specified during registration, this name prefix will not be used for that client.</p> |
| Revocation Reason        | Reason for revoking the client registration.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

## Getting the Fingerprint of the Server Certificate

Fingerprint of the server's web interface certificate can be viewed on the API playground or on the GUI. This topic describes steps to get fingerprint on the API playground.

To get the fingerprint:

1. Acquire an authorization token.
2. In the left pane of the API playground, click **Client-Management/Tokens**.
3. Under **Client-Management/Tokens**, click **Web Certificate Fingerprint**. The **Web Certificate Fingerprint** section of the API playground is displayed in the right pane.
4. Click **GET**. View the fingerprint details in the response output.

The SHA256 and SHA512 fingerprints of the server's web interface certificate are displayed.

## Clients

All clients with supported product installations (for example, ProtectFile, ProtectV, and CTE) can be managed on the CipherTrust Manager. A CipherTrust Manager Administrator can register clients (except ProtectFile clients) with the CipherTrust Manager, view registered clients, view and modify details, revoke registrations, and delete clients when they are no longer needed.

The following table lists the parameters that are required when managing a client on the CipherTrust Manager:

| Parameter                         | Description                                                             |
|-----------------------------------|-------------------------------------------------------------------------|
| Name                              | Name for the client to display on the CipherTrust Manager.              |
| Registration Token                | Registration token to register the client with the CipherTrust Manager. |
| Certificate Signing Request (CSR) | CSR to be signed by the CipherTrust Manager.                            |

**\*\*WARNING\*\*** As soon as a client is deleted from the CipherTrust Manager, all communication between the CipherTrust Manager and the client will stop immediately. It is recommended that the clients are decrypted before deletion, otherwise the encrypted data will become inaccessible.

## Connection Manager

The Connection Manager contains a list of all connections to the resources that are external to the CipherTrust Manager server. Any resource that you intend to use with CipherTrust Manager and that resides outside of the CipherTrust Manager infrastructure has to be added using the Connection Manager.

**NOTE** In the first phase of the Connection Manager implementation only the connections for CipherTrust Cloud Key Manager (CCKM) to Amazon Web Services (AWS) and Microsoft Azure will be configured from there. Both are cloud computing platforms and CCKM manages cloud keys for those cloud services.

## Accessing the Connection Manager

To access the Connection Manager, log in to CipherTrust Manager as administrator. Next, click **Keys & Access Management** on the main screen, and then select **Connections** from the sidebar on the left.

The **Connections Management** screen is displayed. The tabular view lists all the currently configured connections. You can arrange the list in different orders, by clicking on the column headers to sort it by that column. The table has the following columns:

- > **Name** - the name of the connection
- > **Creation** - the date when the connection was created
- > **Type** - the type of connection (currently, only AWS or Azure)

- > **Products** - the name of the product that uses the connection (currently, only CCKM)
- > **State** - the state that the connection is in. It is one of these states:
  - Not tested - the connection has not been tested
  - Fail - the connection has been tested and the test failed. The date when the connection failed is displayed to the right
  - Ready - the connection is properly configured and ready

The last column contains an ellipsis button (...) which, when you click it, displays a menu that allows you to edit or delete the corresponding connection.

Use the filters in the column headers to filter through multiple connections and display only those that you wish to display.

Use the **Search** box to search for a specific connection.

Click the **+ Add Connection** button to add a new connection.

## Adding a New Connection

Click the **+ Add Connection** button to open the **Add Connection** wizard. The wizard consists of these four steps:

- > Select Store Type
- > General Info
- > Configure Connection
- > Add Products

### 1. Select Store Type

In the **Select Category** section, click one of the tiles. For now, only a Cloud type category is available.

From the **Select Cloud Type** menu select AWS or Azure. Click **Next** to move on to the next step.

### 2. General Info

In this step provide a **Name** and **Description** (optional) for the new connection. Click **Next** to move on to the next step.

### 3. Configure Connection

In this step you configure your connection. Depending on the cloud type that you selected in step one - AWS or Azure - certain options may look slightly different.

#### Amazon Web Services

- **Access Key ID** - an access key is a long-term credential for an AWS account root user. An access key consists of two parts: an access key ID and a secret access key (pretty much like a user name and password). This is the user name part.
- **Secret Access Key** – this is the password part of the access key. Select the **Show Secret Access Key** check box to view the password as open text.



- **Cloud Name** - the name of the AWS cloud to connect to. Currently, only the following options are available:
  - AWS
  - AWS-US-GOV
  - AWS-CN
- **Assume Role** - the AWS Assume Role. For more details, refer to the AWS documentation, (see "Identity and Access Management (IAM) role").
- **Assume Role External ID** - the external ID for an IAM role in AWS (Assume Role). This is something that you may want to use when you need to give access to your AWS resources to a third party. For more details, refer to the AWS documentation (see "Identity and Access Management (IAM) role").

## Microsoft Azure

- **Access Key ID** - an access key is a long-term credential for an Azure account root user. An access key consists of two parts: an access key ID and a client secret (pretty much like a user name and password). This is the user name part.
- **Client Secret** – this is the password part of the access key. Select the **Show Secret Access Key** check box to view the password as open text.
- **Tenant ID** - this is your Office 365 tenant ID. It is a globally unique identifier (GUID). For more details, refer to the Azure documentation.
- **Cloud Name** - the name of the Azure cloud to connect to. Currently, only the following options are available:
  - Azure Cloud
  - Azure China Cloud
  - Azure German Cloud
  - Azure US Government

Click the **Test Connection** button to test if you have configured your connection correctly. Click **Next** to move on to the next step.

## 4. Add Products

Use the checkboxes in the **Products** list to select a product associated with the connection. Currently, only CipherTrust Cloud Key Manager (CCKM) is supported.

Click **Save** to save your connection. The new connection is now listed in the **Connections Management** table.

## Editing a Connection

1. Navigate to the **Connections Management** screen and find a connection that you want to edit.
2. Click the ellipsis button (...) to display a pop-up menu.
3. Select **Edit** from the menu.
4. Go over the steps of the connection wizard and update the connection as desired.

## Deleting a Connection

1. Navigate to the **Connections Management** screen and find a connection that you want to delete.
2. Click the ellipsis button (...) to display a pop-up menu.
3. Select **Delete** from the menu.

The connection is removed from the list of connections.

## Configuring DNS Hosts

CipherTrust Manager allows its users to specify DNS records, eliminating the need to remember the IP addresses. While specifying a hostname, its IP address is required so that CipherTrust Manager can connect directly to that address whenever its hostname is specified.

Hostname can be specified using:

### > API playground

Refer to the **DNS Hosts** section.

### > CLI tool (ksctl)

Command to create a DNS record "example.com" with IP address "12.12.xx.xx"

```
ksctl dnshosts create --name "example.com" --ip "12.12.xx.xx"
{
 "id": "07eb1e56-8b22-4c72-a6a2-464f24f590a1",
 "uri": "kylo:kylo:solo:dns:07eb1e56-8b22-4c72-a6a2-464f24f590a1",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2020-08-17T09:40:06.951331083Z",
 "name": "example.com",
 "ip": "12.12.xx.xx",
 "updatedAt": "2020-08-17T09:40:06.951331083Z"
}
```

---

# Operation

---

This chapter describes in detail concepts relevant to the operation of the Thales CipherTrust Manager. It contains the following sections:

- > ["CLI toolkit" on page 109](#)
- > ["Keys" on page 120](#)
- > ["Crypto Operations" on page 127](#)
- > ["Certificate Authority" on page 129](#)
- > ["Alarms" on page 133](#)

# CLI toolkit

---

This chapter describes how to install the CLI toolkit **ksctl** and use it to perform typical administrating tasks on Thales CipherTrust Manager VM or instance. It contains the following sections:

- > ["Installation" below](#)
- > ["Try it!" on page 113](#)
- > ["Batching Commands with Tokens" on page 117](#)
- > ["Support CLI" on page 119](#)

## Installation

Thales CipherTrust Manager includes a CLI toolkit, named **ksctl**, that can be downloaded and run locally to control a remote CipherTrust Manager.

**NOTE** **ksctl** is designed to be run from a remote system, not on the CipherTrust Manager itself.

**ksctl** exclusively uses the REST API to communicate with CipherTrust Manager, so anything you can do with the CLI 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 adding users and groups, and end-user functions, such as creating keys.

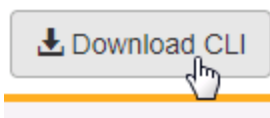
To get started with **ksctl** you must do the following:

1. Download and unzip the **ksctl\_images.zip** file.
2. Setup the *ksctl-os* file for your system.
3. Run **ksctl**.

---

### To download and unzip the **ksctl\_images.zip** file:

1. Enter the IP address of your Thales CipherTrust Manager system in your browser.
2. On the Thales CipherTrust Manager start page, select **API & CLI Documentation** link.
3. At the top of the **API & CLI Documentation** page, click on **Download CLI** button:



4. Unzip the **ksctl\_images.zip** file.

Example set of available images in **ksctl\_images.zip** file:

```
ksctl-darwin-amd64
ksctl-linux-amd64
ksctl-win-amd64.exe
```

---

## To setup the **ksctl-os** for your system:

### Windows

1. After unzipping, rename either `ksctl-win-386.exe` or `ksctl-win-amd64.exe` to `ksctl.exe`.

**NOTE** **ksctl.exe** is a single executable with no dependencies, so it can be run from anywhere.

2. Move the **ksctl.exe** file to a folder easily accessed by the local command prompt.

**NOTE** You can also add the location of the **ksctl.exe** file to your PATH variable or copy **ksctl.exe** to a location already in your PATH.

3. Create a **ksctl** configuration file using the local command prompt **cmd.exe**:

- a. To find or confirm your **%HOMEPATH%** directory, enter:

```
$ set homopath
```

- b. Create a **.ksctl** directory in your **%HOMEPATH%** directory:

```
$ mkdir %HOMEPATH%\ksctl
$ cd %HOMEPATH%\ksctl
```

- c. In the **%HOMEPATH%\ksctl** directory, create a **config.yaml** file with the following items:

```
KSCTL_USERNAME: admin
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<ciphertrust_ipaddress>
KSCTL_NOSSLVERIFY: true
```

### Linux

1. After unzipping, rename the file of choice to **ksctl** and move it to a directory within your PATH.

**NOTE** **ksctl** is a single executable with no dependencies, so it can be run from anywhere.

2. Create a **ksctl** configuration file:

- a. Create a **.ksctl** directory in your **\$HOME** directory:

```
$ mkdir $HOME/.ksctl
$ cd $HOME/.ksctl
```

- b. In the **\$HOME/.ksctl** directory create a **config.yaml** file with the following items:

```
KSCTL_USERNAME: admin
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<ciphertrust_ipaddress>
KSCTL_NOSSLVERIFY: true
```

3. Bash completions can be generated by typing `ksctl bashcomp` at the bash prompt and following the instructions.

---

## To override **ksctl** configuration file

Summary of **ksctl** configuration variables for the `config.yaml` configuration file:

---

```
KSCTL_USERNAME: <your username>
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<ciphertrust_ipaddress>
KSCTL_NOSSLVERIFY: <true or false>
KSCTL_CONNECTION: <LDAP connection name>
KSCTL_VERBOSITY: <true or false for verbosity>
KSCTL_JWT: <JWT>
KSCTL_RESPFMT: <response output format, the default "json" is the only supported value at present>
KSCTL_TIMEOUT: <timeout in seconds for TCP connection attempts>
```

**ksctl** is easiest to use with the configuration file (config.yaml). However, those parameters can be overridden by using these flags on the command line:

```
--user (replaces KSCTL_USERNAME)
--password (replaces KSCTL_PASSWORD)
--url (replaces KSCTL_URL)
--nosslverify (replaces KSCTL_NOSSLVERIFY)
```

**NOTE** If you wish to override KSCTL\_NOSSLVERIFY it must be set to false in the configuration file (or removed).

```
--connection (replaces KSCTL_CONNECTION)
--verbose (replaces KSCTL_VERBOSITY)
--jwt (replaces KSCTL_JWT)
--respfmt (replaces KSCTL_RESPFMT)
--timeout int (replaces KSCTL_TIMEOUT)
```

---

## To run ksctl:

**NOTE** This is an example only; not all supported CLI commands and flags are shown.

1. At the command line prompt, enter **ksctl**. You will get the following example output:

```
$ ksctl
This command line utility can be used to interface with a CipherTrust Manager system.
```

```
Configuration parameters can be used to pass in the username, URL, etc.
These parameters can be passed to the application in several ways.
```

```
When the configuration parameters are available from multiple sources,
the order of precedence (highest to lowest) is:
```

- (a) command line arguments,
- (b) configuration file passed via the command line,
- (c) environmental variables, and
- (d) configuration file in the default location.

```
The parameters passed in as command line arguments override
everything else.
```

```
The structure of the configuration file must be based on
'config_example.yaml'. The name of the default configuration file is
'config.yaml'. It should be located in the '.ksctl' subdirectory of the
user's home directory. On Linux, this file should be located here: '~/.ksctl/con
fig.yaml'.
```

---

If defined, proxy environment variables (i.e. `http_proxy`) will be honored for all functions except `'init getpw'`. `'init getpw'` internally uses `ssh` which means that `ssh` must be configured to be proxy aware for the users local operating system.

Usage:

`ksctl [command]`

Available Commands:

|                          |                                                                            |
|--------------------------|----------------------------------------------------------------------------|
| <code>alarms</code>      | Manage Alarms. List, state and clear commands.                             |
| <code>backup</code>      | Perform system or domain database backups.                                 |
| <code>backupkeys</code>  | Manage Backup keys. Used for securing backups.                             |
| <code>bashcomp</code>    | Generate bash completion script file for all commands and flags.           |
| <code>ca</code>          | Manage Certificate Authorities and Certificates.                           |
| <code>changepw</code>    | Change a user password.                                                    |
| <code>clientmgmt</code>  | Create Client Tokens and Register Clients.                                 |
| <code>cluster</code>     | Manage Clusters.                                                           |
| <code>connections</code> | Manage authentication connections                                          |
| <code>crypto</code>      | Cryptographic data processing commands that use keys stored in the system. |
| <code>diskenc</code>     | Disk encryption configuration, query and unlocking commands.               |
| <code>gendocs</code>     | Generate man pages and documentation for all commands and flags.           |
| <code>.</code>           |                                                                            |
| <code>.</code>           |                                                                            |
| <code>.</code>           |                                                                            |

Flags:

|                                  |                                                                                                                               |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <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 <code>"local_account"</code> )                     |
| <code>-h, --help</code>          | help for ksctl                                                                                                                |
| <code>--jwt string</code>        | The JSON web token (JWT) can be passed instead of user/password (optional). <code>'ksctl tokens create'</code> creates a JWT. |
| <code>--nosslverify</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 <code>"json"</code> )                 |
| <code>--timeout int</code>       | Timeout in seconds for TCP connection attempts                                                                                |
| <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)                                                                     |

Use `"ksctl [command] --help"` for more information about a command.

## 2. Try creating a key by entering the command `ksctl keys create --autoname`:

```
$ ksctl keys create --autoname
```

This will create output similar to the following:

```
{
 "id": "b979d9c7-1374-4b97-a274-dfaa20f8139b",
 "uri": "kylo:kylo:vault:keys:keys-f0f3ellhgh5r7xavk3l0-rx-ktkmwgzf3-rudppxrw4-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
```

```
"createdAt": "2016-12-15T15:35:51.322996318Z",
"name": "Keys-f0F3eLlhQh5R7XAVK310_rx-KtkmWGzF3-rUDPpXRW4",
"updatedAt": "2016-12-15T15:35:51.322996318Z",
"material": "",
"usage": "blob",
"meta": null,
"version": 0,
"algorithm": "AES",
"size": 256,
"format": "raw",
"unexportable": false,
"undeletable": false,
"publickey": ""
}
```

## Try it!

The following example will create a non-admin user named **sarah** and create a key. **sarah** will own the key and will be able to do things with it.

We will create a second user, named **bob**, and show he does not have access to **sarah's** key.

Next, we will add **bob** to a group. **sarah** will add a permission to the key, granting members of the group some permissions to the key. We'll then show that **bob** now has these permissions.

You can try these steps using the API playground, curl, or by using the CLI commands.

This exercise assumes that the user has already installed the CLI tool (`ksctl`) and has created a `config.yaml` file that is specific to their installation. Note that we will use the built-in user in some cases and override them in others. Any command without user/password credentials is using the built-in (admin) user.

### 1. Create a new user called **sarah**.

```
$ ksctl users create -n sarah -p Sarah_pw1
```

The response to the create user command is a user resource that looks like:

```
{
 "created_at": "2018-04-27T21:10:53.191577Z",
 "email": "",
 "last_login": null,
 "logins_count": 0,
 "name": "bob",
 "nickname": "bob",
 "updated_at": "2018-04-27T21:10:53.191577Z",
 "user_id": " local|0c617800-58f0-4cf9-85bd-8d75037d8bfb",
 "username": "sarah",
 "failed_logins_count": 0,
 "account_lockout_at": null,
 "failed_logins_initial_attempt_at": null,
 "last_failed_login_at": null,
 "password_changed_at": "2018-04-27T21:10:53.192083Z",
 "password_change_required": false
}
```

### 2. Now create an AES key, which will be owned by **sarah**.

We'll need to create a file containing the attributes of the key we want to create, then pass that file to the CLI tool:

## Windows



---

```
$ echo {"name":"aeskey"} > keyparams.json
```

## **linux**

```
$ echo "{\"name\":\"aeskey\"}" > keyparams.json
```

## **All systems**

```
$ ksctl --user sarah --password sarah_pwl keys create -j keyparams.json
```

The response to the create key command is a key resource that looks like:

```
{
 "id": "cf8c8c215e8077e11a7af5304fe77082791af13e3906d52ba7237eb6fd2df868",
 "uri": "kylo:kylo:vault:keys:aeskey-v0-HEU",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2018-04-27T20:56:11.731123593Z",
 "name": "aeskey",
 "updatedAt": "2018-04-27T20:56:11.731123593Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|0c617800-58f0-4cf9-85bd-8d75037d8bfb"
 },
 "algorithm": "AES",
 "size": 256,
 "format": "raw",
 "unexportable": false,
 "undeletable": false,
 "defaultIV": "5bec41b53dd4a64aa25d64c175b05fb0",
 "objectType": "Symmetric Key",
 "activationDate": "2018-04-27T20:56:11.729121368Z",
 "state": "Active"
}
```

### **3. sarah is the owner of the key, which grants her permission to read the key:**

```
$ ksctl keys get -n aeskey
```

The response looks like:

```
{
 "id": "cf8c8c215e8077e11a7af5304fe77082791af13e3906d52ba7237eb6fd2df868",
 "uri": "kylo:kylo:vault:keys:aeskey-v0-HEU",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2018-04-27T20:56:11.731124Z",
 "name": "aeskey",
 "updatedAt": "2018-04-27T20:56:11.731124Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|0c617800-58f0-4cf9-85bd-8d75037d8bfb"
 },
 "algorithm": "AES",
 "size": 256,
 "format": "raw",
 "unexportable": false,
 "undeletable": false,
 "defaultIV": "5bec41b53dd4a64aa25d64c175b05fb0",

```

```
 "objectType": "Symmetric Key",
 "activationDate": "2018-04-27T20:56:11.729121Z",
 "state": "Active"
}
```

#### 4. Now we'll create a second user, named **bob**:

```
$ ksctl users create -n bob -p bob_password1
```

The response looks like:

```
{
 "created_at": "2018-04-27T21:10:53.191577Z",
 "email": "",
 "last_login": null,
 "logins_count": 0,
 "name": "bob",
 "nickname": "bob",
 "updated_at": "2018-04-27T21:10:53.191577Z",
 "user_id": "local|c7560d06-3628-4fb0-81e3-5881d2f8d7b3",
 "username": "bob",
 "failed_logins_count": 0,
 "account_lockout_at": null,
 "failed_logins_initial_attempt_at": null,
 "last_failed_login_at": null,
 "password_changed_at": "2018-04-27T21:10:53.192083Z",
 "password_change_required": false
}
```

#### 5. We'll also create a new group **eng** and put **bob** in it:

```
$ ksctl groups create -i "local|c7560d06-3628-4fb0-81e3-5881d2f8d7b3" -n eng
```

The response looks like:

```
{
 "name": "eng",
 "created_at": "2018-04-27T21:15:36.644959Z",
 "updated_at": "2018-04-27T21:15:36.644959Z"
}
```

#### 6. We will connect as **bob** and try to read the key. This will fail, since **bob** has no access to the key.

```
$ ksctl --user bob --password bob_password1 keys get -n aeskey
```

The response looks like:

```
{
 "code": 16,
 "codeDesc": "NCERRResourceNotFound: Resource not found"
}
```

#### 7. Let us connect as **sarah** and give the **eng** group, of which bob is a member, permission to get info about the key.

##### Windows

```
$ echo {"meta":{"permissions":{"ReadKey":["eng"]}}} > keypermissions.json
```

##### linux

```
$ echo "{\"meta\":{\"permissions\":{\"ReadKey\":[\"eng\"]}}}" > keypermissions.json
```

##### All systems

```
$ ksctl --user sarah --password sarah_pw1 keys modify -n aeskey -j keypermissions.json
```

---

The response looks like:

```
{
 "id": "cf8c8c215e8077e11a7af5304fe77082791af13e3906d52ba7237eb6fd2df868",
 "uri": "kylo:kylo:vault:keys:aeskey-v0-HEU",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2018-04-27T20:56:11.731124Z",
 "name": "aeskey",
 "updatedAt": "2018-04-27T21:23:05.713305295Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|0c617800-58f0-4cf9-85bd-8d75037d8bfb",
 "permissions": {
 "ReadKey": [
 "eng"
]
 }
 },
 "algorithm": "AES",
 "size": 256,
 "format": "raw",
 "unexportable": false,
 "undeletable": false,
 "defaultIV": "5bec41b53dd4a64aa25d64c175b05fb0",
 "objectType": "Symmetric Key",
 "activationDate": "2018-04-27T20:56:11.729121Z",
 "state": "Active"
}
```

## 8. Now bob is able to get info about the key.

```
$ kscctl -user bob -password bob_password1 keys get -n aeskey
```

The response looks like:

```
{
 "id": "cf8c8c215e8077e11a7af5304fe77082791af13e3906d52ba7237eb6fd2df868",
 "uri": "kylo:kylo:vault:keys:aeskey-v0-HEU",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2018-04-27T20:56:11.731124Z",
 "name": "aeskey",
 "updatedAt": "2018-04-27T21:24:29.898123Z",
 "usage": "blob",
 "usageMask": 12,
 "meta": {
 "ownerId": "local|0c617800-58f0-4cf9-85bd-8d75037d8bfb",
 "permissions": {
 "ReadKey": [
 "eng"
]
 }
 },
 "algorithm": "AES",
 "size": 256,
 "format": "raw",

```

```

 "unexportable": false,
 "undeletable": false,
 "defaultIV": "5bec41b53dd4a64aa25d64c175b05fb0",
 "objectType": "Symmetric Key",
 "activationDate": "2018-04-27T20:56:11.729121Z",
 "state": "Active"
 }

```

## Batching Commands with Tokens

As described in section ["Tokens" on page 45](#), a token is returned after logging in to the CipherTrust Manager, for example, using the `config.yaml` file or the `--user` and `--password` command line parameters. This token, when used with subsequent commands, can improve performance with the CipherTrust Manager, since verifying the token is much faster than looking up the user and password on each invocation.

If the user wishes to perform batched commands and desires greater performance, you can use this token as described in this section.

### Download and install the jq program

Since we are dealing with `.json` responses, the first step is to download and install the `jq` program from <https://stedolan.github.io/jq/download/>. This program parses information from returned `json` parameters, allowing the user to manipulate them as needed. Follow the instructions on this page for installation on your operating system and be sure that `jq` is in your `PATH`.

### Utilizing tokens to perform operations

You can now retrieve a token for a user on the platform and by utilizing this information, perform operations on the CipherTrust Manager. We will detail connecting as a user that has already been created and using the token from that user to perform a key creation on the CipherTrust Manager.

**NOTE** The environment variable `KSCTL_JWT` is utilized by **ksctl** for accessing the token used. Refer to the CLI > ["Installation" on page 109](#) for other variables utilized by **ksctl**, available either in the environment or the `config.yaml` file. In addition, the examples below are specific to `*nix` operating systems; small differences will be necessary if running under Windows. The CLI > ["Installation" on page 109](#) has sections that detail these differences.

We will now get a token and `user_id` from a previously created user, Sarah. These will be used to create a key. The key creation step can be repeated numerous times to create as many keys as you want and are able to support.

### Get Sarah's token

#### 1. To retrieve a user's token:

```
$ ksctl tokens create --user sarah --password Sarah_pw1
```

Returns the following response:

```

{
 "jwt":
 "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiIzNjU3OWYyYi03ZmUyLTRlYWYtOTQ3Ni03ZGU0MmNjYjFlYTkiLCJzdzIiOiJsb2NhbnHw3N2JmMTI0Yy1jNGYyLTRhODktYTc3Ny1iZTlmOGM2NmQwOTkiLCJpc3MiOiJreWxvIiwiaWF0IjYWNj

```

```
Ijoia3lsbyIsInByZWZlcnJlZF91c2VybmFtZSI6InNhcmFoIiwia3VzdCI6eyJncm91cHMlOlsiS2V5IFVzZXJzIl19LCJq
d3RpZCI6ImJlNTI5NTIzLWE0YzktNGMzZS05ZTdhdjZDZmNzA4Y2I5YiIsImhhdCI6MTUyOTY5NDMzMzMywizXhwIjoxNT
I5Njk0NjMzZfQ.0z-x_lIue37Jju94ZsU1UvTR0s09ick7BT2N6uPmAoo",
 "duration": 300,
 "token_type": "Bearer"
}
```

## 2. To assign the token to the KSCTL\_JWT environment variable, enter:

```
$ export KSCTL_JWT=$(ksctl tokens create --user sarah --password Sarah_pwl | jq -r ".jwt")
```

**NOTE** If you have user `sarah` defined in your `config.yaml` file, the `--user` and `--password` variables are not needed.

For the purposes of this example, Sarah's `user_id` must be retrieved; it is used in the creation of a key.

## Get Sarah's id

### 1. To retrieve Sarah's `user_id`, enter:

```
$ ksctl users list --name sar
```

Returns the following response:

```
{
 "skip": 0,
 "limit": 10,
 "total": 1,
 "resources": [
 {
 "created_at": "2017-06-22T16:25:56.441357Z",
 "email": "",
 "last_login": "2017-06-22T18:34:15.835436Z",
 "logins_count": 2,
 "name": "sarah",
 "nickname": "sarah",
 "updated_at": "2017-06-22T18:34:15.835436Z",
 "user_id": "local|77bf124c-c4f2-4a89-a777-be9f8c66d099",
 "username": "sarah",
 "failed_logins_count": 0,
 "account_lockout_at": null,
 "failed_logins_initial_attempt_at": null,
 "last_failed_login_at": null,
 "password_changed_at": "2017-06-22T16:25:56.441806Z",
 "password_change_required": false
 }
]
}
```

### 2. To assign Sarah's `user_id` to an environment variable, enter:

```
$ export SARAH=$(ksctl users list --name sarah | jq -r ".resources[0]|jq -r ".user_id")
```

## Create a key

```
$ ksctl keys create --autoname --ownerid $SARAH --jsonfile keyparams.json
```

---

This command can be run in a loop to automatically create as many keys as desired. If your loop runs longer than the lifetime of your JWT, you can periodically refresh the token with a new create. The lifetime of your JWT is returned in the create call; see the duration field. The value is in seconds.

---

## Refresh the token

```
$ export KSCTL_JWT=$(ksctl tokens create --user sarah --password Sarah_pwl | jq -r ".jwt")
```

As you can see, there is some setup effort involved in these commands. However, the payoff in ease-of-use and performance is worth the time spent.

## Support CLI

This section describes how to use **ks\_support** (the Support CLI tool) to request an admin support token and then use that token to start a shell as the support user. No installation is necessary, it comes with the CipherTrust Manager appliance.

### Request

1. Connect to your CipherTrust Manager instance via: `ssh ksadmin@ip.address`

The following prompt is displayed:

```
ksadmin@keysecure:~$
```

2. Start the `ks_support` tool, as follows:

```
sudo ks_support request
```

The following output is displayed:

Copy the following request to KeySecure support:

```
U3BhY2U6IHRoZSBmaW5hbCBmcm9udGllci4gVGhlc2UgYXJlIHRoZSB2b3lhZ2VzIG9mIHRoZSB
zdGFyc2hpcCBFbnRlcnByaXNlLiBJdHMgZml2ZS15ZWZyIGlpc3Npb246IHRvIGV4cGxvcmUgc3
RyYW5nZSBuZlZxcgd29ybGRzLiBUbyBzZWVrIG9ldCBuZlZxcGBlZSBhbmQgbmV3IGNpdmlsaXphd
GlbnbnMuIFRvIGJvbGRseSBnbyB3aGVyZSBub3R5Y2V4IGV4cGxvcmUgc3RyYW5nZSBuZlZxcgd29ybGRzLiBUbyBzZWVrIG9ldCBuZlZxcGBlZSBhbmQgbmV3IGNpdmlsaXphd
```

3. Select and copy all of the string after the `support:` (starting at `U3BhY2U6...` and ending with `...JlLgo=` in this example) to the clipboard.
4. Paste the string into an email that you send to your CipherTrust Manager support contact.

### Shell

After you have received a response from the CipherTrust Manager support:

1. Start the `ks_support` tool again:

```
sudo ks_support shell
```

The following prompt is displayed:

```
Paste text here, then press <ENTER> :
```

2. Copy the string you received from your CipherTrust Manager support contact.
3. Paste the string immediately after the prompt, as follows:

```
Paste text here, then press <ENTER>: SXQgaXMgYSBwZXJpb2Qgb2YgY2l2aWwgd2FyL
iBSZWJlbCBzcGFjZXNoaXBzLCBzdHJpa2luZyBmcm9tIGVgaGlkZGVuIGJhc2UsIGhhdmUgd29
uIHRoZWlyIGZpcnN0IHZpY3RvcnkgYWdhW5zdCB0aGUgZlZxcGBlZSBhbmQgbmV3IGNpdmlsaXphd
ERlcm1uZyB0aGUgYmF0dGx1LCBSc2VzIGV4cGxvcmUgc3RyYW5nZSBuZlZxcgd29ybGRzLiBUbyBzZWVrIG9ldCBuZlZxcGBlZSBhbmQgbmV3IGNpdmlsaXphd
```

---

```
wbGFucyB0byB0aGUgRWlwaXJl4oCZcyB1bHRpbWF0ZSB3ZWFWb24sIHRoZSBERUFUSCBTVEFSL
CBhbiBhcm1vcmVkiHNwYWNlIHNOYXRpb24gd2l0aCB1bm91Z2ggcG93ZXIgdG8gZGVzdHJveSB
hbiB1bnRpcmUgcGxhbmV0LiBQdXJzdWVkiGJ5IHRoZSBFbXBpcmXigJlziHNpbmlzdGVyIGFnZ
w50cywgUHJpbmNlc3MgTGVPYSByYWNlcyBob21lIGFib2FyZCB0ZXIgc3RhcnoaXAsIGNlc3R
vZGlhbiBvZiB0aGUgc3RvbGVuIHBSYW5zIHRoYXQgY2FuIHhdmUgaGVyIHBlb3BsZSBhbmQgc
mVzdG9yZSBmcmVlZG9tIHRvIHRoZSBnYWxheHkK
```

#### 4. Press **Enter**.

---

### Success

If the string matches the necessary controls, the following support shell is displayed:

```
root@keysecure:~#
```

At this point, you can perform the steps as instructed by the CipherTrust Manager support personnel. When you log out from this shell, you are returned to the `ksadmin` shell prompt (as shown above). Note that the response that you receive can only be used once and is only valid for one day. Any attempted reuse or use after expiration results in an error message (as shown below).

---

### Failure

If the string does not match the required controls, the following message is returned:

```
Error: The challenge response was not accepted. Here are some possible reasons why:
* bad input - check the input for missing leading or trailing characters, additional
whitespace, etc.
* expired challenge - challenges expire 1 day after being requested
* replayed - a challenge response can only be used once
```

If this occurs, relay these issues to your support personnel so that they can be remedied.

---

### Multiple Requests

In the event that you need support shell access multiple times in a single day (for example, if the appliance needs to be rebooted), then multiple requests may be generated and sent to your CipherTrust Manager support contact.

### Cleanup

The Support CLI tool automatically cleans up the files that it no longer needs, as they are used. However, it is possible that a number of requests may have been created that were not needed. To clean all of these requests from your system, run the following command as the `ksadmin` user:

```
sudo ks_support cleanup
```

This command returns no output except in the case of error. Any errors should be reported to your CipherTrust Manager support contact.

---

## Keys

Keys are similar to SafeNet KeySecure Classic keys, with a few differences. Keys can be created, modified, imported, exported, and deleted. Keys can be symmetric or asymmetric, variable sizes, and can be constrained to particular usages.

| Algorithm   | Sizes (* = default)          |
|-------------|------------------------------|
| aes         | 128, 192, 256*               |
| tdes        | 128, 192*                    |
| rsa         | 512, 1024*, 2048, 3072, 4096 |
| hmac-sha1   | >= 128, 160*                 |
| hmac-sha256 | 128, 192, 256*               |
| hmac-sha384 | 192, 288, 384*               |
| hmac-sha512 | 256, 384, 512*               |

| Algorithm | Curve (Size is defined by the curve.)                                                                                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ec        | brainpoolP224r1, brainpoolP224t1, brainpoolP256r1, brainpoolP256t1, brainpoolP384r1, brainpoolP384t1, brainpoolP512r1, brainpoolP512t1, prime256v1, secp224k1, secp224r1, secp256k1, secp384r1, secp521r1 |

## To create a key

Key can be created using the GUI, KSCTL, or API.

**NOTE** The CTE keys must be:

- > AES128, AES-256, ARIA-128, or ARIA-256
- > Shared with the CTE Clients group
- > Set as exportable

Only the clients that support the CBC CS1 encryption mode can have GuardPoints that use a policy with a CBC CS1 key.

To create a key using KSCTL, execute the command:

```
$ ksctl keys create -alg AES --name "testAESkey"
```

The response looks like:

```
{
 "id": "f0c2b40017dca0cdbbda90e2d224d0dcb4488ec412e156a594af3bc11a879b1a",
 "uri": "kylo:kylo:vault:keys:testaeskey-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2018-04-23T17:57:15.899060389Z",
 "name": "testAESkey",
 "updatedAt": "2018-04-23T17:57:15.899060389Z",
 "usage": "blob",
}
```



```

 "usageMask": 12,
 "meta": {
 "ownerId": "local|ee328ec6-f37b-4e9c-85be-8867120b6b79"
 },
 "algorithm": "AES",
 "size": 256,
 "format": "raw",
 "unexportable": false,
 "undeletable": false,
 "defaultIV": "3ddc71eea888f8ec554b63759f371af9",
 "objectType": "Symmetric Key",
 "activationDate": "2018-04-23T17:57:15.886182496Z",
 "state": "Active"
}

```

## Setting Key Permissions

Keys can be granted permissions to perform operations such as read, encrypt, and decrypt data. You can set key permissions on the RAW tab of the Key Metadata section. This can be done when creating new keys or when modifying existing keys.

Key permissions can be set using the GUI, KSCTL, or API.

### To set key permissions using the GUI

#### When creating a key

To set key permissions when creating a key:

1. Log on to the CipherTrust Manager console as administrator.
2. Open the **Keys & Access Management** application.
3. Click **Create a New Key**.
4. Specify required details.
5. Under the **Key Metadata** section, click the **Raw** tab.
6. In the **Key Metadata** text box, specify key permissions in the "permissions": {} parameter. Sample settings are given below:

```

"ownerId": "local|81f34b79-e3c4-4391-a4c9-4bdd85b13f98",
 "permissions": {
 "UseKey": [
 "stars"
],
 "ReadKey": [
 "stars"
],
 "DecryptWithKey": [
 "stars"
],
 "EncryptWithKey": [
 "stars"
],
 "customAttributes": [
 {
 "name": "color",
 "value": "blue"
 }
]
 }

```

```
}
],
}
```

Refer to the "Keys" section in the API playground for details on permissions.

## 7. Click **Create**.

### When modifying a key

To set key permissions when modifying a key:

1. Log on to the CipherTrust Manager console as administrator.
2. Open the **Keys & Access Management** application.
3. Under **Key name and aliases**, click the key name link. The key details are displayed.
4. Click the action button. A shortcut menu appears.
5. Click **Edit**. The **Edit <Key Name>** page is displayed.
6. Under the **Key Metadata** section, click the **Raw** tab.
7. In the **Key Metadata** text box, specify key permissions in the "permissions": {} parameter. Sample settings are given below:

```
"ownerId": "local|81f34b79-e3c4-4391-a4c9-4bdd85b13f98",
 "permissions": {
 "UseKey": [
 "stars"
],
 "ReadKey": [
 "stars"
],
 "DecryptWithKey": [
 "stars"
],
 "EncryptWithKey": [
 "stars"
],
 "customAttributes": [
 {
 "name": "color",
 "value": "blue"
 }
],
 }
}
```

Refer to the "Keys" section in the API playground for details on permissions.

## 8. Click **Update**.

## Versioned Keys

### KeySecure Classic

A KeySecure Classic versioned key is a key that maintains a single set of key metadata but contains multiple sets of key data. Each set of key data belongs to a unique version of the key.

---

A KeySecure Classic crypto operation using a versioned key prepends a header containing key version information to the resulting ciphertext. A crypto operation using an unversioned key does not prepend the ciphertext with this header.

## CipherTrust Manager

The CipherTrust Manager handling of versioned keys depends on the interface used and the setting of key flag (versionedKey). For backward compatibility with KeySecure Classic, when the NAE-XLM interface is used and if the key flag (versionedKey = true), the key is handled as a versioned key and the crypto operation prepends a header to the resulting ciphertext. When the REST interface is used, crypto operations always produce the kind of ciphertext that KeySecure Classic produces when using unversioned keys, that is, produce unadorned ciphertext with no header.

Similarly, if you perform the inverse crypto operations on the ciphertext encrypted using a versioned key (e.g. decrypt, verify...), the operation must be done via the NAE-XML interface, since the REST interface does not handle these header bytes.

To maintain backward compatibility with KeySecure Classic, when you create a key through the NAE-XML interface with the **<Versioned/>** element present, CipherTrust Manager server adds a flag to the key (versionedKey = true), indicating that when the key is used via the NAE-XML interface, the key should behave like a legacy NAE-XML versioned key (i.e. the produced ciphertext results that contain the legacy header bytes). This internal flag is only honored when the key is used via the NAE-XML interface; the flag has no effect on REST interface interactions.

**NOTE** The (versionedKey = true) flag can be created through the REST interface. Detailed documentation is available in the "API playground".  
To create the (versionedKey = true) flag via the GUI, select **Set as "Versioned Key" for backward compatibility**.

**NOTE** When using NAE, 'VersionedKey' information stored in KeySecure Classic and CipherTrust Manager cannot be exported by or imported from either product. If 'VersionedKey' must be set, it must be noted by the operator that it exists and turned on in the GUI after the import has occurred. An alternative to export/import is to run a migration from KeySecure Classic to the CipherTrust Manager as this operation preserves 'VersionedKey' information.

## Key States (KMIP and REST)

A key can be in any of the following states at any given time:

- > **Pre-Active:** The key exists, but should not be used for any cryptographic purpose.
- > **Active:** The key can be used for any cryptographic purpose.
- > **Deactivated:** The key should **not** be used for cryptographic operations such as encryption, signing, wrapping, MACing, and deriving. The key should only be used for decryption, signature verification, unwrapping, and MAC verification.
- > **Compromised:** The key should **not** be used for cryptographic operations such as encryption, signing, wrapping, MACing, and deriving. The key should only be used for decryption, signature verification, unwrapping, and MAC verification.

- 
- > **Destroyed:** The key should not be used for any cryptographic purpose.
  - > **Destroyed Compromised:** The key should not be used for any cryptographic purpose. Its compromised status should be retained for audit or security purposes.

## Effects of Negative Actions on Keys

When you delete, destroy, revoke, or archive a key, the following happens:

- > **Delete:** The key is deleted from the database altogether, including the key metadata and key material.
- > **Destroy:** The key material is deleted from the database, while the key metadata is kept.
- > **Revoke:** The key state is set to "Revoked". The key material is available on the server, which:
  - Can be used only for decryption, signature verification, unwrapping, and MAC verification.
  - **Cannot** be used for encryption, signing, wrapping, MACing, and deriving etc.
- > **Archive:** The key state is set to "Archived". The key cannot be used for any cryptographic operations. Later the key can be recovered and used or modified, as required.

## Key States (NAE)

The NAE Interface supports the following Key States.

**NOTE** The CipherTrust Manager supports these Key States only when using the NAE interface.

### Pre-operational (Pre-active)

In this state the key is created but has not yet been used for any crypto operation.

### Operational (Active)

In this state the key can be used for both protection and process, i.e. Encrypt and Decrypt, sign and signV.

### Post-operational (Restricted/Retired)

In this state the key can be used for process only and not for protection, for example:

- > When in Restricted state,
  - the key can be used to decrypt but not to encrypt,
  - the key can be used to signV but not to sign.
- > When in Retired state,
  - the key cannot be used for any crypto operation (i.e. no encrypt, decrypt, sign or signv).

It is possible to move the key back to Operational (Active) state if required. The reason for transitioning between the Operational (Active) state and the Post-operational (Restricted) should be recorded.

### Destroyed (Wiped)

In this state the key bytes are deleted but meta data is preserved for audit purposes. Changing a key's state to Destroyed (Wiped) requires manual intervention and cannot be done via key-rotation/key-retention.

---

## Key Rotation

The key rotation operation generates a new version of a key with same key name and attributes, but with new key material. The operation provides an easy way to rotate keys in bulk periodically at the specified time.

**NOTE** Scheduler itself is a generic service, so it provides granularity of a minute. A job configuration should take the nature of operation information into consideration. For example, rotating the same set of multiple keys every minute is a highly unlikely use case.

### Scheduling Key Rotation

The CipherTrust Manager can periodically run key rotation jobs in the background. This can be achieved by creating scheduler configurations for key rotation.

This section provides examples of key rotation scheduler configurations using the ksctl utility. Refer to the ksctl or API documentation for details.

To create a scheduler configuration for key rotation, run the command:

```
ksctl scheduler configs create key-rotation --name <config-name> --run-at "<cron-expression-format>"
```

Here,

- > **<config-name>**: Name for the job configuration.
- > **<cron-expression-format>**: Time when the job runs. The format must have *five* fields. If the number of fields is not equal to five, the format becomes invalid. Refer to ["Time Specification" on page 89](#) for details.

Optionally, you can specify the following:

- > **--key-query-json**: Query used for determining the keys to be rotated. By default, all keys are rotated. The query should be specified as a JSON blob. Refer to the API documentation for the JSON schema associated with the key query.
- > **--metadata-json**: Metadata for the key that is replaced by the key rotation operation. The metadata should be specified as a JSON blob. The key metadata is also a JSON blob. The supplied metadata is added to the key metadata (existing data is overwritten). For example, this provides a mechanism to change the NAE state of a key by supplying `{"naeState": "Restricted"}`.
- > **--deactivate-replaced-key**: Time interval in seconds ( $\geq 0$ ) after which the key that is replaced by the key rotation procedure is deactivated. By default, the key is not deactivated.

**NOTE** Refer to the ksctl or API documentation for the complete list of parameters needed for scheduling key rotation operations.

### Examples

To schedule a key rotation everyday at 9 AM, and modify the metadata of the key that is replaced, run the command:

```
ksctl scheduler configs create key-rotation --name change-meta --run-at "0 9 * * *"
--metadata-json '{"naeState": "Restricted"}'
```

To rotate all AES keys every Monday at 9 AM:

```
ksctl scheduler configs create key-rotation --name rotate-aes --run-at "0 9 * * 1"
--key-query-json '{"algorithm": "AES"}'
```

To activate the above key rotation on May 8, 2021 at 5:57:51 PM:

```
ksctl scheduler configs create key-rotation --name rotate-aes --run-at "0 9 * * 1"
--start-time "May 8, 2021 5:57:51 PM"
```

To schedule a key rotation everyday at 9 PM, and rotate keys that were created at least a year ago:

```
ksctl scheduler configs create key-rotation --name rotate-old-keys --run-at "0 21 *
* *" --metadata-json '{"createdBefore": "-1y"}'
```

## Crypto Operations

The Thales CipherTrust Manager supports the following crypto operations:

### Symmetric Encryption

Symmetric encryption uses AES, TDES, or AES-GCM (with or without authentication data).

A variety of modes and key sizes are supported. Operations can be chained to encrypt a stream of data, split over multiple requests.

| Algorithm | Mode     | Padding (* = default) |
|-----------|----------|-----------------------|
| AES       | GCM      |                       |
| AES       | ECB, CBC | none, PKCS7*          |
| TDES      | CBC      | none, PKCS5*          |

### Asymmetric Encryption

Asymmetric encryption uses RSA keys. A variety of paddings are supported.

| Algorithm | Mode | Padding (* = default) |
|-----------|------|-----------------------|
| RSA       |      | OAEP, PKCS1*          |

### Asymmetric Sign and Verify

| Algorithm |
|-----------|
| SHA1      |
| SHA256    |
| SHA384    |

### Algorithm

SHA512

## MAC and MAC Verify

### Algorithm

SHA1

SHA256

SHA384

SHA512

## Format Preserving Encryption

Format-Preserving Encryption is a special type of encryption that keeps the output of the algorithm in the same format and length as the input. This can be important when the result is stored in a database that expects a certain format. A common use case is for Credit Card Numbers, zip codes, telephone numbers, names, addresses etc.

**NOTE** Using this service/api/product does not confer compliance with any regulation regarding the use, storage and processing of Credit Card data.

Normally FPE algorithms limit the size of the data that can be encrypted to a number based on the size of the alphabet. Selecting the "alphabet" determines two things - how long your input can be - and what characters will be encrypted and which ones will be left in place.

| sample       | alphabet     | sample result  |
|--------------|--------------|----------------|
| 443-555-1055 | digit        | 231-601-4293   |
| 443-555-1055 | alphabet     | 443-555-1055   |
| 443-555-1055 | alphanumeric | ab4-56k-DG3e   |
| 443-555-1055 | printable    | &08yfh210f8\$5 |
| Hello World! | digit        | Hello World!   |
| Hello World! | alphabet     | djheF dkoRE!   |
| Hello World! | alphanumeric | 4JhR6 0e5w2!   |

| sample       | alphabet  | sample result |
|--------------|-----------|---------------|
| Hello World! | printable | 3J;e6#0e^1&d  |

| Alphabet     | Maximum size |
|--------------|--------------|
| digit        | 56           |
| alphabet     | 32           |
| alphanumeric | 32           |
| printable    | 28           |

The algorithm only handles ASCII chars 0x20 through 0x7E. There is currently no support for other character sets.

The FPE implementation also supports inputs that are longer than the limits imposed above. However using the API this way is more complicated and additional parameters must be passed.

## Certificate Authority

A Certificate Authority (CA) acts as the initially trusted shared entity between peers and issues signed certificates to make it possible for each party to trust the other. The CA issues and installs digital certificates and issues certificate signing requests (CSR).

A certificate generally acts as the identity of a server or client and this API can be used to issue server and client certificates for setting up trusted communication channels to the Thales CipherTrust Manager system.

CipherTrust Manager distinguishes between local CAs and external CAs. A local CA can issue signed certificates since the private signing key is stored inside the CipherTrust Manager system. An external CA does not store the private key. Instead an external CA is used as a trusted entity for various interfaces and services inside the system. In this case certificates are issued externally. It is fine to have a mix of both.

The first time a CipherTrust Manager is started, a new local `KeySecure root` CA is automatically generated. This CA is used to issue initial server certificates for the interfaces available in the system.

An easy way to inspect the certificate chain is to view the certificates in your browser when you connect to the web interface. All interfaces and services will by default trust this CA, meaning that for interfaces that support client authentication, a client certificate, issued from this initial `KeySecure root` CA, will automatically be trusted by the system. If preferred, it is possible to create new local CAs and/or external CAs and use these instead for the internal interfaces and services.

Many CA operations can be performed via the CipherTrust Manager GUI.

Here are a few basic procedures for creating CAs and issuing certificates using the CLI Tool (ksctl).

### Creating a local CA

When creating a new local CA, it remains pending until signed:



---

```
$ ksctl ca locals create --cn "Test CA" --csr-outfile csrfile
```

This returns a CSR that can then be signed by an external CA if desired.

To just self-sign the CA with a one year duration, use the id returned in the call above:

```
$ ksctl ca locals self-sign --id 3593c53b-fbeb-4edb-b84d-c85526ae2f83 -x 365
```

## Issuing a signed certificate

CipherTrust Manager can only issue certificates for “local” CAs. To create a certificate, a CSR must first be generated.

For the highest level of security, it is usually best for the CSR to be generated externally to CipherTrust Manager so the private key is never exposed.

For convenience, CipherTrust Manager can create a CSR and private key for you:

```
$ ksctl ca csr -cn "My Client" --csr-outfile csrfile --key-outfile keyfile
```

Once you have a CSR, you can have the CipherTrust Manager issue a certificate:

```
$ ksctl ca locals certs issue --ca-id 3593c53b-fbeb-4edb-b84d-c85526ae2f83 --csr-infile csrfile -x 700 -o client
```

**NOTE** It is also possible to create the CSR and the private key using any other software, as this API is stateless and doesn't store anything within CipherTrust Manager.

## Certificate Expiration Check

CipherTrust Manager inspects the expiration date of each certificate everyday, at a preset system time. CipherTrust Manager then creates list of certificates on the following basis:

- > Certificates whose expiration dates are within 91 days  
*This list is logged in the **Records** section once every week*
- > Certificates whose expiration dates are within 7 days  
*This list is logged in the **Records** section once every day*
- > Certificates which have already expired  
*This list is logged in the **Records** section once every day*

You can also create alarm triggers for these records. For more details, go to [Creating Alarm Trigger for Certificate Expiration](#).

## Creating an external CA

To create an external CA you just need to upload the PEM-formatted certificate file:

```
$ ksctl ca externals upload -cert-infile mycert.pem
```

# Verifying Revocation Status of Client Certificates

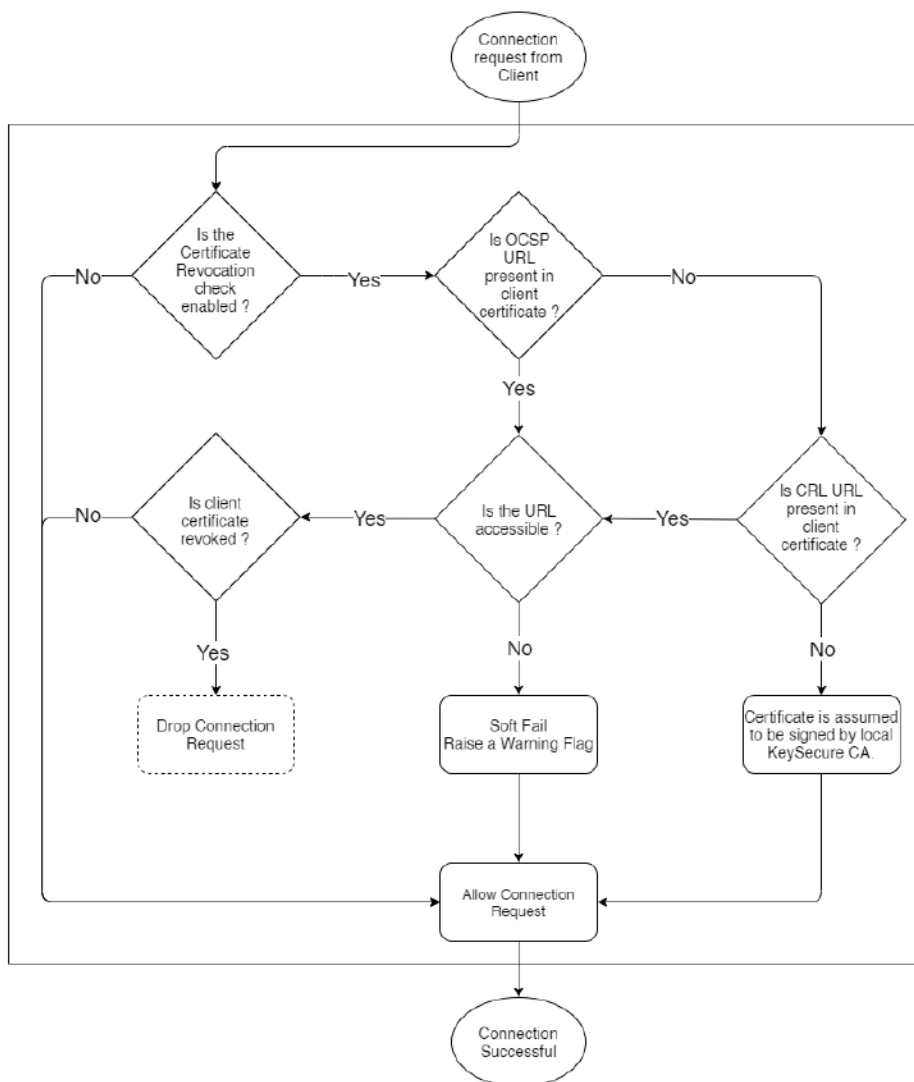
CipherTrust Manager can be configured to verify the revocation status of client certificate presented to NAE or KMIP interface before establishing a connection with the client.

To configure the CipherTrust Manager for inspecting client certificate revocation status:

- > Select `TLS verify client cert` [Interface Mode](#).
- > [Enable](#) the Certificate Revocation Check.

A certificate contains an OCSP responder URL and a Certificate Revocation List (CRL) URL, which are used for verifying the revocation status of the certificate.

Let's understand how CipherTrust Manager verifies the revocation status of a certificate and permits/drops connection requests in such cases:



CipherTrust Manager looks the client certificate for OCSP responder URL and CRL URL.

## 1. If OCSP URL is present

CipherTrust Manager accesses this URL to verify the revocation status of the Client Certificate.

- 
- a. If OCSP URL is accessible, the status of certificate gets verified successfully, and CipherTrust Manager allows/drops connection request accordingly.
  - b. If OCSP URL is not accessible due to any reason, CipherTrust Manager considers the situation as a "Soft Fail". It allows the connection to establish, but reports a warning. This warning audit log and its details can be viewed in the Records.

## 2. If OCSP URL is not present, but CRL URL is present

CipherTrust Manager verifies the status of certificate using the CRL, and allows/drops connection request accordingly.

- a. If CRL URL is accessible, the status of certificate gets verified successfully, and CipherTrust Manager allows/drops connection request accordingly.
- b. If CRL URL is not accessible due to any reason, CipherTrust Manager considers the situation as a "Soft Fail". It allows the connection to establish, but reports a warning. This warning audit log and its details can be viewed in the Records.

## 3. If both OCSP URL and CRL URL are not present

CipherTrust Manager considers the client certificate to be signed by its local CA, and allows the connection to establish.

**NOTE** In any case, if the certificate is found to be revoked, CipherTrust Manager drops the connection request and logs it in Records.

## OCSP and CRL Caching

As stated above, verifying the revocation status of a certificate involves establishing a connection with the URL (OCSP or CRL) present in the certificate, and verifying its revocation status. Once the revocation status of a certificate is verified, CipherTrust Manager stores this information for some preset time.

Let's understand how long this information is stored in and how it is used in this time frame:

### > For OCSP method

After successfully connecting to the OCSP URL, CipherTrust Manager stores the revocation status of the client certificate for a duration of 5 minutes.

- If CipherTrust Manager receives another connection request from the same client within 5 minutes of a previously successful connection, then CipherTrust Manager refers to the cached revocation status value to verify its revocation status.
- If CipherTrust Manager receives another connection request from the same client after 5 minutes of a previously successful connection, then CipherTrust Manager verifies its revocation status through the OCSP URL again.

### > For CRL method

Each CA promises to update its CRL at the day and time specified in the *Next Update* field for that CA. While performing a certificate revocation check, the CipherTrust Manager inspects the *Next Update* value for the CRL associated with each CA on the CipherTrust Manager.

- If the *Next Update* value for that CRL is in the past, the CipherTrust Manager attempts to connect to the CRL distribution point (CDP) for the CA to download the updated CRL.

- If the *Next Update* value for that CRL is in the future, the CipherTrust Manager waits until that specified time to attempt to connect to the CDP and download the updated CRL.

## Enabling/Disabling Certificate Revocation Check

The Certificate Revocation Check is enabled by default. You can enable or disable the Certification Revocation Check using:

### > API playground

Refer to "Properties" section (*/v1/configs/properties*).

### > CLI tool (ksctl)

Refer the following examples:

- Command to enable Certificate Revocation Check

```
$ ksctl properties modify -n ENABLE_CERT_REV_CHECK -p true
```
- Command to disable Certificate Revocation Check

```
$ ksctl properties modify -n ENABLE_CERT_REV_CHECK -p false
```
- Available Flags:

| Flags       | Input Type            | Description                                  |
|-------------|-----------------------|----------------------------------------------|
| -h, --help  | <i>not applicable</i> | Command help                                 |
| -n, --name  | string                | Name of the system configuration             |
| -p, --value | string                | Value to be set for the system configuration |

## Alarms

Alarms are Thales CipherTrust Manager's mechanism for notifying administrators when the state of the CipherTrust Manager or one of its nodes is not healthy or is not configured securely and should be investigated. Alarms can be fetched via the REST API, CLI, or the GUI.

Alarm states are listed in the following table:

| State   | Description                                                                                                                                                                                                   |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| off     | The alarm is inactive and does not need to be investigated.                                                                                                                                                   |
| on      | The alarm is active and should be investigated. It remains active until the condition that triggered it is not longer valid.                                                                                  |
| unknown | The alarm's state could not be determined and should be investigated. Typically this occurs when the service responsible for triggering the alarm failed to communicate its state (e.g. the service is down). |

The CipherTrust Manager has the following built-in alarms:

| Name                                | Severity | Trigger                                                                                                                                 | Remediation                                                                                                               |
|-------------------------------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Disk Full                           | Critical | The root file system's 'used space' percentage exceeds the configured threshold of 80%.                                                 | Increase the node's disk/partition/file-system, or replace the node with a new node that has sufficient storage.          |
| NAE TLS Disabled                    | Critical | The system is started with NAE's interface mode configured to not use TLS <link to interfaces>.                                         | Modify NAE's interface mode <link to interfaces> to one of the options that specifies TLS and restart <link to services>. |
| HSM Offline                         | Critical | The system cannot access the HSM.                                                                                                       | Restore connectivity to the HSM.                                                                                          |
| Cluster Node Down                   | Critical | A node within a cluster is down.                                                                                                        | Restore connection of the down node to the cluster.                                                                       |
| Cluster Node Certificate Expiration | Critical | An internal CipherTrust Manager certificate that is used for database access and clustering has expired, or will expire within 30 days. | Reboot the box when the alarm triggers. A new certificate is automatically generated after the reboot.                    |
| Syslog Connection Offline           | Critical | A Syslog connection goes offline.                                                                                                       | Restore connectivity to the specified Syslog connection.                                                                  |

The CipherTrust Manager server also has dynamic alarms that are triggered based on audit record conditions. Refer to: ["Configuring alarm triggers based on an audit record" on page 53](#)

## To list all alarms

```
$ ksctl alarms list
```

Example response:

```
{
 "skip": 0,
 "limit": 10,
 "total": 4,
 "resources": [
 {
 "name": "Disk Full",
 "state": "off",
 "state_change_at": "2018-07-03T18:30:31.430320038Z",
 "description": "Disk full alarm triggers above 80% of capacity (currently
at 48%)",
 "severity": "critical"
 },
 {
 "name": "NAE TLS Disabled",
 "state": "off",
 "state_change_at": "2018-07-03T18:30:57.28077419Z",
 "description": "TLS is disabled on the NAE interface",

```

---

```
 "severity": "critical"
 },
 {
 "name": "HSM Offline",
 "state": "off",
 "state_change_at": "2018-07-05T20:02:58.326696594Z",
 "description": "HSM is offline",
 "severity": "critical"
 }
 {
 "name": "Cluster Node Down",
 "state": "on",
 "state_change_at": "2018-07-05T20:02:58.326696594Z",
 "description": "Current down node(s): 127.0.0.1:5555",
 "severity": "critical"
 }
}
]
```

---

### To acknowledge that an alarm is under investigation:

```
$ ksctl alarms acknowledge -i 1650789b-e8e6-4915-91ab-b067e073f39f
```

---

### To clear (i.e. turn off) an alarm:

```
$ ksctl alarms clear -i 1650789b-e8e6-4915-91ab-b067e073f39f
```

End of Document



# CipherTrust Manager

**Version: v2.22 (latest)**

29 January 2026

---

# Contents

|                                                                      |     |
|----------------------------------------------------------------------|-----|
| CipherTrust Manager Deployment . . . . .                             | 3   |
| Hardening Guidelines . . . . .                                       | 5   |
| Install Virtual CipherTrust Manager . . . . .                        | 10  |
| Plan Configuration Settings for Cloud Init . . . . .                 | 11  |
| Deploying with Cloud-init. . . . .                                   | 20  |
| Amazon Web Services Deployment . . . . .                             | 29  |
| Google Cloud Deployment . . . . .                                    | 42  |
| Alibaba Cloud Deployment . . . . .                                   | 49  |
| Microsoft Azure Deployment . . . . .                                 | 55  |
| Microsoft Azure Stack Deployments . . . . .                          | 72  |
| Microsoft Azure Dedicated HSM Provisioning . . . . .                 | 79  |
| Oracle Cloud Deployment . . . . .                                    | 106 |
| Private Cloud Deployment . . . . .                                   | 113 |
| Resizing the Virtual Hard Disk After Deployment . . . . .            | 146 |
| Install Physical CipherTrust Manager Appliance . . . . .             | 147 |
| Securely Transporting a CipherTrust Manager k570 Appliance . . . . . | 184 |
| Remote PED-Authenticated k570 Initialization . . . . .               | 193 |
| Backing Up k570 Root of Trust Keys . . . . .                         | 200 |
| Hardware Specifications . . . . .                                    | 212 |
| HSM Tamper Behavior . . . . .                                        | 221 |
| Reboot the Appliance . . . . .                                       | 223 |
| Power Supply and Fan Maintenance . . . . .                           | 225 |
| Serial Connections . . . . .                                         | 231 |
| Front Locking Bezel . . . . .                                        | 235 |
| Network Configuration Tutorial . . . . .                             | 236 |
| Planning Network Configuration . . . . .                             | 240 |
| Install the CLI Toolkit . . . . .                                    | 253 |
| Changing the Initial Password . . . . .                              | 261 |
| Network Time Protocol Server Configuration . . . . .                 | 263 |
| Starting Services After Deployment . . . . .                         | 264 |
| System Configuration Utility . . . . .                               | 267 |
| Resetting a CipherTrust Manager. . . . .                             | 285 |



---

# CipherTrust Manager Deployment

The CipherTrust Manager server can be deployed in a number of cloud based environments as well as in a physical appliance.

## Product references

CipherTrust Manager - refers to all platforms running the CipherTrust Manager software.

Virtual CipherTrust Manager k170v and Virtual CipherTrust Manager k470v (or simply k170v and k470v) - refer specifically to virtual and private cloud instances running the CipherTrust Manager Software.

- k170v is intended for simplified and centralized key management. For example, k170v is suited to lab environments, low transaction encryption use cases, and storage encryption key management with KMIP (Key Management Interoperability Protocol). k170v allows for the usage of four CPUs or fewer. k170v is applied as a [license](#).
- k470v is intended to support high transaction-per-second encryption operations as is typically required from the CipherTrust Data Security Portfolio suite. k470v allows for the usage of more than four CPUs. k470v is applied as a [license](#).

CipherTrust Manager k470, k570, and k160 Appliances (or simply k470, k570, and k160) - refers specifically to the k470, k570, and k160 physical appliances running the CipherTrust Manager Software.

SafeNet KeySecure Classic (or simply KeySecure Classic) - refers to the previous generation of SafeNet KeySecure Appliances (models k450 and k460). You can [migrate data](#) from these appliances to CipherTrust Manager k470 and k570 physical appliances.

## Deployment environments

How the CipherTrust Manager is distributed and deployed is based on the environment. The supported environments are listed below along with the supported distribution method:

### Public Cloud Images

- Amazon Web Services: SafeNet Cloud Provisioning System

Refer to: [AWS Deployment](#)

- Google Cloud:

Refer to: [Google Cloud Deployment](#)

- Microsoft Azure: Available as a BYOL image in the Microsoft Azure Marketplace

Refer to: [Azure Deployment](#)

- Oracle Cloud

---

Refer to: [Oracle Cloud Deployment](#)

- Alibaba Cloud

Refer to: [Alibaba Cloud Deployment](#)

## Private Cloud Images

- VMware vSphere: OVA image file

Refer to: [Private Cloud Deployment](#)

- Microsoft Hyper-V: VHDX image file

Refer to: [Private Cloud Deployment](#)

- Nutanix AHV: VMDK image file

Refer to: [Private Cloud Deployment](#)

- OpenStack: qcow2 image file

This can be installed on to KVM or QEMU hypervisor.

An example command with KVM hypervisor is:

```
virt-install --virt-type kvm --name <virtual_image_name> --ram 2048 --disk path=
h=<path_to_qcow2_image>,size=16,format=qcow2 --network network=default --graph
ics vnc,listen=0.0.0.0 --noautoconsole --os-type=linux --os-variant= ubuntu16.0
4 --import
```

- RedHat OpenShift: qcow2 image file

Refer to: [Private Cloud Deployment](#)

For network configuration on private cloud images, refer to [Network Interface Configuration Tutorial](#)

## Hybrid Cloud Images

- Azure Local: VHDX image file

Refer to: [Azure Stack Deployments](#)

- Azure Stack Hub: VHDX image file

Refer to: [Azure Stack Deployments](#)

## Physical Appliances

### CipherTrust models:

- CipherTrust k470
  - Refer to: [Physical Deployment](#)
  - Refer to: [Migrating from KeySecure Classic](#)
- CipherTrust k570
  - Refer to: [Physical Deployment](#)
  - Refer to: [Migrating from KeySecure Classic](#)
- TCT CipherTrust Manager k160

Refer to the [CipherTrust Manager k160 Quick Start Guide](#)

For network configuration on physical appliances, refer to [Network Configuration Tutorial](#)

## Hardening Guidelines

The CipherTrust Manager should be deployed into as secure an environment as possible. Every effort has been made to make the CipherTrust Manager as secure as possible, however, additional precautions should be taken especially when the CipherTrust Manager is deployed into an untrusted environment.

## Network Security Groups

A network security group includes security rules that permit or deny inbound network traffic to required ports of CipherTrust Manager interfaces and outbound network traffic from CipherTrust Manager clients. For the list of available CipherTrust Manager network traffic ports, refer to the following Recommended Interface Types and Port Assignments.

### Recommended Interface Types and Port Assignments

The recommended interface types and port assignments for the CipherTrust Manager are:

| Type                                  | Protocol | Port Number                                                                                                                                                                                                                                                                                |
|---------------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AWS CloudHSM root of trust connection | TCP      | If you have a Virtual CipherTrust Manager deployed on an AWS EC2 instance, attach the cluster security group to the EC2 instance, as described in <a href="#">AWS documentation</a> . For all other deployments, configure the network security group to allow 2223-2225 Inbound/Outbound. |
| Clustering                            | TCP      | 5432 Inbound/Outbound,                                                                                                                                                                                                                                                                     |

| Type                                                                                                                                                                                                                                                                                                        | Protocol | Port Number                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-----------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                                                                                                                                             |          | 2380 Inbound/Outbound                                                                                           |
| Data Discovery and Classification (DDC) Agent                                                                                                                                                                                                                                                               | TCP      | 11117 Inbound                                                                                                   |
| DNS Queries                                                                                                                                                                                                                                                                                                 | UDP      | 53 Outbound                                                                                                     |
| HTTP                                                                                                                                                                                                                                                                                                        | TCP      | 80 Inbound                                                                                                      |
| HTTPS, which includes Web UI, ksctl CLI, REST API, CipherTrust Transparent Encryption (CTE), CipherTrust Transparent Encryption UserSpace (CTE-U) client access and Data Protection Gateway (DPG), CipherTrust RESTful Data Protection (CRDP), CT-V WebService, CT-VL, CADP for JAVA Webservice connectors. | TCP      | 443 Inbound for most deployment types.<br>9443 Inbound for Virtual CipherTrust Manager launched from AWS China. |
| HTTPS for DPoD Luna Cloud HSM Service root of trust connection                                                                                                                                                                                                                                              | TCP      | 443 Outbound                                                                                                    |
| Key Management Interoperability Protocol (KMIP)                                                                                                                                                                                                                                                             | TCP      | 5696 Inbound                                                                                                    |
| LDAP                                                                                                                                                                                                                                                                                                        | TCP      | 389 Outbound                                                                                                    |
| LDAPs                                                                                                                                                                                                                                                                                                       | TCP      | 636 Outbound                                                                                                    |
| Luna Network HSM (NTLS type) or TCT Luna T-Series Network HSM connections, any usage                                                                                                                                                                                                                        | TCP      | 1792 Outbound                                                                                                   |
| Luna Network HSM (STC type), root of trust connection only                                                                                                                                                                                                                                                  | TCP      | 5656 Outbound                                                                                                   |
| Network-Attached Encryption (NAE) interface                                                                                                                                                                                                                                                                 | TCP      | 9000 Inbound                                                                                                    |
| NTP                                                                                                                                                                                                                                                                                                         | UDP      | 123 Outbound                                                                                                    |
| SCP server for automatic backup storage                                                                                                                                                                                                                                                                     | TCP      | 22 Outbound                                                                                                     |
| SMTP                                                                                                                                                                                                                                                                                                        | TCP      | 587 Outbound                                                                                                    |
| SNMP                                                                                                                                                                                                                                                                                                        | UDP      | 161 Inbound,<br>162 Outbound                                                                                    |
| SSH                                                                                                                                                                                                                                                                                                         | TCP      | 22 Inbound                                                                                                      |
| SSH (if using Luna Network HSM, TCT Luna T-Series HSM, or AWS CloudHSM as root of trust)                                                                                                                                                                                                                    | TCP      | 22 Outbound                                                                                                     |
| Syslog                                                                                                                                                                                                                                                                                                      | UDP      | 514 Outbound                                                                                                    |
| Syslog                                                                                                                                                                                                                                                                                                      | TCP      | 6514 Outbound                                                                                                   |
| Thales Data Platform (TDP) Knox connection for Data Discovery and                                                                                                                                                                                                                                           | TCP      | 8443 Outbound                                                                                                   |

| Type                 | Protocol | Port Number |
|----------------------|----------|-------------|
| Classification (DDC) |          |             |

If you are using the [Secrets Management](#) feature, the included Akeyless Gateway service requires [public network connectivity to Akeyless SaaS Core Services](#).

### Caution

- Rules with source IP of 0.0.0.0/0 (IPv4) and ::/0 (IPv6) allow all IP addresses to access the instance. It is recommended to set security group rules to allow access from known IP addresses only.
- Whenever a new interface is added, the respective port should be added to the security group also.
- Port 9100 is reserved for an internal service and is not available for any CipherTrust Manager interface.

## Port Assignments for Related Components

Some CipherTrust Manager use cases require other connections between related components, which also require open network traffic to certain ports. These connections are not directly to or from the CipherTrust Manager appliance.

- DDC agents form connections to data stores. Refer to [DDC documentation](#) for a list of ports used for communication between DDC agents and supported data store types.
- As a prerequisite to setting up a Luna Network HSM as [root-of-trust](#) for CipherTrust Manager, you must create a partition on the Network HSM and register the CipherTrust Manager as a client. As indicated in [Luna documentation on port usage](#), SSH access for this prerequisite configuration requires port 22.

## Encryption of Virtual CipherTrust Manager

It is best practice to encrypt any Virtual CipherTrust Manager used in production. This is especially true if the Virtual CipherTrust Manager is deployed into an untrusted environment. When a Virtual CipherTrust Manager instance first boots, there are a number of secrets generated specific to that instance. To ensure that these secrets are never exposed, the CipherTrust Manager should be encrypted on first boot before it generates these. As well, [disabling the cloud-init service](#) on an unencrypted instance makes the disk resize operation unavailable.

Please refer to [Disk Encryption](#) for details.

## System Administrative Key

The SSH Private Key, used to access the System Administrative account "ksadmin", is extremely sensitive and should be kept in a secure environment.

## Root of Trust Hardware Security Module (HSM) Configurations

If configured to use an HSM as root of trust, the CipherTrust Manager will protect all of its secrets with a non

exportable HSM key. To protect all secrets, the CipherTrust Manager must be connected to the HSM on first boot. This is the most secure configuration. Special configuration is required to use an HSM with a [cluster of appliances](#).

## TLS Compatibility

This table identifies the supported TLS versions for each of the CipherTrust Manager interfaces. The default minimum value reflects the default `minimum_tls_version` setting. This setting controls the lowest acceptable TLS version allowed for connections to the interface.

| Interface | Minimum TLS version | Maximum TLS version | Default Minimum TLS version |
|-----------|---------------------|---------------------|-----------------------------|
| Web UI    | TLS 1.2             | TLS 1.3             | TLS 1.2                     |
| NAE       | TLS 1.0             | TLS 1.3             | TLS 1.2                     |
| KMIP      | TLS 1.0             | TLS 1.3             | TLS 1.2                     |

### Caution

TLS 1.0 and TLS 1.1 support will be discontinued in a future release.

### Warning

In a production environment, always enable SSL/TLS with the NAE interface. You should only disable SSL/TLS with NAE for troubleshooting purposes.

## Administrative Session Timeout

By default, there is no timeout for a `ksadmin` administrative session taking place through SSH or serial connection to the appliance.

We recommend setting a timeout so that `ksadmin` must re-authenticate after a period of inactivity.

### To set a timeout

1. Login via SSH as `ksadmin`.
2. Edit the `/home/ksadmin/.bashrc` file.

```
vi /home/ksadmin/.bashrc
```

3. Append the following line:

```
TMOUT=<desired timeout value in seconds>
```

4. Reload the `.bashrc` file to apply the new setting.

```
source /home/ksadmin/.bashrc
```

## Restricting Cluster Port

Configure `cluster allowlist` to only allow specific IPs to connect to the cluster port 5432.

## Virtual CipherTrust Manager Instance Management on Public and Private Clouds

The ability to launch, start, and stop a Virtual CipherTrust Manager instance should be restricted to specific authorized administrators, and considered a privileged operation.

## Disable Cloud-Init After Launching Virtual CipherTrust Manager Instances

The cloud-init service has a security vulnerability described in [KB0029319](#), which means that you must disable the cloud-init service after launching any Virtual CipherTrust Manager instance.

As you cannot resize the disk on an unencrypted instance with the cloud-init service disabled, we recommend [encrypting the disk](#).

1. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
2. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                                 | Required Private Key Format |
|-------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

3. Login as `ksadmin` to an SSH session or cloud web console.
4. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

## Install Virtual CipherTrust Manager

Virtual CipherTrust Manager instances are available in select private and public cloud environments.

### Virtual Deployment Environments

We support Virtual CipherTrust Manager in the following environments.

#### Public Cloud Images

- Amazon Web Services: SafeNet Cloud Provisioning System

Refer to: [AWS Deployment](#)

- Google Cloud:

Refer to: [Google Cloud Deployment](#)

- Microsoft Azure: Available as a BYOL image in the Microsoft Azure Marketplace

Refer to: [Azure Deployment](#)

- Oracle Cloud

Refer to: [Oracle Cloud Deployment](#)

- Alibaba Cloud

Refer to: [Alibaba Cloud Deployment](#)

#### Private Cloud Images

- VMware vSphere: OVA image file

Refer to: [Private Cloud Deployment](#)

- Microsoft Hyper-V: VHDX image file

Refer to: [Private Cloud Deployment](#)

- Nutanix AHV: VMDK image file



---

Refer to: [Private Cloud Deployment](#)

- OpenStack: qcow2 image file

This can be installed on to KVM or QEMU hypervisor.

An example command with KVM hypervisor is:

```
virt-install --virt-type kvm --name <virtual_image_name> --ram 2048 --disk path=<path_to_qcow2_image>,size=16,format=qcow2 --network network=default --graphics vnc,listen=0.0.0.0 --noautoconsole --os-type=linux --os-variant= ubuntu16.04 --import
```

- RedHat OpenShift: qcow2 image file

Refer to: [Private Cloud Deployment](#)

For network configuration on private cloud images, refer to [Network Interface Configuration Tutorial](#)

## Hybrid Cloud Images

- Azure Local: VHDX image file

Refer to: [Azure Stack Deployments](#)

- Azure Stack Hub: VHDX image file

Refer to: [Azure Stack Deployments](#)

## Cloud-Init Configuration

For Amazon Web Services, Microsoft Azure, Oracle Cloud, and Private Cloud environments, there are options to pass in a Cloud-init configuration (user-data) file to specify CipherTrust Manager settings. For more information, see [Plan Configuration Settings for Cloud Init](#) and [Deploying with Cloud Init](#).

## Plan Configuration Settings for Cloud Init

With a **user-data** file, you can set the initial password for the initial user, the default CA and server certificates, IPv6 network settings, a static IP, disk encryption, and HSM integration.

### Note

Cloud-init configuration (user-data) files uses YAML syntax; indentation is important and tabs can not be used.

## Configuring Initial Password for the Initial User

The initial password for the initial Application Administrator can be auto-generated using a cloud-init configuration file containing the following setting:

```
#cloud-config
keysecure:
 initial-user:
 generate-password: true
```

## Configuring Default CA and Server Certificates

Two certificates are created by default during initial boot-strapping of the CipherTrust Manager, one Self-signed Root CA Certificate and one Server Certificate that is signed by the Root CA.

### Note

CipherTrust Manager does not support `dns/ip atm`; only the `emails` field is supported, regardless of the `oid` setting.

## Configuring Initial Default Certificates

Before the first boot of the system, the content of these certificates can be overwritten with the help of a cloud-init configuration as follows:

```
#cloud-config
keysecure:
 default-ca:
 CN: <Common Name of Root CA Certificate, for example "Cloud-init test C
A">
 emails:
 - <Subject Alternate Name E-Mail, for example "foo@example.com">
 - ...
 names:
 - C: <Country, for example "US">
 ST: <State/province, for example "TX">
 L: <Location, for example "Austin">
 O: <Organization, for example "Gemalto Inc.">
 OU: <Organizational Unit, for example "Gemalto">
 - ...
 dns-names:
 - <Subject Alternate Name DNS-Name, for example "web.keysecure.loca
l">
 - ...
 ip-addresses:
 - <Subject Alternate Name IP-Address, for example "13.32.240.3">
 - ...
 default-server-cert:
 CN: <Common Name of Server Certificate, for example "Cloud-init test serv
er cert">
```

```

 emails:
 - <Subject Alternate Name E-Mail, for example "foo@example.com">
 - ...
 names:
 - C: <Country, for example "US">
 ST: <State/province, for example "TX">
 L: <Location, for example "Austin">
 O: <Organization, for example "Gemalto Inc.">
 OU: <Organizational Unit, for example "Gemalto">
 - ...
 dns-names:
 - <Subject Alternate Name DNS-Name, for example "web.keysecure.local">
 - ...
 ip-addresses:
 - <Subject Alternate Name IP-Address, for example "13.32.240.3">
 - ...

```

A working cloud-init configuration looks as follows:

```

#cloud-config
keysecure:
 default-ca:
 CN: Cloud-init test CA
 emails:
 - certadmin1@example.com
 - support1@example.com
 names:
 - C: US
 ST: TX
 L: Austin
 O: Gemalto Inc.
 OU: Gemalto
 - C: Sweden
 L: Gothenburg
 O: Gemalto AB
 dns-names:
 - web.keysecure.local
 - web.ciphertrustmanager.local
 ip-addresses:
 - 13.32.240.3
 - 93.184.216.34
 default-server-cert:
 CN: Cloud-init test server cert
 emails:
 - certadmin2@example.com
 - support2@example.com
 names:
 - C: US
 ST: MD
 L: Belcamp
 O: Gemalto Inc.
 OU: Gemalto
 - C: Sweden
 L: Stockholm
 O: Gemalto AB
 dns-names:

```

```

- web.keysecure.local
- web.ciphertrustmanager.local
ip-addresses:
- 13.32.240.3
- 93.184.216.34

```

The only required field is 'CN'. If it is missing the rest are ignored. If any of the remaining are missing, they will be empty when the certificate is created. If there is no default-ca or default-server-cert section, the defaults will be used for the missing section.

## Configuring IPv6

There are two formats for IPv6 configuration, 'auto' and 'dhcp'. All systems come configured to use 'auto' as the default.

To set 'dhcp' as the default, add the following cloud-init directive to the user-data:

```

#cloud-config
keysecure:
 netcfg:
 config_ipv6: dhcp

```

### Note

This parameter must be set before the first boot of the system since it is ignored after the first boot.

### Note

Do not use 'dhcp' on AWS if there are no IPv6 interfaces defined. This will result in a five-minute delay during each boot of your firmware.

## Configuring the CipherTrust Manager with a Static IP

The CipherTrust Manager OVA and Qcow2 disk images can be configured with a Static IP using cloud-init. The configuration format is as follows:

### Note

Static IP is only supported via cloud-init, and must be done on initial boot of a service.

```

#cloud-config

```

```

keysecure:
 netcfg:
 iface:
 name: <interface name>
 type: <static>
 address: <ip address to be assigned>
 netmask: <net mask in x.x.x.x format>
 gateway: <gateway ip address>
 dns1: <dns server's ip address, optional>
 dns2: <alternate dns server's ip address, optional>

```

The interface names vary between cloud providers. The current name for the first network interface for various clouds is listed in the table below. These names might change in the future depending on how a cloud provider presents the virtual network interface to the operating system.

| Cloud Provider      | First Network Interface Name |
|---------------------|------------------------------|
| Alibaba             | ens5                         |
| Amazon Web Services | ens5                         |
| Google Cloud        | ens4                         |
| Microsoft Azure     | eth0                         |
| Oracle Cloud        | ens3                         |
| VMWare vSphere      | ens32                        |

## Configuring Disk Encryption

For added security, the disk of CipherTrust Manager can be fully encrypted with the public SSH key.

Encryption can either be initiated when an instance is first launched with cloud-init or on an [already deployed instance](#) with CipherTrust Manager CLI commands.

After encrypting the disk, you need to unlock the encrypted instance on every boot using the 'ksctl diskenc secureboot' command and the private SSH key associated with the instance. See [To unlock an encrypted instance](#). Disk encryption is always applied, and this behavior cannot be disabled. If you wish to store your keys on an unencrypted CipherTrust Manager, you can deploy a new CipherTrust Manager and then use [backup and restore](#) to transfer keys and other data.

## Security Best Practices

- We always recommend enabling disk encryption.
- Because installation specific secrets are generated the first time a Virtual CipherTrust Manager instance is launched, it is recommended that the instance be encrypted at launch time when possible to ensure these secrets are never exposed.
- Provide an externally signed certificate to the preboot interface if possible. As you cannot provide this

certificate with cloud-init configuration, this is the preferred deployment sequence for Virtual CipherTrust Manager deployments:

- a. [Enable disk encryption with cloud-init.](#)
- b. Launch the Virtual CipherTrust Manager instance.
- c. [Upload an externally signed certificate.](#)
- d. Reboot the Virtual instance.
- e. Verify the preboot certificate when [unlocking the disk.](#)

## Encrypting an Instance at First Launch

To configure the instance to encrypt at launch, use the cloud-init settings below for **private cloud** deployments. You must provide the public SSH key you will associate with the CipherTrust Manager instance in OpenSSH format. The corresponding private key to unlock can be in OpenSSH, PKCS1, or PKCS8 format. The supported key algorithm is RSA. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security.

### Note

ssh-keygen produces OpenSSH format public keys by default. You can also use ssh-keygen to convert an existing PEM public key to OpenSSH with `ssh-keygen -y -f public_key.pem > public_key.pub`

```
#cloud-config
diskenc:
 encrypt: true
ssh_authorized_keys:
 - <replace with user's ssh public key>
```

For **public cloud** deployments on Amazon Web Services, Google Cloud, Microsoft Azure or Oracle Cloud, the SSH key is already provided during virtual image configuration, so you don't have to specify the SSH key in the **user-data** file.

```
#cloud-config
diskenc:
 encrypt: true
```

This file must be passed to the CipherTrust Manager instances, which uses cloud-init to enable the encryption. How this file is passed to the CipherTrust Manager appliance depends on the environment in which it is run. For other examples of passing cloud-init config.dat file in supported environments, go to [Deploying with Cloud Init](#).

## Checking Encryption Status

To check the encryption progress, you can run following CLI command:

```
ksctl diskenc status -p
```

This returns a response similar to:

```
Encrypting...
14.81 GiB / 15.52 GiB [=====>-----] 95.44% 11s
```

This command might time out during system restart or due to a slow connection. As an alternative, you can view the Console for the instance to see disk encryption progress.

## Unlocking an Encrypted Instance

```
ksctl diskenc secureboot -i <ssh key used to launch the instance>
```

### Note

We recommend [uploading an externally signed certificate to the preboot](#), to allow more secure unlocking.

The instance will start up after the encryption has finished. To unlock the instance, you must run following CLI command and provide the SSH private key in PEM format. OpenSSH format private keys are not supported. This is required to unlock the instance and allow admins and users to login and configure the Virtual CipherTrust Manager or access their keys.

## Configuring HSM Integration

The CipherTrust Manager can be configured to use an HSM as a root of trust. This can be done using cloud-init.

### Note

HSM configuration is now performed through the API. Using the cloud-init method is not recommended because it exposes sensitive information, such as the partition password, that can be viewed by anyone with access to the associated AWS console. For more information refer to [Hardware Security Module](#).

### Note

Before connecting an HSM to the CipherTrust Manager, the HSM must be set up and configured by an HSM Administrator. This includes creating a partition and partition password, and generating a client certificate for the CipherTrust Manager.

To enable HSM integration using cloud-init, user-data is supplied to cloud-init. This file must be passed to the

CipherTrust Manager instances that use cloud-init to enable the encryption. How this file is passed to the CipherTrust Manager appliance depends on the environment in which it is run. For a list of supported environments, see [Overview](#).

### Note

Cloud-init configuration (user-data) files use YAML syntax; indentation is important and tabs can not be used.

## Example - Configuring CipherTrust Manager to use SafeNet Luna Network HSM

```
#cloud-config
keysecure:
 hsm:
 type: luna
 # Replace dummy partition name and password values below with your existing Luna HSM settings.
 conninfo: '{"partition_name": "kylo-partition", "partition_password": "sOm eP@ssword"}'
 config:
 # luna host name or ip address
 host: 172.20.32.11
 # luna server PEM encoded cert
 server-cert: |
 -----BEGIN CERTIFICATE-----
 ...
 -----END CERTIFICATE-----
 # luna client PEM encoded cert
 client-cert: |
 -----BEGIN CERTIFICATE-----
 ...
 -----END CERTIFICATE-----
 # luna client PEM encoded cert private key
 client-cert-key: |
 -----BEGIN RSA PRIVATE KEY-----
 Proc-Type: 4, ENCRYPTED
 DEK-Info: DES-EDE3-CBC, 92852AD5AFE768B8
 ...
 -----END RSA PRIVATE KEY-----
```

## Example - Configuring CipherTrust Manager to use AWS CloudHSM (cavium)

```
#cloud-config
keysecure:
 hsm:
 type: aws
 # Replace dummy crypto user credentials below with your actual CloudHSM C
 U credentials.
 conninfo: '{"partition_name": "cavium", "partition_password": "hsmuser:password"}'
 config:
```



```
CloudHSM Cluster ENI IP Address
host: 172.20.32.11
CloudHSM Cluster certificate
server-cert: |
 ----BEGIN CERTIFICATE----
 ...
 ----END CERTIFICATE----
```

## Example - Configuring CipherTrust Manager to DPoD Luna Cloud HSM service

### Note

DPoD cloud configurations are sometimes above the AWS size limit for user data. Verify the user data is 16KB or less, otherwise it will be ignored by AWS. If it is larger then remove unnecessary whitespace (e.g. indent lines with a single space) and comments.

```
#cloud-config
keysecure:
 hsm:
 # type: Specifies HSM type as dpod.
 type: dpod

 # conninfo: "partition_name" and "partition_password"
 # {
 # "partition_name": '', # HSM partition name provided during pa
rtition provisioning
 # "partition_password": '', # HSM Crypto Officer password
 # }
 conninfo: '{"partition_name": "kylo-partition","partition_passwor
d": "sOmeP@ssword"}'

 # config: include the following attributes:
 # {
 # 'server_cert': '', # Server certificate in PEM format from se
rver-certificate.pem file.
 # 'partition_cert': '', # Partition certificate in PEM format f
rom partition-certificate.pem file
 # 'partition_ca_cert': '', # Partition CA certificate in PEM fo
rmat from partition-ca-certificate.pem file
 # 'cv_app_specific_data': '', # CVAppSpecificData value from Ch
rystoki.conf file
 # 'cv_partition_data': '', # PartitionData00 value from Chrysto
ki.conf file
 # }
 #
 config:
 server_cert: |
 -----BEGIN CERTIFICATE-----
 ...
 -----END CERTIFICATE-----
 partition_cert: |
 -----BEGIN CERTIFICATE-----
```

```

...
-----END CERTIFICATE-----
partition_ca_cert: |
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
cv_app_specific_data: <CVAppSpecificData>
cv_partition_data: <PartitionData00>

```

## Deploying with Cloud-init

The Virtual CipherTrust Manager supports initial boot time configuration using cloud-init. On first boot, the appliance can be passed a configuration text file, known as a **user-data** file. The Virtual CipherTrust Manager reads this file to retrieve various configuration settings. To plan your configuration settings in the **user-data** file, see [Plan Configuration Settings for Deploying with Cloud-Init](#).

How a cloud-init configuration file is passed to the Virtual CipherTrust Manager appliance depends on the environment in which it runs. Below are examples for each of the supported environments.

### Note

Cloud-init configuration (user-data) files uses YAML syntax; indentation is important and tabs can not be used.

## Passing the cloud-init file in AWS

When launching the instance from AWS web , at Step 3 - Configure Instance Details, select 'As file' in the 'User data' section and browse to 'config.dat' file. Then continue with the rest of the launch steps.

If launching the instance using AWS CLI, then use the '--user-data' parameter with the run-instances command. Use the 'file://' prefix to pass in the user data from 'config.dat' file.

### Note

Verify the user data is 16KB or less, otherwise it will be ignored by AWS. If it is larger then remove unnecessary whitespace (e.g. indent lines with a single space) and comments.

## Passing the cloud-init file in Azure

When launching the Azure instance using Azure CLI, use the '-d' or '--custom-data' flag with the `vm create` command to pass in the user data from 'config.dat' file.

## Passing the cloud-init file for a Private Cloud Image

When using the disk image, how the cloud-init data is passed to CipherTrust Manager depends on the virtualization platform - please refer to documentation or notes for your specific cloud environment.

The following are two examples of passing cloud-init data when using a disk image, one using 'libvirt' and one using VMware/vSphere.

### Example using 'libvirt'

When launching a virtual machine with the Qcow2 image using 'libvirt', the cloud-init data has to be passed in as an ISO file. The ISO can be generated as follows:

1. Prepare the **user-data** file as follows:
  - a. Rename the file **config.data** to **user-data**.
  - b. Because the user's SSH key is used for wrapping a layer of encryption keys, it must be added to the cloud-init config. So, the 'user-data' file should look like:

```
#cloud-config
hostname: <host name for the instance>
diskenc:
 encrypt: true
ssh_authorized_keys:
 - <replace with user ssh public key>
```

#### Note

'ssh\_authorized\_keys' can be configured with multiple ssh public keys.

2. Create a **meta-data** file and provide instance parameters, for example:

```
instance-id: <some instance id>
```

3. Create an ISO image file:
  - a. Make sure **genisoimage** utility is installed.
  - b. Create the ISO file.

```
genisoimage -o config.iso -volid cidata -joliet -rock user-data meta-data
```

4. Launch instance using **virt-install**. OpenStack example:

```
virt-install --virt-type kvm --name <virtual_image_name> --ram 2048 --disk path=
h=<path_to_qcow2_image>,size=16,format=qcow2 --disk path=<path_to_config.iso>
--network network=default --graphics vnc,listen=0.0.0.0 --noautoconsole --os-t
ype=linux --os-variant= ubuntu16.04 --import
```

## Example using VMware/vSphere

This example describes how to deploy CipherTrust Manager on VMware with a static IP configuration. In general, if you have virtual machines you intend to use frequently or for extended periods of time, it can be convenient to assign a static IP address, or configure DHCP server to always assign the same IP address, to each of these virtual machines.

For virtual machines that you do not expect to keep for extended periods of time, use DHCP and let it allocate IP addresses for these machines.

Use the following procedure to deploy CipherTrust Manager on VMware with a static IP configuration.

### Note

This procedure includes preparation of a cloud-init configuration file used to set up a static IP address during launch of the CipherTrust Manager.

1. Get CipherTrust Manager installation file for VMware from Gemalto Support Portal.
2. Deploy OVA in ESXi Server.
  - a. Select “Deploy OVF Template”.
  - b. On the **Select an OVF template** page, choose OVA file and select **NEXT**.

## Deploy OVF Template

### 1 Select an OVF template

#### 2 Select a name and folder

#### 3 Select a compute resource

#### 4 Review details

#### 5 Select storage

#### 6 Ready to complete

### Select an OVF template

Select an OVF template from remote URL or local file system

Enter a URL to download and install the OVF package from the Internet, or browse to a location accessible from your computer, such as a local hard drive, a network share, or a CD/DVD drive.

☐ URL

[http | https://remoteserver-address/filetoinstall.ovf](http://https://remoteserver-address/filetoinstall.ovf)

☒ Local file

Choose Files

610-000121-001...1.4.0\_RevA.ova

CANCEL

BACK

NEXT

- On the **Select a name and folder** page, select the name of the virtual machine and its location.
- Validate the CipherTrust Manager Virtual Machine Configuration. On the **Select a compute resource** page, select the destination compute resource (if applicable).

## Deploy OVF Template

### ✓ 1 Select an OVF template

### ✓ 2 Select a name and folder

### 3 Select a compute resource

#### 4 Review details

⚠ OVF package with compressed disks is currently not supported for OVF import.

### Select a compute resource

Select the destination compute resource for this operation

SE vDC

If you see the above error, please perform the following steps:

- Use `ovftool.exe` to convert ova file into uncompressed file(s).

Execute the following command:

```
ovftool.exe --lax <source_OVA_file> <destination_OVF_file>
```

### Note

OVF with compressed disk is not supported on newer version of Vsphere client. It may work on older versions.

- b. Repeat Step 2 with a new installer file.
5. On the **Review details** page, verify the template details of the CipherTrust Manager image and if correct, select **NEXT**.

## Deploy OVF Template

- ✓ 1 Select an OVF template
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- 4 Review details**
- 5 Select storage
- 6 Select networks
- 7 Ready to complete

### Review details

Verify the template details.

|               |                             |
|---------------|-----------------------------|
| Publisher     | No certificate present      |
| Product       | KeySecure k170v             |
| Version       | 1.4.0.2158                  |
| Vendor        | Gemalto NV                  |
| Download size | 1.8 GB                      |
| Size on disk  | 4.9 GB (thin provisioned)   |
|               | 16.0 GB (thick provisioned) |

CANCEL

BACK

NEXT

- a. On the **Select storage** page, select the storage location to install the CipherTrust Manager and then select **NEXT**.

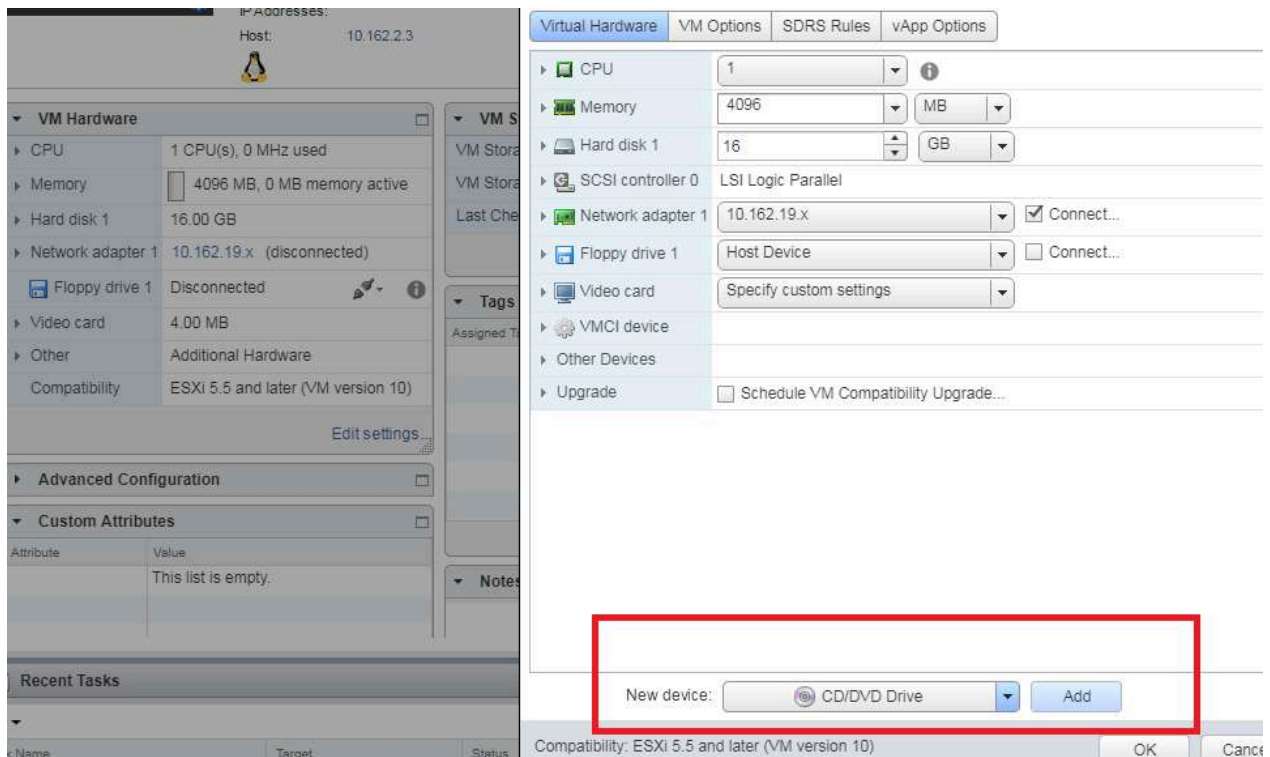
- b. On the **Select network** page, select the network and then select **FINISH**.

### Caution

Do not launch/start the machine at this time.

6. Prepare the cloud-init configuration.

- a. Add a CD drive to the VM.



- b. Before booting up the VM, prepare the cloud-init configuration. The following cloud-init example configures the VM's ens32 port with a static IP address. Copy this example and edit it for your desired network settings.

```
#cloud-config
keysecure:
 netcfg:
 iface:
 name: ens32
 type: static
 address: 192.168.1.150
 netmask: 255.255.255.0
 gateway: 192.168.1.1
```

```
dns1: 192.168.1.100
```

### Note

Cloud-init configuration files use YAML syntax; indentation is important and tabs cannot be used.

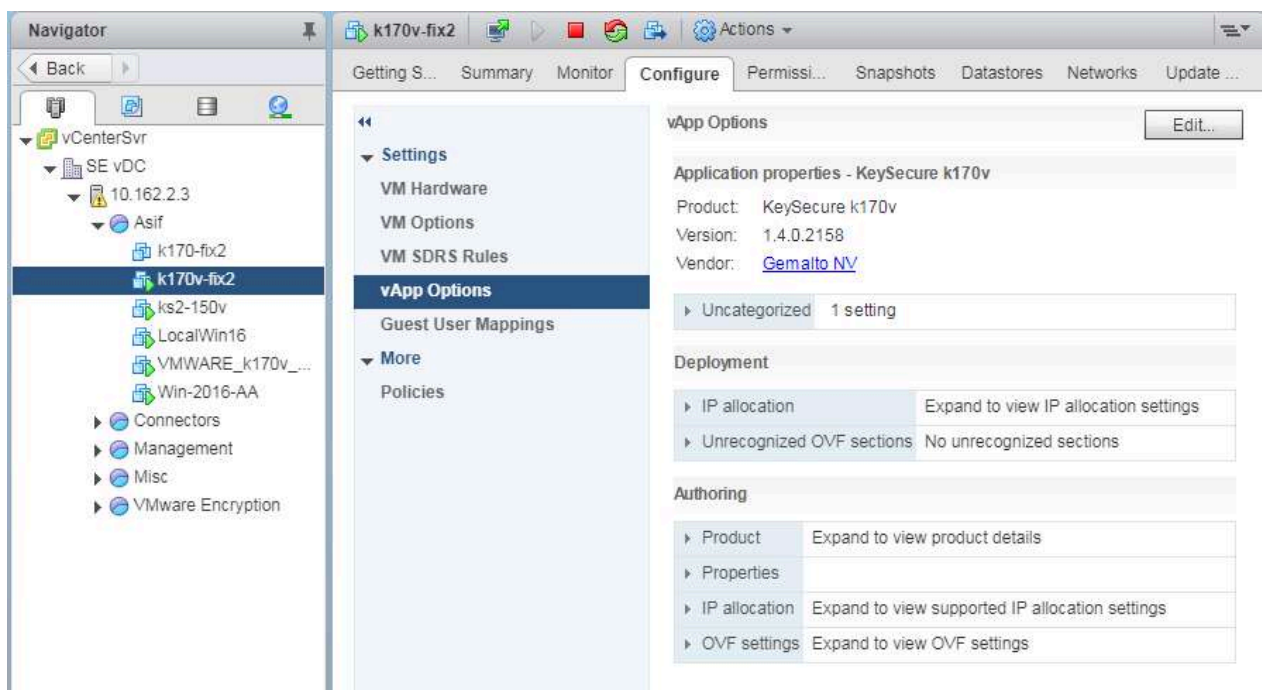
- c. Convert the string to base64. To convert to base64, use the `openssl` command:

```
openssl base64 -in <infile> -out <outfile>
```

- d. Save this base64 string to use in next steps.

7. Add the base64 configuration to the VM. This step shows vSphere web client (Flash) version as demonstration. You may find similar options in other clients.

- a. Select: *virtual machine* > Configure > Settings > vApp Options



- b. Press **Edit** button on the top right on this page.
- c. Under OVF Settings, select the **ISO Image** check box, which is next to OVF environment transport.



**kesecure 170v-2 - Edit Settings**

Virtual Hardware | VM Options | SDRS Rules | **vApp Options**

Product: KeySecure k170v  
 Version: 1.4.0.2158  
 Vendor: [Gemalto NV](#)

*There are no properties to edit*

**Deployment**

|                             |                                   |
|-----------------------------|-----------------------------------|
| ▶ IP allocation             | Expand for IP allocation settings |
| ▶ Unrecognized OVF sections | No unrecognized sections          |

**Authoring**

|                           |                                                 |                                                                                                        |
|---------------------------|-------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| ▶ Product                 | Name                                            | KeySecure k170v                                                                                        |
| ▶ Properties              |                                                 |                                                                                                        |
| ▶ IP allocation           | Expand to edit supported IP allocation settings |                                                                                                        |
| ▶ OVF settings            |                                                 |                                                                                                        |
| OVF environment           | <a href="#">View...</a>                         | The OVF environment is only available when the VM is powered on.                                       |
| OVF environment transport | <input checked="" type="checkbox"/> ISO image   | An ISO image, containing the OVF environment document, is mounted on the first available CD-ROM drive. |
|                           | <input type="checkbox"/> VMware Tools           | The VMware tools guestInfo.ovfEnv variable is initialized with the OVF environment document.           |
| Installation boot         | <input type="checkbox"/> Enable                 | The installation boot automatically gets                                                               |

Compatibility: ESXi 5.5 and later (VM version 10)

OK Cancel

- d. On the same page, expand "Properties" to add configuration.

**kesecure 170v-2 - Edit Settings**

Virtual Hardware | VM Options | SDRS Rules | **vApp Options**

Product: KeySecure k170v  
Version: 1.4.0.2158  
Vendor: [Gemalto NV](#)

*There are no properties to edit*

**Deployment**

- IP allocation *Expand for IP allocation settings*
- Unrecognized OVF sections *No unrecognized sections*

**Authoring**

Product Name: KeySecure k170v

**Properties**

| Key | Label | Category | Type |
|-----|-------|----------|------|
|     |       |          |      |
|     |       |          |      |
|     |       |          |      |
|     |       |          |      |
|     |       |          |      |
|     |       |          |      |

New Edit Delete

Compatibility: ESXi 5.5 and later (VM version 10)

OK Cancel

e. Press the **New** button to add a property for the configuration. On following screen, there are two fields which need to be changed.

- **Label:** user-data
- **Default value:** <base64 string of configuration, generated in step6-b>

#### Note

Key ID will change automatically when you change Label.

**Edit Property Settings**

Category: [Dropdown]

Label: user-data

Key class ID: [Field]

Key ID: user-data

Key instance ID: [Field]

Description: [Text Area]

Type: ☒ Static property

Type: String

Length: 0 - 65535

Default value: 3kbnMxOiAxMC4xNjluMTkuMTAwCg==

User configurable: ☒

☐ Dynamic property

Macro: IP address

Network: 10.162.19.x

OK Cancel

f. Press **OK** to save the Property Settings.

g. Then press **OK** again to save the vApp Options page.

8. Launch the instance. The VM should boot up configured with a static IP.

## Amazon Web Services Deployment

You can deploy a CipherTrust Manager image within Amazon Web Services (AWS).

### Minimum Requirements

To deploy a Virtual CipherTrust Manager instance, the following minimum requirements apply:

- System volume: 300 GB

- Memory: 16 GB
- vCPUs:
  - 2 for unclustered, standalone Virtual CipherTrust Managers
  - 4 for **clustered** Virtual CipherTrust Managers
- NICs: 1

#### 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 CipherTrust Manager images (AMI) are obtained by provisioning. Provisioning places a CipherTrust Manager AMI in your target AWS account in the target region(s).

Use the Thales Customer Support Portal to request provisioning for a CipherTrust Manager 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 **CipherTrust Manager** 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 CipherTrust Manager image.

## Deploying in AWS

This section provides the steps for deploying a Virtual CipherTrust Manager instance in AWS.

### Prerequisites

- A [provisioned](#) Virtual CipherTrust Manager AMI.
- If using a Windows client use PuTTY or similar utility to SSH to your CipherTrust Manager instance as KeySecure Administrator (ksadmin).

If needed, use PuTTYgen or similar utility to format the SSH Key Pair. We support OpenSSH format and RSA key algorithm for the public key, and OpenSSH, PKCS1, or PKCS8 format for the private key. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security.

- If using a Linux client use SSH to login as KeySecure Administrator (ksadmin).

### To launch a CipherTrust Manager instance in AWS

More detailed descriptions of the launch instance wizard settings are available in [Amazon EC2 documentation](#).

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. Under the **Instances>Launch Instance**
  - a. Provide a **Name** and **Tags** for the new instance, if desired.
  - b. Select **My AMIs** and search for the AMI (e.g. ami-1f6b5b60) that was provisioned for you by the **SafeNet Cloud Provisioning System**.
5. Next to the image name, click on **Select**.

The AWS instance launch page opens.

6. In the **Instance Type** section, choose a size that meets the [minimum requirements](#).
7. Select which **Key Pair (login)** option to use. We support OpenSSH for the public key format, and OpenSSH, PKCS1 or PKCS8 for the corresponding private key 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.)

### 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.

8. In the **Network Settings** section.
  - a. If applicable to your AWS account settings, you are presented with a setting for **Networking platform**, which allows you to choose **Virtual Private Cloud (VPC)**.
  - b. In most cases, select an existing **VPC** and **Subnet**. You can also create a new **Subnet**.
  - c. Set **Auto-assign Public IP** to **Enabled**.
  - d. Under **Firewall (security groups)**, create a new security group or re-use a previously created group.

**Note**

Refer to [Network Security Groups](#). Consider adding all of the recommended rules.

To launch the CipherTrust Manager, you need at a minimum Port 22 inbound/outbound for **ksadmin**'s SSH connection.

For deployments from most AWS locations, you need port 443 inbound for an HTTPS connection to the CipherTrust Manager GUI, CLI, or REST API. For deployments from AWS China, we recommend port 9443 inbound for an HTTPS connection to the CipherTrust Manager GUI, CLI, or REST API.

9. In the **Configure Storage** section, select the desired **Size (GiB)**, minimum being **300 GiB**.

10. At the bottom of the screen there is a section **Advanced Settings**.

Any information that you need configured before the first boot must be entered here in the **User data** area. See [Plan Configuration Settings for Cloud Init](#).

11. In the **Summary** section to the right:
  - a. Review your launch configuration settings.
  - b. Make any changes necessary on the sections to the left.
  - c. After your review, select **Launch instance**.

It can take a several minutes for your instance to launch.

12. 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.

At this point, if you wish to connect as System Administrator (ksadmin) to perform system level duties, duties such as performing network configurations, system upgrades, continue with the next step.

**Note**

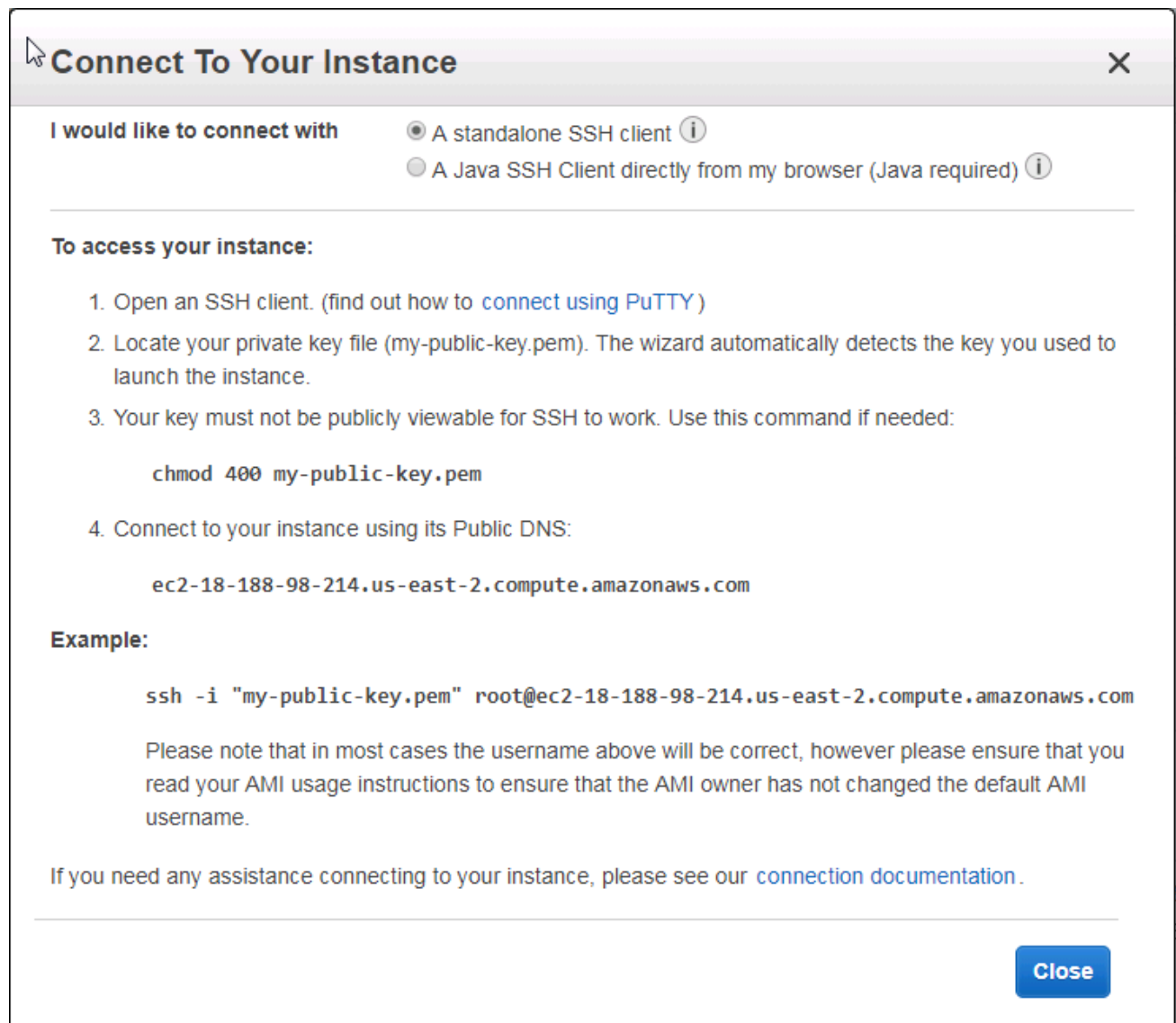
Refer to [Groups](#) for a more detailed list of ksadmin duties.

Otherwise, browse to step 18 to connect to the **CipherTrust Manager Web Page**.

13. Select your instance in the table and then select **Connect**.



The **Connect To Your Instance** dialog box appears:

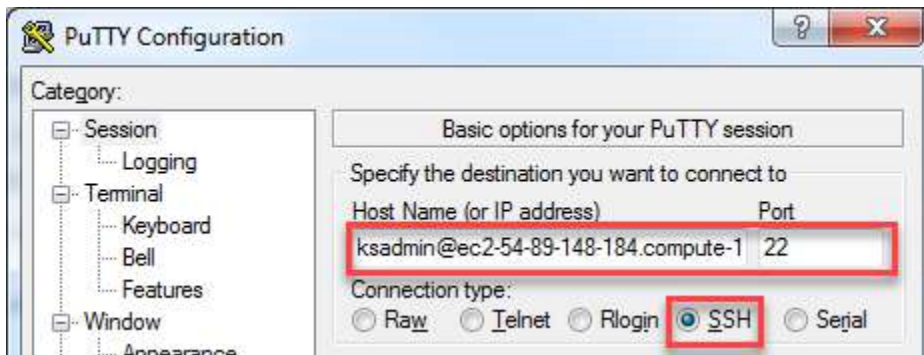


14. Copy the instance **Public DNS** string, for example: `ec2-54-89-148-184.compute-1.amazonaws.com`

15. If using PuTTY, begin a new session.

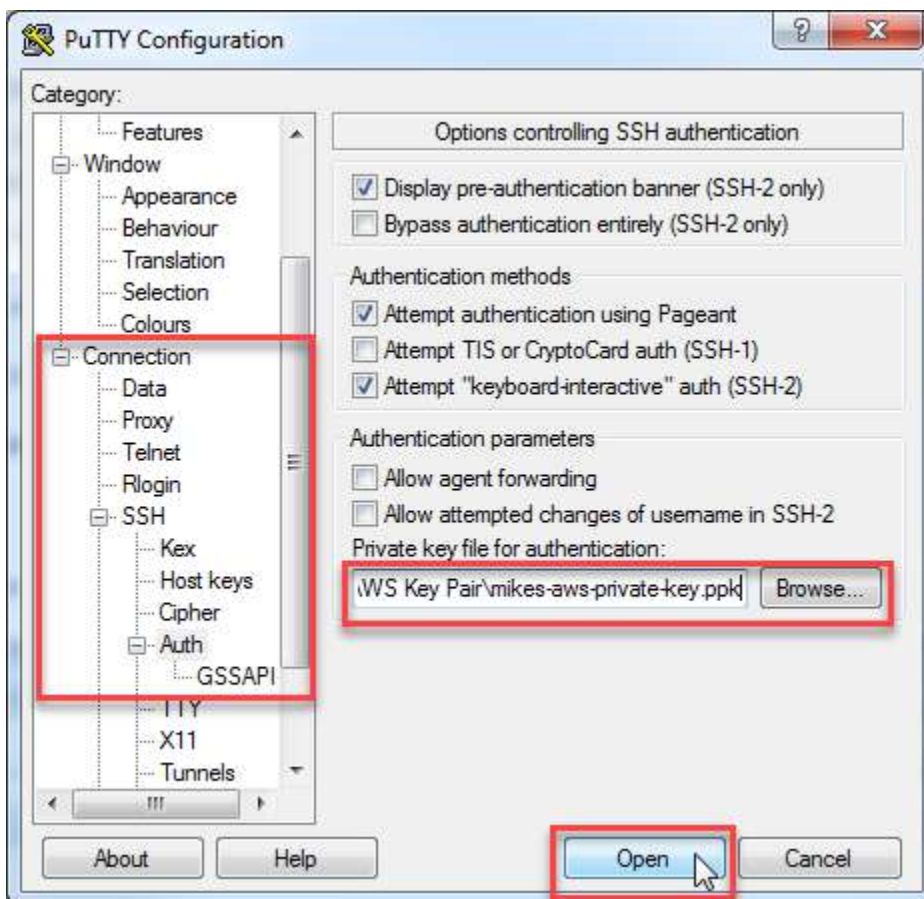


- a. Paste the Public DNS in the **Host Name (or IP address)** field and preface this string with "ksadmin@".
- b. Set the **Connection Type** to **SSH**.



16. 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**. The following **ksadmin** screen appears:

[illegible]

### 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.

17. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
18. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                                            | Required Private Key Format |
|-------------------------|-------------------------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i<br/>&lt;path_to_private_SSH_key&gt;<br/>&gt; &lt;archive_file_name&gt;<br/>ksadmin@&lt;ip&gt;:.</pre> | OpenSSH format              |
| Windows                 | <pre>pscp -i</pre>                                                                                                | PuTTY PPK format            |

| Source Operating System | Example Command Syntax                                                                         | Required Private Key Format |
|-------------------------|------------------------------------------------------------------------------------------------|-----------------------------|
|                         | <pre>&lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> |                             |

19. Login as `ksadmin` to an SSH session or cloud web console.
20. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

21. Proceed to connect to the CipherTrust Manager GUI. The steps depend on which AWS Marketplace you launched from.

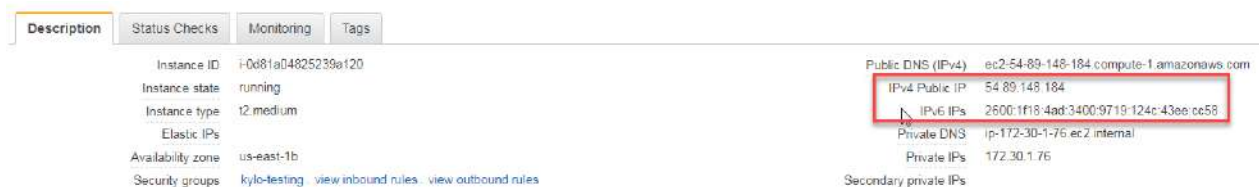
- Connect from AWS China
- Connect from all other locations

## To connect to the CipherTrust Manager GUI from most AWS Marketplace Locations

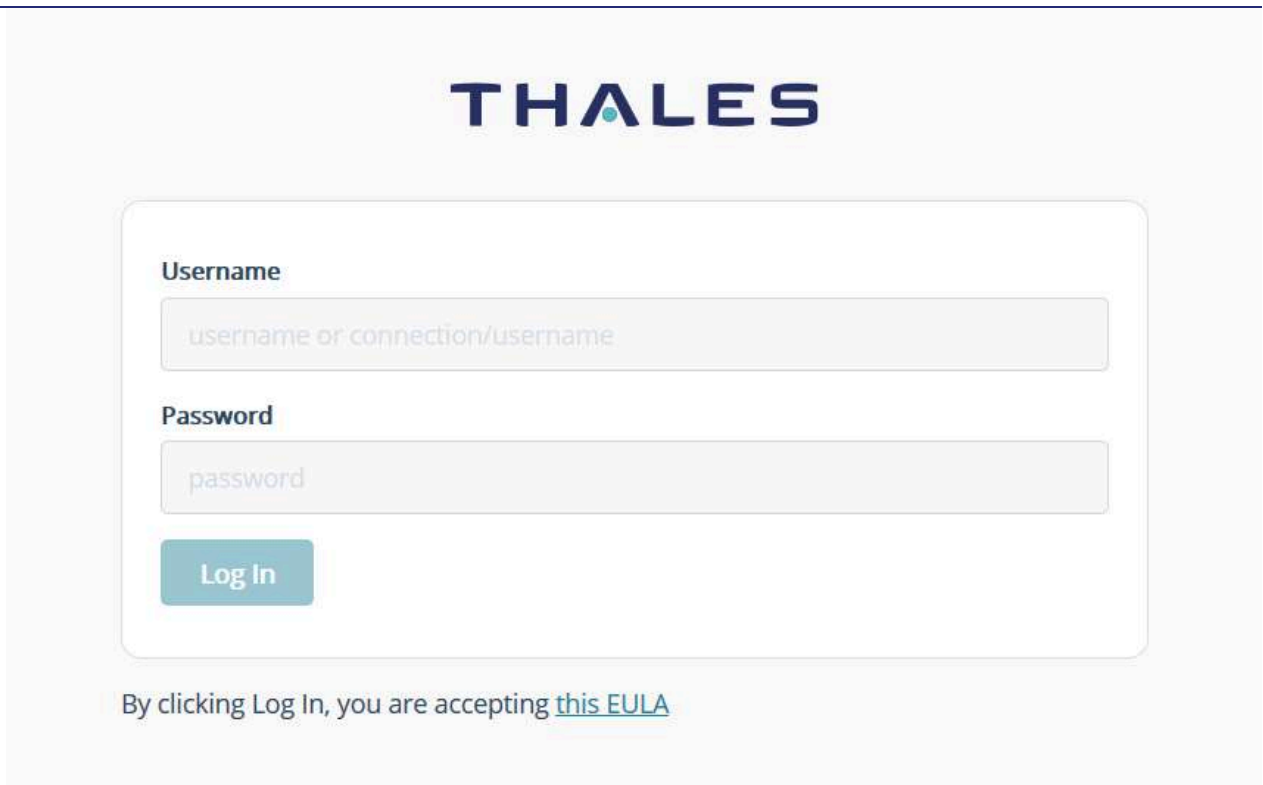
### Note

There are different steps to connect to the CipherTrust Manager GUI from AWS China.

1. Connect to the **CipherTrust Manager** Web Page.
  - a. Copy the **IPv4 Public IP** or **IPv6 IP** address shown in the **Instances > Description** tab of your AWS instance.



- b. Browse to `HTTPS://<ip address>`. The **Log In** screen appears.



The image shows the Thales login interface. At the top is the Thales logo. Below it is a login form with two input fields: 'Username' and 'Password'. The 'Username' field has a placeholder text 'username or connection/username'. The 'Password' field has a placeholder text 'password'. Below the fields is a 'Log In' button. At the bottom of the form, there is a text line: 'By clicking Log In, you are accepting [this EULA](#)'.

2. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



#### Note

If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [Changing the Initial Password](#).

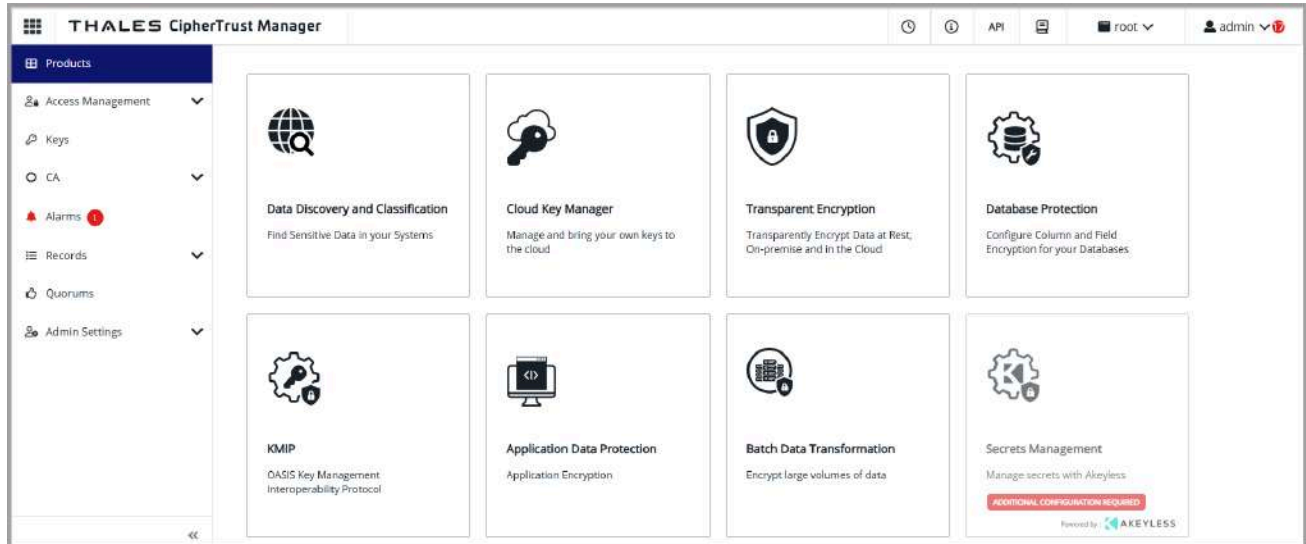
3. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
```

Min number of other characters: 1

A new Login screen appears.

4. Using your new password, log in again. The **CipherTrust Manager** Web Page appears.



5. At this point, it's strongly recommended to configure an NTP server.
  - a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

Congratulations! You have successfully deployed your Virtual CipherTrust Manager instance.

### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

## To connect to the CipherTrust Manager GUI from AWS China

There are different steps to [connect to the CipherTrust Manager GUI from other AWS marketplace locations](#).

If you are launching from AWS China, you must change the web port from the default 443 value to 9443.

1. Launch a Linux or Windows EC2 instance in the same VPC as the Virtual CipherTrust Manager, following AWS documentation.

This instance will act as a client to the Virtual CipherTrust Manager to issue the command to change the web port.

2. Obtain and install the `ksctl` CLI tool on the Linux or Windows EC2 instance. You can obtain `ksctl` from either an existing CipherTrust Manager or through Thales customer support.
3. Copy the **Private IPs** address shown in the **Instances > Description** tab of your AWS instance.
4. In the `ksctl config.yaml` file, apply the following settings:

```
KSCTL_USERNAME: admin
KSCTL_PASSWORD: admin
KSCTL_URL: https://<Private_IP_address_from_AWS>:443
KSCTL_NOSSLVERIFY: true
```

5. In `ksctl`, change the password for `admin` from the default, providing the `config.yaml` file.

The default Password Policy applies:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

To change the admin password:

```
ksctl changepw --new-pw <desired-password> --confirm-pw <desired-password> <path_to_config_file>config.yaml
```

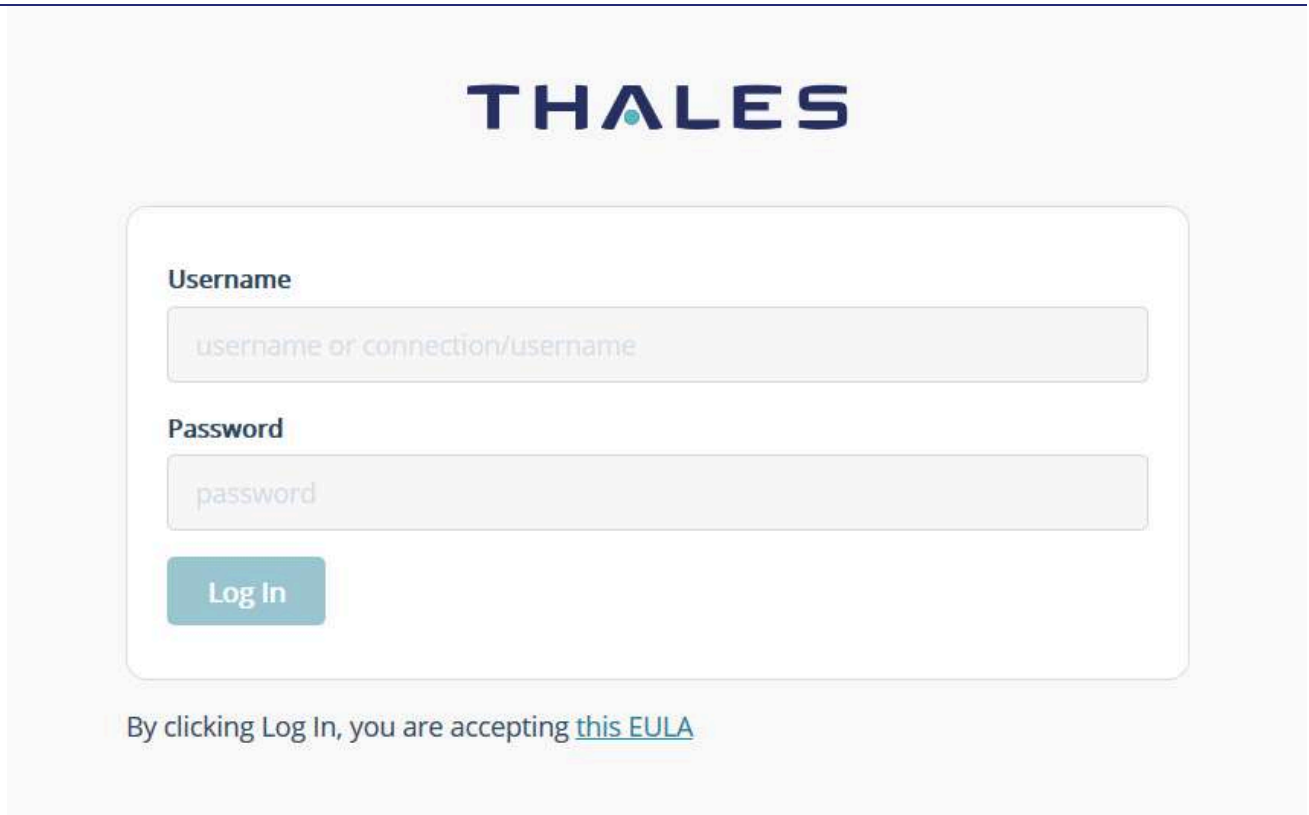
You are asked to confirm this action.

6. Run the following command to change the web port to 9443:

```
ksctl interfaces modify --name web --port 9443
```

You are asked to confirm this action, and to acknowledge that the service will become unavailable temporarily.

7. Use a laptop or PC outside the VPC to connect to the **CipherTrust Manager** Web Page by navigating to `HTTPS://<public_ip_address>:9443`. The **Log In** screen appears.



**THALES**

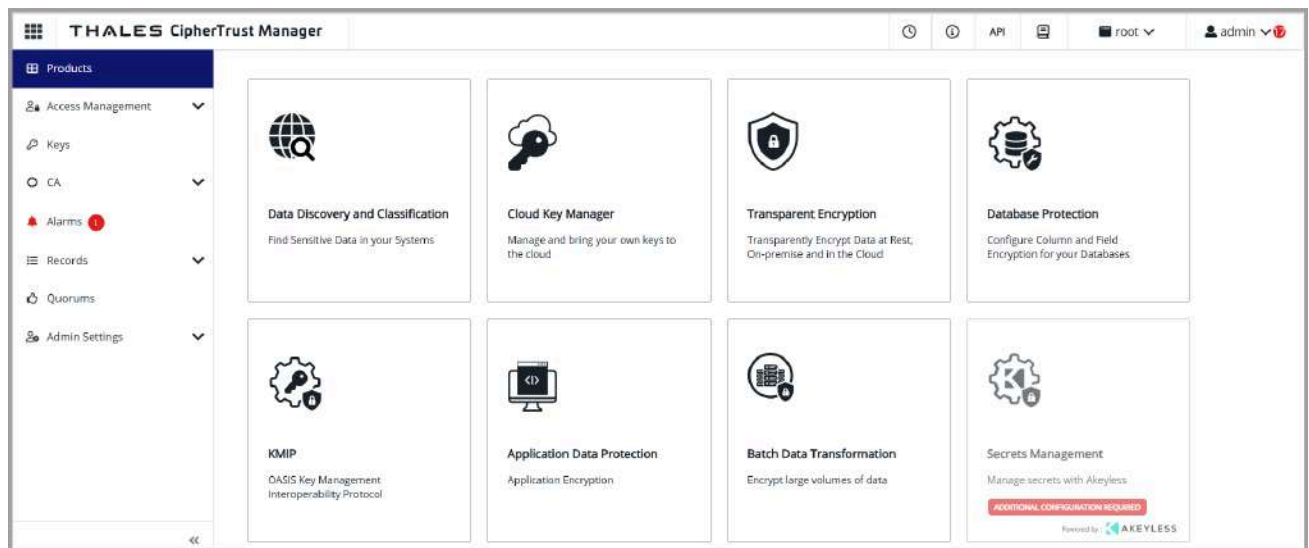
**Username**

**Password**

**Log In**

By clicking Log In, you are accepting [this EULA](#)

8. Using your new password, log in as **admin**. The **CipherTrust Manager Web Page** appears.



9. At this point, it's strongly recommended to configure an NTP server.
- a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.

- 
- d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

Congratulations! You have successfully deployed your Virtual CipherTrust Manager instance.

#### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

## Enhanced Networking

The CipherTrust Manager AMI ships with Elastic Network Adapter (ENA) enabled. ENA provides high-performance networking on supported EC2 instance types. For more information visit <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html>.

## Disk Resize Available with Disk Encryption

After you launch an unencrypted Virtual CipherTrust Manager and disable the cloud-init service, you cannot resize the disk. If you require a Virtual CipherTrust Manager with a larger or smaller disk size:

- [Encrypt the disk](#) for the Virtual CipherTrust Manager instance and then attempt re-size.
- [Take a backup](#) on the existing Virtual CipherTrust Manager, deploy a new Virtual CipherTrust Manager instance with the desired disk size, and restore the backup file on the new instance.

## Google Cloud Deployment

You can deploy a CipherTrust Manager instance in Google Cloud.

## Minimum Requirements

To deploy a Virtual CipherTrust Manager instance, the following minimum requirements apply:

- System volume: 300 GB
- Memory: 16 GB
- vCPUs:
  - 2 for unclustered, standalone Virtual CipherTrust Managers
  - 4 for [clustered](#) Virtual CipherTrust Managers
- NICs: 1



**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 Google Cloud

This section provides the steps for deploying a Virtual CipherTrust Manager instance from the Google Cloud Marketplace. Refer to the Google Cloud Platform documentation for general information on launching a VM in Google Cloud.

**Note**

We recommend first launching the Virtual CipherTrust Manager instance, and then providing the SSH key for the `ksadmin` user using the CipherTrust Manager web UI, as presented in the steps below. If you wish to instead use Google Cloud tools to provide the SSH key, you must provide the username `ubuntu`.

## Prerequisite

If you wish to set a static IP for your Virtual CipherTrust Manager instance, you must reserve a static external IP address for your project before launching the VM. Consult [Google documentation](#) to do so.

## To launch a CipherTrust Manager instance

1. Using your gmail address, sign in to the Google Cloud Platform.
2. In the **Google Cloud Platform** top banner, select a project or create a new one.



To create a New Project:

- a. Select **New Project**.



- b. Enter a **Project Name** and **Project ID**.

- 
- c. Enter a **Location** if you already have a parent organization you want to use.
  - d. Select **Create**.

The project's **DASHBOARD** appears.

3. Navigate to the [Google Cloud Marketplace](#) and search for "CipherTrust Manager" images.
4. On the results, select **CipherTrust Manager Community Edition**.
5. Select **Launch**.
6. To create a CipherTrust Manager instance, review and modify these fields as needed:
  - a. Enter a **Deployment name** for the instance.
  - b. Select the **Zone** closest to your location.
  - c. Review that the **Machine type** is **e2-standard-4 (4vCPU, 16 GB memory)**, which is optimized for Virtual CipherTrust Manager. If you want to change the **Machine type**, make sure the selection is in keeping with the [minimum requirements](#).
  - d. Review **Boot disk type** and **Boot disk size in GB**. The default of **Standard Persistent Disk** and **100** fulfill the minimum requirements of the image.
  - e. If desired, in the **Network interface** section **External IP** drop down, you can select any static IP which is reserved for your project. Consult [Google documentation](#) on reserving a static IP address for a project.

#### Note

After launching the VM, you can confirm the static IP was applied using the nmcli tool in an [SSH session](#).

- f. In the **Firewall** section, ensure that checkboxes for **Allow HTTP traffic** and **Allow HTTPS traffic** are enabled.
  - g. Enable the checkbox to accept the **GCP Marketplace Terms of Service** and **Thales - European Union - Frankfurt Terms of Service**.
7. Select **Deploy** to launch the instance.

Your new virtual CipherTrust Manager is created and appears in **Deployment Manager**.

---

 **CipherTrust Manager**

Solution provided by Thales - European Union - Frankfurt

|                       |                                                                                                                                                     |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Site address          | <a href="https://34.73.62.144:443/">https://34.73.62.144:443/</a>  |
| Instance              | <a href="#">ciphertrust-manager-1-vm</a>                                                                                                            |
| Instance zone         | us-east1-c                                                                                                                                          |
| Instance machine type | e2-standard-4                                                                                                                                       |

▼ MORE ABOUT THE SOFTWARE

## Get started with CipherTrust Manager

[VISIT THE SITE](#) [SSH](#)

- Click on the **Site address** for the newly created vCM.

A new browser tab opens to this address and the **CipherTrust Manager** web page appears.

- You are prompted to enter an SSH key to authenticate the `ksadmin` user in an **SSH session**.

# THALES

## Error



- Following services are starting up: platform, ciphertrust-cloud-key-manager, scheduler, backup, key-vault, crypto, licensing, nae, kmip
- The CipherTrust Manager will be functional after the default SSH public key for the ksadmin user is replaced

## SSH Public Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQBDiLlOH4tInSyWQuJkBLJ2sdce4BsvaFa5eL
6Xh1xgELXArNpjZtWvikQkz3JD/2TZbyzdLVACvfoVIFx2MToOw83WsuxeUua95NW
r1ehqeaS9Bx1
/v5OkogEuTzYN1gXyzQQbfbPEOhIDR+y7FNvlyw5++ZUhi2iMIHQEPeZlhoTYHXG1
RpR438JXoh/C8ey8aaP6cdZcTudQdsdDP3HGWXjsP
/NpOw4yeBthoy5NFuPpmS0Uvd0k2uCYkq6bkW7wwiRbW21joGBEf2muyKiCc2Fa
uVIFvgQNORaj1WjnRcWOjMlf2slwue8h2YsAfnzeB/gYhH+gdxfdK8aZ
user@example.local
```

Add

## Note

We support OpenSSH for the public key format. The corresponding private key can be OpenSSH, PKCS1, or PKCS8 format. You can generate this key pair using 'PuTTYgen' or similar utility. Save this SSH Public Key at a safe location. You will need this key for future SSH access.

After replacing the default SSH Public Key, the Log In screen appears. For more options to replace the default SSH Public Key, see [Starting Services After Deployment](#).

10. Log in using the initial default credentials for the root administrator: Username = admin, Password = admin

The following notice is displayed:



### Note

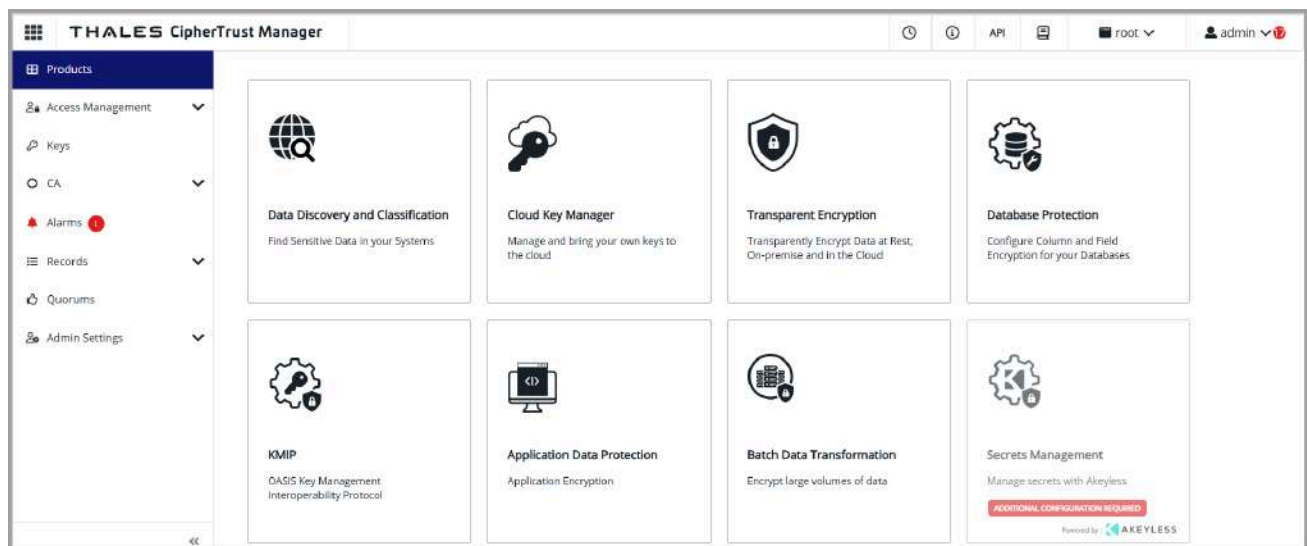
If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [Changing the Initial Password](#).

11. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

12. Using your new password, log in again. The **CipherTrust Manager** Web Page appears.



13. At this point, it's strongly recommended to configure an NTP server.

- a. Navigate to **Admin Settings > NTP**.

- b. Enter in an **NTP Server hostname**.
- c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
- d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

14. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
15. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                            | Required Private Key Format |
|-------------------------|---------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

16. Login as `ksadmin` to an SSH session or cloud web console.
17. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

Congratulations! You have successfully deployed a CipherTrust Manager instance.

### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

## SSH Access to the New Instance

If using a Windows client, use PuTTY or similar utility to SSH to your CipherTrust Manager instance as

---

KeySecure Administrator (ksadmin). If needed, use PuTTYgen or similar utility to format the SSH Key Pair.

If using a Linux client use SSH to login as KeySecure Administrator (ksadmin).

## Disk Resize Available with Disk Encryption

After you launch an unencrypted Virtual CipherTrust Manager and disable the cloud-init service, you cannot resize the disk. If you require a Virtual CipherTrust Manager with a larger or smaller disk size:

- [Encrypt the disk](#) for the Virtual CipherTrust Manager instance and then attempt re-size.
- [Take a backup](#) on the existing Virtual CipherTrust Manager, deploy a new Virtual CipherTrust Manager instance with the desired disk size, and restore the backup file on the new instance.

## Alibaba Cloud Deployment

You can deploy a CipherTrust Manager instance in the Alibaba Cloud.

### Minimum Requirements

To deploy a Virtual CipherTrust Manager instance, the following minimum requirements apply:

- System volume: 300 GB
- Memory: 16 GB
- vCPUs:
  - 2 for unclustered, standalone Virtual CipherTrust Managers
  - 4 for [clustered](#) Virtual CipherTrust Managers
- NICs: 1

#### 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 Alibaba Cloud

### Prerequisites

- CipherTrust Manager Image for Alibaba Cloud.

- If using a Windows client, use PuTTY or similar utility to SSH to your CipherTrust Manager instance as KeySecure Administrator (ksadmin).

If needed, use PuTTYgen or similar utility to format the SSH Key Pair.

- If using a Linux client use SSH to login as KeySecure Administrator (ksadmin).

## To Launch a CipherTrust Manager instance

Steps for deploying a CipherTrust Manager instance in the Alibaba Cloud:

1. Log in to [Alibaba Cloud](#).
2. Click the **Menu** () icon on the top left corner, then select **Elastic Compute Service** from the left navigation pane.
3. Click **Create Instance**.  
The ECS Custom Launch page is displayed.
4. In the **Basic Configurations** section, specify/select the following settings:
  - a. Select the desired Billing Method.

### Note

To learn more about Billing Methods, [click here](#).

- b. Select **Region** in which the instance is to be launched.
  - c. In **Instance Type**, choose **Architecture** and **Category** as required.
  - d. In **Image**, select **Custom Image**. Select the CipherTrust Manager custom image from the drop-down list.
  - e. In **Storage**, specify disk capacity to minimum 100 GB.
5. In the **Networking** section, specify/select the following settings:
    - a. In **Network Type**, select **VPC**.

### Note

To learn more about VPC, [click here](#).



- 
- b. In **Public IP Address**, select/clear the **Assign Public IPv4 Address** checkbox.

If you select the checkbox, it makes this instance accessible through Internet. Otherwise, this instance is accessible through private network only.

- c. It is recommended to use a **Security Group** to monitor/control the network traffic.

**Note**

To learn more about Security Group, [click here](#).

- d. Add **Elastic Network Interface**.

**Note**

To learn more about Elastic Network Interface, [click here](#).

6. In the **System Configurations** section, specify/select the following settings:

- a. In **Logon Credentials**, select **Key Pair**. Key pair can be auto generated or you can import the public key.

**Note**

To learn more about Key Pair Credentials, [click here](#).

- b. Provide **Instance Name**.

- c. In **Advanced Settings**, provide **User Data** or cloud init scripts.

**Note**

To learn more about User Data, [click here](#).

7. In the **Preview** section, specify/select the following settings:

- a. Select the **Automatic Release** checkbox, if required, when Pay-As-You-Go billing method is opted.

b. Select the **ECS Terms of Service and Product Terms of Service** checkbox.

c. Click **Create Instance**.

The Alibaba Cloud Instance will be created. Use its public/private IP to access the CipherTrust Manager.

8. Browse to this IP address (in this example enter **https://147.139.41.245**). The **Log In** screen appears.

9. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



### Note

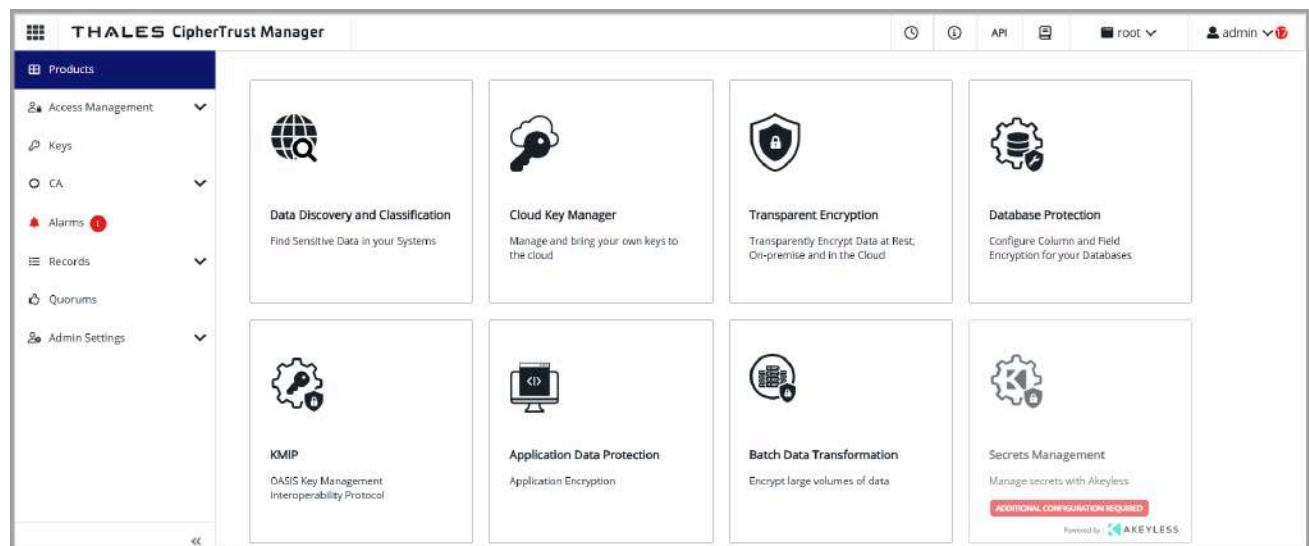
If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [changing the initial password](#).

10. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

11. Using your new password, log in again. The **CipherTrust Manager** Web Page appears.



12. At this point, it's strongly recommended to configure an NTP server.

- a. Navigate to **Admin Settings > NTP**.

- b. Enter in an **NTP Server hostname**.
- c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
- d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

13. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
14. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                                 | Required Private Key Format |
|-------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

15. Login as `ksadmin` to an SSH session or cloud web console.
16. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

## Disk Resize Available with Disk Encryption

After you launch an unencrypted Virtual CipherTrust Manager and disable the cloud-init service, you cannot resize the disk. If you require a Virtual CipherTrust Manager with a larger or smaller disk size:

- 
- **Encrypt the disk** for the Virtual CipherTrust Manager instance and then attempt re-size.
  - **Take a backup** on the existing Virtual CipherTrust Manager, deploy a new Virtual CipherTrust Manager instance with the desired disk size, and restore the backup file on the new instance.

## Microsoft Azure Deployment

You can deploy a CipherTrust Manager within the Microsoft Azure Cloud Computing Platform, including in the Azure Government environment (Arizona, Iowa, Virginia, and Texas regions).

## Minimum Requirements

To deploy a Virtual CipherTrust Manager instance, the following minimum requirements apply:

- System volume: 300 GB
- Memory: 16 GB
- vCPUs:
  - 2 for unclustered, standalone Virtual CipherTrust Managers
  - 4 for **clustered** Virtual CipherTrust Managers
- NICs: 1

### 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 CipherTrust Manager 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 CipherTrust Manager instance as KeySecure Administrator (ksadmin).

If needed, use PuTTYgen or similar utility to format the SSH Key Pair.

- If using a Linux client: use, SSH to login as KeySecure Administrator (ksadmin).

---

## To launch a Virtual CipherTrust Manager instance

1. Sign in to the Azure or Azure Government portal.
2. Search for **CipherTrust Manager** on the Marketplace page.

Only the [latest version](#) is available on the Marketplace page, as **CipherTrust Manager Community Edition**.

Older versions back to 2.6 are available through the [Azure CLI](#). Some versions older than 2.6 are available from Thales customer support as a VHD file.

## To deploy the latest Virtual CipherTrust Manager Version from Marketplace

1. Select the **Thales Virtual CipherTrust Manager** image from the Virtual Machines group.

### Note

The following steps apply to the Azure recommended 'Resources Manager' deployment model.

2. Click **Create**. The first screen of the **Create virtual machine** page is displayed.

# Create a virtual machine ...

 Changing Basic options may reset selections you have made. Review all options prior to creating the virtual machine.

## 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 * ⓘ | azure-sample-group ▼       |
|                    | <a href="#">Create new</a> |

## Instance details

|                          |                                                         |
|--------------------------|---------------------------------------------------------|
| Virtual machine name * ⓘ | kylo ✓                                                  |
| Region * ⓘ               | (US) West US ▼                                          |
| Availability options ⓘ   | No infrastructure redundancy required ▼                 |
| Image * ⓘ                | Virtual CipherTrust Manager k170v 2.3.0 (BYOL) - Gen1 ▼ |
|                          | <a href="#">See all images</a>                          |
| Azure Spot instance ⓘ    | <input type="checkbox"/>                                |
| Size * ⓘ                 | Standard_F4s - 4 vcpus, 8 GiB memory (\$181.77/month) ▼ |

3. Change the **Subscription** type if desired.
4. Select an existing **Resource group** or enter the name for a new one.

### Note

For deployments with Azure Dedicated HSMs, this must be the same resource group as for Luna client host VM and the Dedicated HSM.

5. Specify the **Region** of an Azure Datacenter. For example, **East US**.

### Note

For Azure government, this must be one of USGov Arizona, USGov Iowa, USGov Virginia, or USGov Texas.

6. Enter a **Virtual machine name**, which is the hostname for the virtual machine you are creating, for example, "mycompany-ciphertrust".
7. Select the **Size** for the VM that supports the [Minimum Requirements](#).
8. 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. RSA is the supported key algorithm. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security.

### Administrator account

Authentication type ⓘ

- ☒ SSH public key  
☐ Password

**i** 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 \* ⓘ

ksadmin ✓

SSH public key source

Generate new key pair ▼

Key pair name \*

test ✓

9. For the **Username**, enter "ksadmin", the default name for the System Administrator.

### Note

You **MUST** use the name "ksadmin" for this initial user.

10. 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. RSA is the supported key algorithm. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security.



### 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.

11. In the **SSH Public Key** field, provide the necessary information for the SSH public key source selected in the last step.
12. Select **Next: Disks >**. The **Disks** screen is displayed.

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 HDD (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.

Encryption type \*

(Default) Encryption at-rest with a platform-managed key

Enable Ultra Disk compatibility

☐

13. Change the **OS disk type** to "Standard HDD", unless you desire a faster disk.
14. Select an **Encryption type** depending on the [Azure-level disk encryption](#) you prefer.
15. Select **Next: Networking >**. The **Networking** screen is displayed.

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 *                  | <div>testVNET</div> <div>Create new</div>                                                                                                                                                       |
| Subnet *                           | <div>testSubnet (10.0.0.0/24)</div> <div>Manage subnet configuration</div>                                                                                                                      |
| Public IP                          | <div>(new) Tester-ip</div> <div>Create new</div>                                                                                                                                                |
| NIC network security group         | <div> <input type="radio"/> None<br/> <input type="radio"/> Basic<br/> <input checked="" type="radio"/> Advanced         </div> <div>  This VM image has preconfigured NSG rules         </div> |
| Configure network security group * | <div>(new) Tester-nsg</div> <div>Create new</div>                                                                                                                                               |
| Accelerated networking             | <div><input type="checkbox"/></div> <div>The selected image does not support accelerated networking.</div>                                                                                      |

### Load balancing

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. [Learn more](#)

Place this virtual machine behind an existing load balancing solution? ☐

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 deployments with Azure Dedicated HSMs, the virtual network must also include the Dedicated HSM instance, and the lunaclient host instance. The subnet must also include the lunaclient host instance; this subnet is referred to as the "Compute" subnet.

For a list of security groups/ports, refer to [Network Security Groups](#).

16. Select **Next: Management** >. The Management screen is displayed.
17. Select **Next: Advanced** >. The Advanced screen is displayed.
18. If desired, you can apply [cloud-init configuration](#). Paste cloud-init configuration in the **Custom Data** field.

## Extensions

Extensions provide post-deployment configuration and automation.

Extensions ⓘ

[Select an extension to install](#)

## Custom data

Pass a script, configuration file, or other data into the virtual machine **while it is being provisioned**. The data will be saved on the VM in a known location. [Learn more about custom data for VMs](#) ↗

Custom data

```
#cloud-config
keysecure:
 initial-user:
 generate-password: true
```



19. Select **Next: Tags >**. The **Tags** screen is displayed.

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                           |
|----------------------|---------|------------------------------------|
| <input type="text"/> | :       | <input type="text"/> 12 selected ▼ |

20. Enter any desired tags.

21. 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.

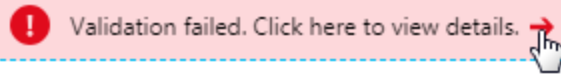
a. If the validation was not successful, a **Validation failed** message is displayed:

---

[Home](#) > [SafeNet KeySecure k170v \(BYOL\)](#) > [Create a virtual machine](#)

---

## 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:

[Home](#) > [SafeNet KeySecure k170v \(BYOL\)](#) > [Create a virtual machine](#)

## Create a virtual machine

✓ Validation passed

[Basics](#) [Disks](#) [Networking](#) [Management](#) [Guest config](#) [Tags](#) [Review + create](#)

### PRODUCT DETAILS

SafeNet Virtual KeySecure k170v

by Gemalto

[Terms of use](#) | [Privacy policy](#)

Not covered by credits ⓘ

0.0000 USD/hr

Standard A2

by Microsoft

[Terms of use](#) | [Privacy policy](#)

Subscription credits apply ⓘ

0.1200 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         | Free Trial                            |
| Resource group       | k170v-test                            |
| Virtual machine name | mycompany-keysecure                   |
| Region               | East US                               |
| Availability options | No infrastructure redundancy required |
| Authentication type  | SSH public key                        |
| Username             | ksadmin                               |

### DISKS

|                     |              |
|---------------------|--------------|
| OS disk type        | Standard HDD |
| Use unmanaged disks | No           |

### NETWORKING

Create

Previous

Next

[Download a template for automation](#)

22. Select **Create** to begin deployment of this VM. This screen is displayed indicating that deployment is in process.

Home > CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844 - Overview

### CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844 - Overview

Deployment

Search (Ctrl+/) « Delete Cancel Redeploy Refresh

**Overview**

Outputs

Inputs

Template

**... Your deployment is underway**

Check the status of your deployment, manage resources, or troubleshoot deployment issues. Pin this page to your dashboard to easily find it next time.

 Deployment name: CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844  
Subscription: [Free Trial](#)  
Resource group: [k170v-test](#)

DEPLOYMENT DETAILS [\(Download\)](#)

Start time: 03/10/2018, 09:33:37  
Duration: 0.55 seconds  
Correlation ID: f8df9e76-fa06-4662-9ae2-089a3986fa79

When deployment completes, this screen is displayed, providing access to all resources supporting the new VM.

Home > CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844 - Overview

### CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844 - Overview

Deployment

Search (Ctrl+/) « Delete Cancel Redeploy Refresh

**Overview**

Outputs

Inputs

Template

**✓ Your deployment is complete**

[Go to resource](#)

 Deployment name: CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844  
Subscription: [Free Trial](#)  
Resource group: [k170v-test](#)

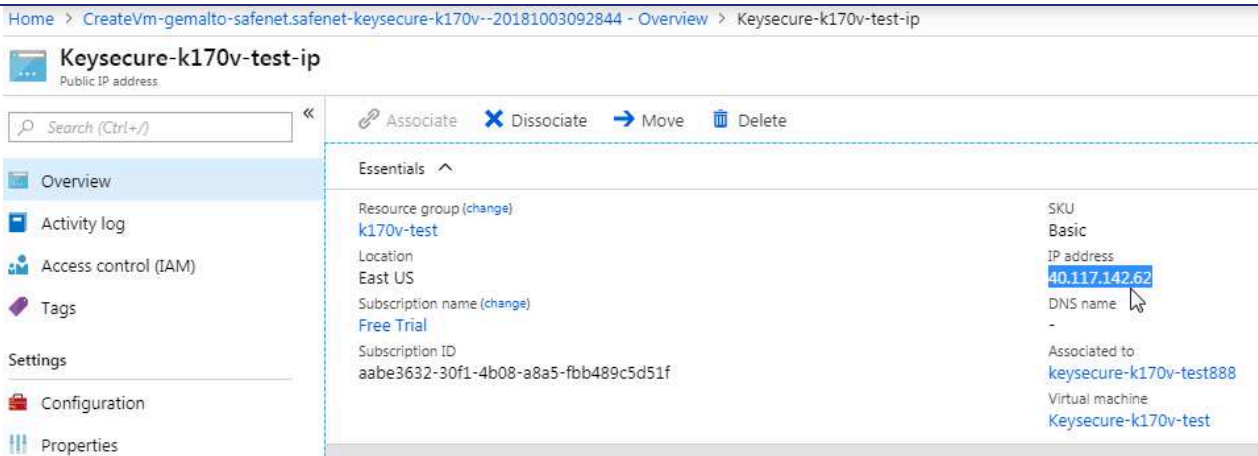
DEPLOYMENT DETAILS [\(Download\)](#)

Start time: 03/10/2018, 09:35:49  
Duration: 3 minutes 26 seconds  
Correlation ID: f8df9e76-fa06-4662-9ae2-089a3986fa79

| RESOURCE                                   | TYPE                           | STATUS  | OPERATION DETAILS                 |
|--------------------------------------------|--------------------------------|---------|-----------------------------------|
| ✓ <a href="#">Keysecure-k170v-test</a>     | Microsoft.Compute/virtualM...  | OK      | <a href="#">Operation details</a> |
| ✓ <a href="#">keysecure-k170v-test888</a>  | Microsoft.Network/networkI...  | Created | <a href="#">Operation details</a> |
| ✓ <a href="#">k170v-test-vnet</a>          | Microsoft.Network/virtualNe... | OK      | <a href="#">Operation details</a> |
| ✓ <a href="#">Keysecure-k170v-test-ip</a>  | Microsoft.Network/publicIp...  | OK      | <a href="#">Operation details</a> |
| ✓ <a href="#">Keysecure-k170v-test-nsg</a> | Microsoft.Network/network...   | OK      | <a href="#">Operation details</a> |
| ✓ <a href="#">k170vtestdiag</a>            | Microsoft.Storage/storageA...  | OK      | <a href="#">Operation details</a> |

23. Connect to the CipherTrust Manager Web Page.

- a. Select the resource with the IP address, in this example: **Keysecure-k170v-test-ip**.



- b. Browse to this IP address (in this example enter **https://40.117.142.62**). The **Log In** screen appears.



24. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



**Note**

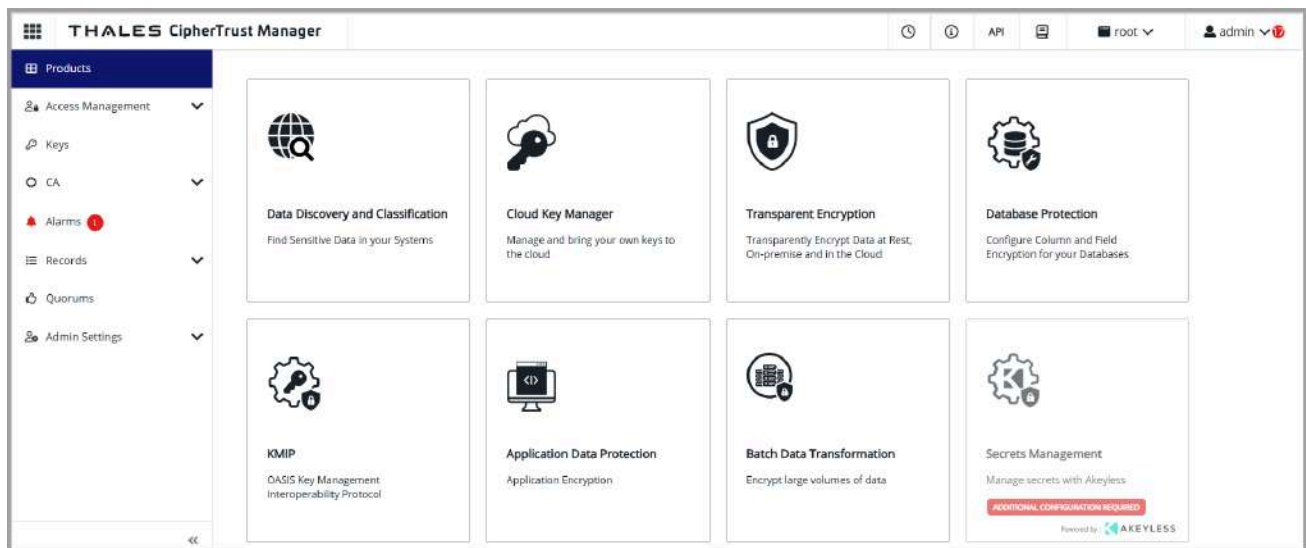
If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [changing the initial password](#).

25. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

26. Using your new password, log in again. The CipherTrust Manager Web Page appears.



27. At this point, it's strongly recommended to configure an NTP server.
- Navigate to **Admin Settings > NTP**.
  - Enter in an **NTP Server hostname**.
  - For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.



28. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
29. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                                 | Required Private Key Format |
|-------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

30. Login as `ksadmin` to an SSH session or cloud web console.
31. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

#### Note

Virtual CipherTrust Manager includes a 90 day trial license. To activate your instance with a term or perpetual license, see [Licensing](#).

## To deploy versions back to 2.6 using the Azure CLI

1. List the Virtual CipherTrust Manager images available through the Azure CLI. Find the URN value for your desired version.

```
az vm image list --offer cm_k170v --all
```

#### Example response:

```
[
 {
```

```

 "offer": "cm_k170v",
 "publisher": "thalesdiscplusainc1596561677238",
 "sku": "ciphertrust_manager",
 "urn": "thalesdiscplusainc1596561677238:cm_k170v:ciphertrust_manager:2.6.6
506",
 "version": "2.6.6506"
 },
 {
 "offer": "cm_k170v",
 "publisher": "thalesdiscplusainc1596561677238",
 "sku": "ciphertrust_manager",
 "urn": "thalesdiscplusainc1596561677238:cm_k170v:ciphertrust_manager:2.7.6
808",
 "version": "2.7.6808"
 }
]

```

2. Accept the terms for the desired Virtual CipherTrust Manager image version.

```
az vm image terms accept --urn <image_urn>
```

3. Create the VM image. You provide or specify the following values:

- The image URN.
- The name of an existing resource group. The image's region will be taken from this resource group.

#### Note

For deployments with Azure Dedicated HSMs, this must be the same resource group as for Luna client host VM and the Dedicated HSM.

#### Note

For Azure government, the region must be one of USGov Arizona, USGov Iowa, USGov Virginia, or USGov Texas.

- A new name for the VM.
- A desired size for the VM, that meets the [Minimum Requirements](#).

- The admin username set to `ksadmin`. This is required to supply an SSH key at launch time, and for SSH access. The only allowed value is `ksadmin`.
- A public IP SKU set to `Standard`
- A source for the SSH key. You can choose to generate a new SSH key pair (`--generate-ssh-keys`), enter a name for an existing key already stored in Azure (`--ssh-key-name <name_of_existing_ssh_key_in_azure>`), or upload an SSH key file (`--ssh-key-value <ssh_key_file>`). We support OpenSSH format for the public key, and OpenSSH, PKCS1, or PKCS8 format for the private key. RSA is the supported key algorithm. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security.

### 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.

- (Optional) If you wish to perform [cloud-init configuration](#) to set some initial values for CipherTrust Manager, you can use the `-d` or `--custom-data` flag to pass in the user data from `config.dat` file.

### Example syntax

```
az vm create --resource-group <resource_group> --name <desired_vm_name> --image <image_urn> --size <desired_image_size> --admin-username ksadmin {--generate-ssh-keys | --ssh-key-name <name_of_existing_ssh_key_in_azure> | --ssh-key-value <ssh_key_file>} [--custom-data <config.dat_file_path>] --public-ip-sku Standard
```

### Example command with Standard\_F4s\_V2 size and SSH key generation

```
az vm create --resource-group myURNVM --name myVM --image thalesdiscplusaincl596561677238:cm_k170v:ciphertrust_manager:2.6.6506 --admin-username ksadmin --size Standard_F4s_v2 --generate-ssh-keys --public-ip-sku Standard
```

### Example response

```
{
 "fqdns": "",
 "id": "/subscriptions/260ecbe7-777b-4d3c-84ea-887620498863/resourceGroups/myURNVM/providers/Microsoft.Compute/virtualMachines/myVM",
```

```

"location": "westus",
"macAddress": "00-00-00-00-00-00",
"powerState": "VM running",
"privateIpAddress": "1.1.1.1",
"publicIpAddress": "2.2.2.2",
"resourceGroup": "myURNVM",
"zones": ""
}

```

Note the public IP address returned in the VM creation response.

4. Open the 443 port to allow web browsing to the Virtual CipherTrust Manager instance.

```

az vm open-port -g <resource_group_name> -n <virtual_machine_name> --port '443'

```

5. Navigate to the `https://<VM_public_IP_address>` to connect to the CipherTrust Manager Web Page. The public IP address was returned when the VM was created. You can also view this value in the Azure portal **Virtual Machine** list.

The **Log In** screen appears.

this EULA'."/>

**THALES**

**Username**

**Password**

Log In

By clicking Log In, you are accepting [this EULA](#)

6. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



#### Note

If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [changing the initial password](#).

7. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

8. Using your new password, log in again. The CipherTrust Manager Web Page appears.
9. At this point, it's strongly recommended to configure an NTP server.
  - a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

10. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
11. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                                 | Required Private Key Format |
|-------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

12. Login as `ksadmin` to an SSH session or cloud web console.

13. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

#### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or term or perpetual license, see [Licensing](#).

## Disk Resize Available with Disk Encryption

After you launch an unencrypted Virtual CipherTrust Manager and disable the cloud-init service, you cannot resize the disk. If you require a Virtual CipherTrust Manager with a larger or smaller disk size:

- [Encrypt the disk](#) for the Virtual CipherTrust Manager instance and then attempt re-size.
- [Take a backup](#) on the existing Virtual CipherTrust Manager, deploy a new Virtual CipherTrust Manager instance with the desired disk size, and restore the backup file on the new instance.

## Microsoft Azure Stack Deployments

Azure Stack offers hybrid cloud platforms, which allows you to deploy Virtual CipherTrust Manager instances to be accessible to both private and public networks. You can deploy Virtual CipherTrust Manager on Azure Local (previously known as Azure Stack hyperconverged architecture (HCI)) or Azure Stack Hub.

## Prerequisites

- The Virtual CipherTrust Manager VHDX image file available from Thales customer support portal.

- A Microsoft account with Azure Local or Azure Stack Hub software.
- For Azure Local Deployments, a configured cluster is necessary.

## Minimum Requirements

To deploy a Virtual CipherTrust Manager instance, the following minimum requirements apply:

- System volume: 300 GB
- Memory: 16 GB
- vCPUs:
  - 2 for unclustered, standalone Virtual CipherTrust Managers
  - 4 for **clustered** Virtual CipherTrust Managers
- NICs: 1

### 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.

Where applicable, we provide some recommended starting values in the deployment steps.

## Azure Local Deployment

1. Download and install the [Windows Desktop Client](#).
2. In the Desktop Client main page, select **Subscribe** and enter your Microsoft account name.
3. Select **Jump Box** to start the remote connection. Enter your Microsoft account credentials when prompted.
4. In a browser, login to the Windows Admin Center with your Microsoft account.
5. On the Windows Admin Center home screen, under **All connections**, select the desired cluster.
6. Navigate to **Volumes > Files & file sharing** and upload the Virtual CipherTrust Manager VDHX file.
7. From the home screen, navigate to **Tools > Virtual machines**.
8. Select the **Inventory** tab, select **Add** and **New**.
9. Provide the necessary values for the new virtual machine (VM).

- 
- a. Enter a **Name**.
  - b. Ensure **Generation 1** is selected for the **Generation**.
  - c. Under **Host**, select the desired host server.
  - d. Under **Path**, choose a folder on the Jump Box file system to save the VM configuration and virtual hard disk files to.
  - e. Under **Virtual processors**, select the number of virtual processors. We recommend 4 processors.  
  
Virtual CipherTrust Managers launch in Community Edition, a trial mode which only allows up to four processors. You must purchase and apply a k470 [license](#) to use more than four processors.
  - f. Under **Memory**, select the amount of startup memory, and a minimum and maximum range of dynamic memory. We recommend 4 GB as a minimum for startup memory.
  - g. Under **Network**, select a virtual switch.
  - h. Under **Storage**, click **Add** and **Browse**. Navigate to the Virtual CipherTrust Manager VDHX file, and select it.
10. Select **Create** to create the VM.
  11. Start the VM. In the **Virtual Machines** list, hover over the new VM, enable the checkbox for it on the left, and select **Start**. Watch the **State** column to verify that the VM is **Running**.
  12. Proceed to [connect to the launched Virtual CipherTrust Manager](#).

## Azure Stack Hub Deployment

1. In a browser, login to Azure Stack Hub portal with your Microsoft account.
2. Select **Storage Account**. Choose an appropriate storage account.
3. Under **Blob service**, select **Containers**. Create a new container or select an existing one.
4. Select **Upload**.

The **Upload blob** box opens on the right side of the browser.

5. Navigate to the Virtual CipherTrust Manager VHDX file on your local system, select it, and click **Upload**.
6. Search for "images" to navigate to the **Images** page.
7. Select **Add**.
8. Provide the necessary values for the new image:
  - a. Select a **Subscription**.



- 
- b. Select a **Resource group**.
  - c. Enter a **Name**.
  - d. Select a **Region**.
  - e. For the **OS Type**, select **Linux**.
  - f. For **Storage blob**, browse in **Storage accounts > Containers** to the Virtual CipherTrust Manager VDHX file, and select it.
  - g. For the **Account Type** select **Standard HDD**.
9. Select **Review + Create**, and then **Create**.
  10. Once the deployment is complete, select **Go to resource**.
  11. Select **+ Create VM**.

**Note**

If you wish to use cloud-init to apply settings before launch, you must use REST, PowerShell, or Azure CLI to deploy the VM, as described in [Azure Stack Hub documentation](#). A cloud-init deployment is recommended to encrypt Virtual CipherTrust Manager instances **be encrypted at launch time** to ensure secrets are never exposed.

12. Provide the necessary values for the new virtual machine (VM), keeping the default values for other fields:
  - a. Provide a **Name**
  - b. Select a **Size**. We recommend 4 CPUs and 16 GB RAM.
  - c. For **OS disk type**, select **Standard HDD**.
  - d. Under **Administrator account**, keep the Authentication type as **SSH Public Key**. Change the Username to `ksadmin`, and paste in an SSH key.

**Note**

We support OpenSSH format for the public key, and OpenSSH, PKCS1, or PKCS8 format for the private key. The supported key algorithm is RSA. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security. You can generate this key using 'PuTTYgen' or similar utility. Save this SSH Public Key at a safe location. You will need this key for future SSH access.

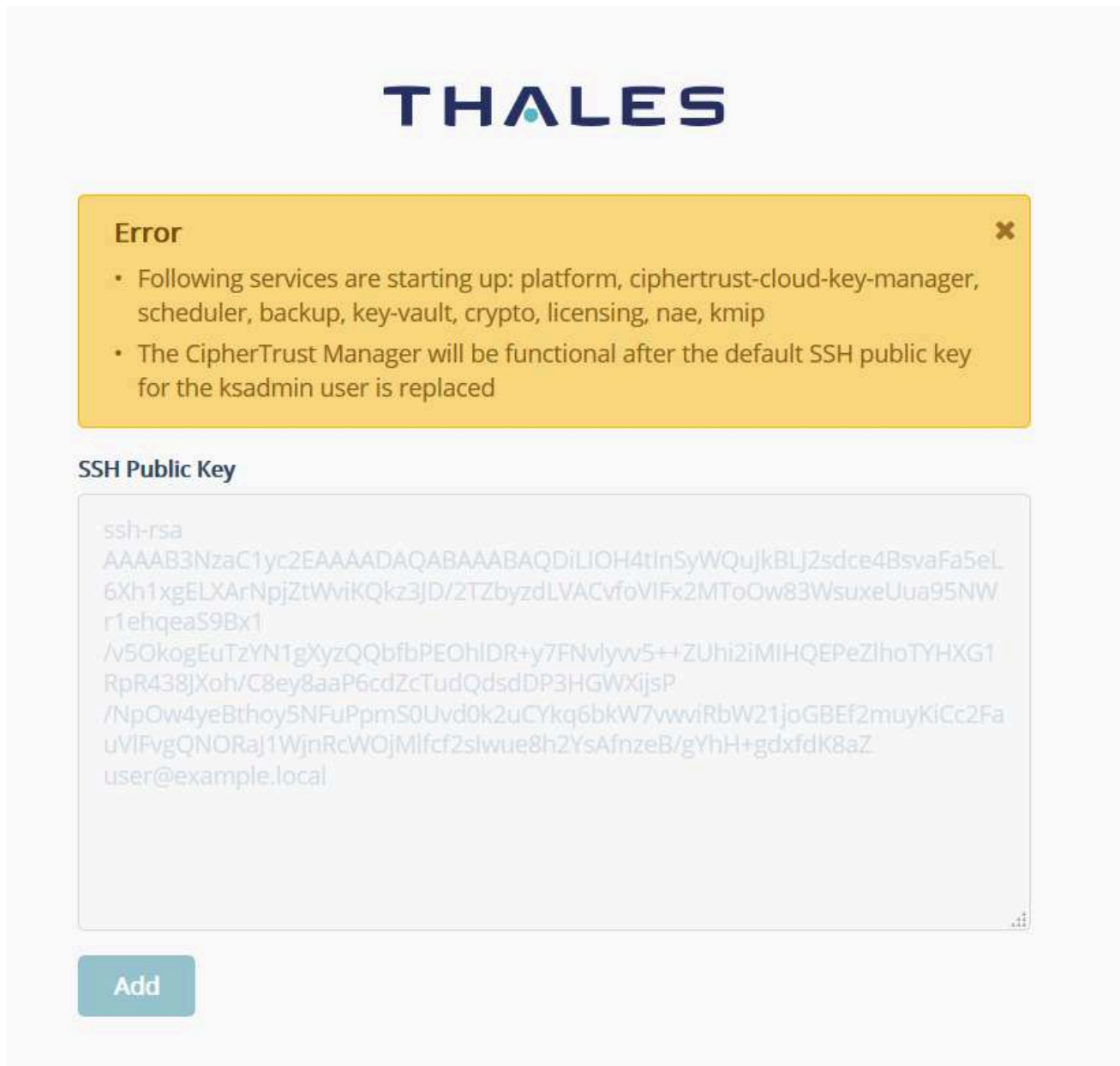
e. For **Select inbound ports**, enable **HTTP (80)**, **HTTPS (443)**, and **SSH(22)**.

13. Select **Review + Create**, and then **Create**.

14. Proceed to [connect to the launched Virtual CipherTrust Manager](#).

## Connect to the Launched Virtual CipherTrust Manager

1. In the Azure Local or Azure Stack Hub interface, use the **Connect** option to navigate to the launched VM. You can see the CipherTrust Manager web console GUI, and the IP/Hostname.
2. If you have not provided an SSH key before launching, you are prompted to replace the default **SSH Public Key**.



**THALES**

**Error** ✕

- Following services are starting up: platform, ciphertrust-cloud-key-manager, scheduler, backup, key-vault, crypto, licensing, nae, kmip
- The CipherTrust Manager will be functional after the default SSH public key for the ksadmin user is replaced

**SSH Public Key**

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQADiL1OH4tInSyWQuJkBLJ2sdce4BsvaFa5eL
6Xh1xgELXArNpjZtWwiKQkz3JD/2TZbyzdLVACvfoVIFx2MToOw83WsuxeUua95NW
r1ehqeaS9Bx1
/v5OkogEuTzYN1gXyzQQbfbPEOHIDR+y7FNvlyw5++ZUhi2iMIHQEPeZlhoTYHXG1
RpR438JXoh/C8ey8aaP6cdZcTudQdsdDP3HGWXjsP
/NpOw4yeBthoy5NFuPpmS0Uvd0k2uCYkq6bkW7vwviRbW21joGBEf2muyKiCc2Fa
uVIFvgQNORaj1WjnRcWOjMlfcf2slwue8h2YsAfnzeB/gYhH+gdxfdK8aZ
user@example.local
```

**Add**

As the initial KeySecure admin (ksadmin) you must paste in your **SSH Public Key** in the box provided and then select **Add**.

#### Note

We support OpenSSH format for the public key, and OpenSSH, PKCS1, or PKCS8 format for the private key. The supported key algorithm is RSA. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security. You can generate this key using 'PuTTYgen' or similar utility. Save this SSH Public Key at a safe location. You will need this key for future SSH access.

After replacing the default SSH Public Key, the Log In screen appears. For more options to replace the default SSH Public Key, see [Starting Services After Deployment](#).

3. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



#### Password Change

After successfully changing your password, you will be logged out and will have to log in again

#### Note

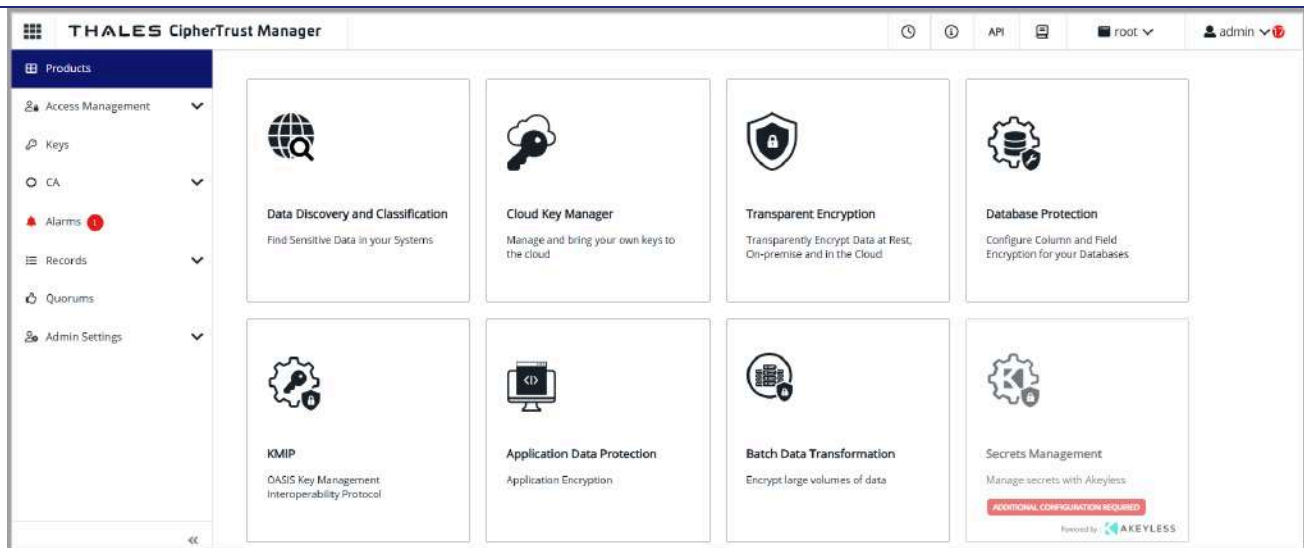
If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [Changing the Initial Password](#).

4. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

5. Using your new password, log in again. The **CipherTrust Manager** Web Page appears.



6. At this point, it's strongly recommended to configure an NTP server.
  - a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

7. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
8. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                            | Required Private Key Format |
|-------------------------|---------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

9. Login as `ksadmin` to an SSH session or cloud web console.

10. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

Congratulations! You have successfully deployed your Virtual CipherTrust Manager.

11. If you did not apply disk encryption with cloud-init, it is available [after first launch with ksctl](#). Because installation specific secrets are generated the first time a Virtual CipherTrust Manager instance is launched, it is recommended that the instance be encrypted at launch time to ensure these secrets are never exposed.

#### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

## Disk Resize Available with Disk Encryption

After you launch an unencrypted Virtual CipherTrust Manager and disable the cloud-init service, you cannot resize the disk. If you require a Virtual CipherTrust Manager with a larger or smaller disk size:

- [Encrypt the disk](#) for the Virtual CipherTrust Manager instance and then attempt re-size.
- [Take a backup](#) on the existing Virtual CipherTrust Manager, deploy a new Virtual CipherTrust Manager instance with the desired disk size, and restore the backup file on the new instance.

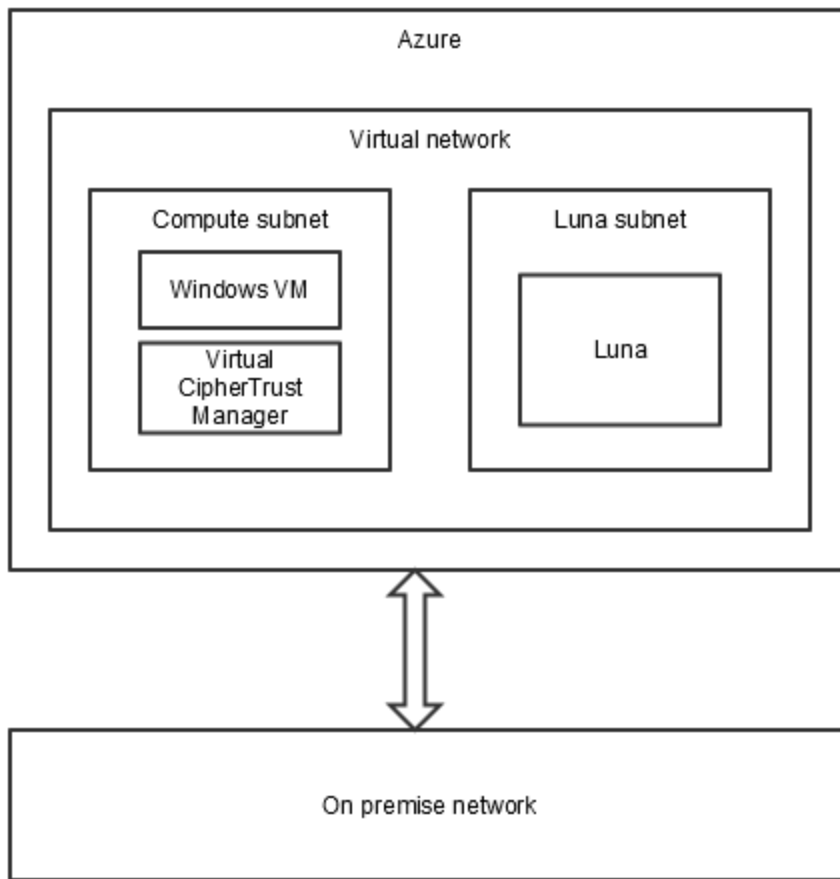
## Microsoft Azure Dedicated HSM Provisioning

Microsoft Azure provides a [dedicated Hardware Security Module \(HSM\) service](#) which can be used as a root of trust to a Virtual CipherTrust Manager which is also hosted on Microsoft Azure.

## Architecture Overview

To integrate together Virtual CipherTrust Manager and a Dedicated HSM on the Microsoft Azure cloud, you must create and configure the following entities, as shown in the diagram:

1. Necessary Azure resources, including a virtual network and two subnets within the virtual network.
2. A dedicated HSM instance within its own subnet, called "Luna" in the diagram.
3. A Virtual CipherTrust Manager instance in another subnet, called "Compute" in the diagram.
4. A Windows VM in the "Compute" subnet to host the Luna Client, which can access and configure an HSM partition.



## Acquire Permissions to a Dedicated HSM

1. Contact the Microsoft HSM account [HSMrequest@Microsoft.com](mailto:HSMrequest@Microsoft.com) to request access to a dedicated HSM. Outline your application, the regions you would like HSMs and the volume of HSMs you are looking for. If you work with a Microsoft representative, such as an Account Executive or Cloud Solution Architect for example, then include them in any request.

Once Microsoft establishes your requirement, they email you a list of documents and schema files that are necessary to deploy a Luna.

2. Check if permissions have been granted in the Azure console by running:

```
az feature show --namespace Microsoft.HardwareSecurityModules --name AzureDedicatedHSM
```

The expected result is that the returned "state" is set to "Registered", as shown in the following example:

```
{
 "id": "/subscriptions/d44a44b4-f4ce-4bb4-b4bc-4444444c444a/providers/Microsoft.HardwareSecurityModules/locations/eastus/providers/Microsoft.HardwareSecurityModules/registered"
}
```

```
ft.Features/providers/Microsoft.HardwareSecurityModules/features/AzureDedicate
dHsm",
 "name": "Microsoft.HardwareSecurityModules/AzureDedicatedHsm",
 "properties": {
 "state": "Registered"
 },
 "type": "Microsoft.Features/providers/features"
}
```

## Create the Microsoft Azure Resources

We need to create resources on the on Microsoft Azure before provisioning the Luna HSM.

We will be creating the following resources:

1. A resource group to place all the resources
2. A virtual network (vnet) that will be used for network communications.
3. A subnet called "Compute", to later contain the Windows VM lunaclient host and Virtual CipherTrust Manager instances.
4. A subnet dedicated to the HSM
5. A gateway

Refer to [Microsoft Azure documentation](#) for more advanced configuration options.

### To create a resource group

1. Open the left sidebar.
2. Hover over **Resource groups**.

A popup opens to the right.

3. Click on the **+ Create** icon.

This opens a new page to create the resource group

4. Enter the **Subscription name** and the **Resource group** name.
5. Select an appropriate region from the **Region** dropdown.
6. Click on **Review + create**
7. Once the validation passes, click on the **Create** button to create the resource group.

### To create a virtual network (vnet)

1. On your resource group page click on the button **Create resources**.

- 
2. Click on the **Networking** button.
  3. Click on **Virtual network**.
  4. Select a **Subscription** and **Resource group**.
  5. Select a **Name** and **Region** for the vnet instance.
  6. Click on **Next: IP Addresses >**

## To create a Compute subnet

1. Click on the **default** subnet name and rename it to the "Compute" subnet.
2. Click on **Save**.

## To create a subnet for the HSM

1. Click on the **+ Add subnet** button.
2. Provide a name for the HSM subnet.
3. Provide a subnet address range and mask.
4. Click on **Add**.
5. Once the subnet is added, click on **Next: Security >**.
6. Leave the default option and click on **Next: Tags >**.
7. Create an optional tag.
8. Click on **Next: Review + create >**.
9. Once the validation passes, click on the button **Create** to create the virtual network and the subnets.

## To delegate the HSM subnet to the dedicated HSM

1. Navigate to the **Subnets** section.
2. Select the HSM subnet.  
  
A panel to the right opens up
3. In the right panel, under **SUBNET DELEGATION**, click on **Delegate subnet to a service**.
4. Select Microsoft.HardwareSecurityModules/dedicatedHSMs
5. Click **OK**.



## To create a gateway

1. Navigate to the **Subnets** section.
2. Click on the **+ Gateway subnet**.
3. Choose a **Name** and **Subnet address range**.
4. Click on **OK**.

## Deploy the Dedicated Luna HSM in the Subnet

After you have delegated the HSM to the subnet, you have to deploy the HSM.

1. Copy the example text into a `Deploy-1HSM-toVNET-Params-v1.1.json` file. Edit the file to provide your previously configured values for the vnet and HSM subnet.

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json",
 "contentVersion": "1.0.0.0",
 "parameters": {
 "ResourcePrefix": {
 "value": "HSMLuna"
 },
 "ExistingVNetName": {
 "value": "Luna_kylo_vnet"
 },
 "ExistingHSMSubnetName": {
 "value": "Luna_kylo_HSM_subnet"
 },
 "HSMResourceName": {
 "value": "kylo-luna-hsm"
 }
 }
}
```

2. Copy the provided `Deploy-1HSM-toVNET-Template-v1.1.json` file and do not edit it:

[Deploy-1HSM-toVNET-Template-v1.1.json](#)

3. Open an Azure powershell terminal.
4. Click on the Upload/Download button on the terminal
5. Upload the two files.

- Run the following command on the powershell terminal to deploy the dedicated Luna HSM:

```
New-AzResourceGroupDeployment -ResourceGroupName Luna_kylo_rg -TemplateFile
.\Deploy-1HSM-toVNET-Template-v1.1.json -TemplateParameterFile .\Deploy-1HSM-t
oVNET-Params-v1.1.json -Name HSMdeploy -Verbose
```

It takes approximately 30 minutes to deploy. You receive a `ProvisioningState: Succeeded` when complete

- Verify the deployment by running the following commands:

```
PS> $subid = (Get-AzContext).Subscription.Id
PS> $resourceGroupName = "Luna_kylo_rg"
PS> $resourceName = "kylo-luna-hsm"
PS > Get-AzResource -Resourceid /subscriptions/$subId/resourceGroups/$resource
GroupName/providers/Microsoft.HardwareSecurityModules/dedicatedHSMs/$resourceN
ame
```

The expected result looks like:

```
Name : kylo-luna-hsm
ResourceGroupName : Luna_kylo_rg
ResourceType : Microsoft.HardwareSecurityModules/dedicatedHSMs
Location : westus2
ResourceId : /subscriptions/d44a44b4-f4ce-4bb4-b4bc-4444444c444a/resour
ceGroups/Luna_kylo_rg/providers/Microsoft.HardwareSecurityModules/dedicatedHSM
s/kylo-luna-hsm
Tags :
 Name Value
 =====
 Environment prod
 resourceType Hsm
```

## Create a Windows instance

A windows instance is necessary to access the Dedicated Luna HSM. From the windows instance, you can SSH into the HSM to perform general HSM configuration, and you can install the Luna client to configure an HSM partition.

Consult [Azure Documentation](#) for more detail on provisioning this resource if necessary.

- Create a Windows 10 Pro instance on the Compute subnet with a Standard HDD disk. Be sure to use the **Resource group** you created for the Windows VM, Dedicated HSM, and Virtual CipherTrust Manager.
- Get the public IP address of the Windows instance and use it with the Remote Desktop Client (RDP) to

---

access the Windows instance.

## Configure the Dedicated Luna HSM

You must perform some configuration on the Dedicated Luna HSM instance to ensure its passwords and policies are secure enough, and to create a partition to act as a root of trust for Virtual CipherTrust Manager.

Consult [Luna Network HSM 7.2 Documentation](#) for more information on these commands, if necessary.

### To configure the HSM

1. To find the HSM's IP address, go to the Azure portal, and click on `HSMLuna-ergwpip`.
2. Use the IP address to SSH to the HSM from your Windows VM. The default user name is **tenantadmin** and the default password is **PASSWORD**.

You are brought into the Luna Shell (LunaSH) utility.

```
PS > ssh tenantadmin@172.22.1.4
The authenticity of host '172.22.1.4 (172.22.1.4)' can't be established.
ECDSA key fingerprint is SHA256:vf96hinpthrML150b4jNGFlaf1oJT0+QQjdfz/B0+yQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.22.1.4' (ECDSA) to the list of known hosts.
tenantadmin@172.22.1.4's password:
Last login: Wed Oct 7 19:28:44 2020

Luna Network HSM Command Line Shell v7.2.0-220. Copyright (c) 2018 SafeNet. All
rights reserved.
```

3. Change your password immediately, as prompted.
4. Run `network show` to confirm the IP address for **eth0**.

#### Warning

Do not change the IP address on eth0. The access to the Luna would be lost.

5. Run `hsm show` to view the current state of the HSM. This command also displays firmware version, software version, and FIPS (Federal Information Process Standards) operating mode.
6. Initialize the HSM with a label. Take care to remember your password.

```
lunash:>hsm init -label LunaHSM -password mystrongpassword -domain domain1
```

---

`hsm show` now shows the HSM label.

7. Run `hsm login` to login to the HSM. Use the password set in the previous step.
8. If your organization requires FIPS compliance, you need to set the HSM to `FIPS 140-2 Operation` mode.
  - a. To change the FIPS mode and restrict non-FIPS algorithms, set the policy 12 to value 0.

`lunash:>hsm changePolicy -policy 12 -value 0`

### Warning

Setting the HSM to the FIPS mode zeroizes the HSM. All the data on the HSM is lost.

- b. Run `hsm show` to confirm the `FIPS 140-2 Operation` mode is on.
9. Create the partition using `partition create`. Provide a name and a slot number.

```
lunash:>partition create -p luna_part -s 10000
```

10. Run `partition list` to verify the partition was created. You can also get more details about the partition using `partition show -p <partition_name>`

## Access and Configure the HSM Partition

Consult [Luna Network HSM 7.2 Documentation](#) for more information on lunaCM commands and partition roles, if necessary.

1. Obtain the Luna Client (version 7.4 or higher) from the [Thales Customer Support Portal](#).
2. Transfer the client package to the Windows instance.
3. Install the client with all options and features.
4. Open powershell in Administrator mode and b to the location where Lunaclient is installed.
5. Run `luna.exe`. No HSMs are currently registered with the client.
6. Use `clientconfig deploy` to register the HSM with the Luna client and to secure the connection. Provide the HSM's IP address, user and password, and the partition name.

```
lunacm:> clientconfig deploy -server <HSM_IP> -client <Windows_VM_IP> -partiti
on <partition_name> -user tenantadmin -password <password> -verbose
```

```
Please wait while we set up the connection to the HSM. This may take several minutes...
```

7. Initialize the partition, providing a label. You are prompted for a password for the Partition Security Officer, and a domain name.

```
partition init -label <partition_name>
```

8. Login as the Partition Officer (PO) and initialize the Crypto Officer (CO) with a temporary password.

```
lunacm:> role login -name po

 enter password: *****

lunacm:> role init -name co

 enter new password: *****

 re-enter new password: *****
```

9. Login as the Crypto Officer and change the temporary password.

```
lunacm:> role login -name co

 enter password: *****

lunacm:> role changepw -name co

 enter existing password: *****

 enter new password: *****

 re-enter new password: *****
```

10. To verify the partition is active, return to the Luna Shell on the Dedicated HSM instance, and run `partition show -p <partition_name>`.

## Deploy the Virtual CipherTrust Manager on Azure

1. Sign in to the Azure or Azure Government portal.
2. Search for **CipherTrust Manager** on the Marketplace page.

Only the **latest version** is available on the Marketplace page, as **CipherTrust Manager Community Edition**.

Older versions back to 2.6 are available through the [Azure CLI](#). Some versions older than 2.6 are available from Thales customer support as a VHD file.

## To deploy the latest Virtual CipherTrust Manager Version from Marketplace

1. Select the **Thales Virtual CipherTrust Manager** image from the Virtual Machines group.

### Note

The following steps apply to the Azure recommended 'Resources Manager' deployment model.

2. Click **Create**. The first screen of the **Create virtual machine** page is displayed.

## Create a virtual machine ...

 Changing Basic options may reset selections you have made. Review all options prior to creating the virtual machine.

### 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 \* ⓘ

azure-sample-group

[Create new](#)

### Instance details

Virtual machine name \* ⓘ

kylo

Region \* ⓘ

(US) West US

Availability options ⓘ

No infrastructure redundancy required

Image \* ⓘ

Virtual CipherTrust Manager k170v 2.3.0 (BYOL) - Gen1

[See all images](#)

Azure Spot instance ⓘ

☐

Size \* ⓘ

Standard\_F4s - 4 vcpus, 8 GiB memory (\$181.77/month)

3. Change the **Subscription** type if desired.
4. Select an existing **Resource group** or enter the name for a new one.

**Note**

For deployments with Azure Dedicated HSMs, this must be the same resource group as for Luna client host VM and the Dedicated HSM.

5. Specify the **Region** of an Azure Datacenter. For example, **East US**.

**Note**

For Azure government, this must be one of USGov Arizona, USGov Iowa, USGov Virginia, or USGov Texas.

6. Enter a **Virtual machine name**, which is the hostname for the virtual machine your are creating, for example, "mycompany-ciphertrust".
7. Select the **Size** for the VM that supports the [Minimum Requirements](#).
8. 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. RSA is the supported key algorithm. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security.

**Administrator account**

Authentication type ⓘ

- ☒ SSH public key
- ☐ Password

**i** 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 \* ⓘ

ksadmin ✓

SSH public key source

Generate new key pair ▼

Key pair name \*

test ✓

9. For the **Username**, enter "ksadmin", the default name for the System Administrator.

**Note**

You MUST use the name "ksadmin" for this initial user.

10. 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. RSA is the supported key algorithm. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security.

**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.

11. In the **SSH Public Key** field, provide the necessary information for the SSH public key source selected in the last step.
12. Select **Next: Disks >**. The **Disks** screen is displayed.



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 HDD (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.

Encryption type \*

(Default) Encryption at-rest with a platform-managed key ▼

Enable Ultra Disk compatibility ⓘ

☐

13. Change the **OS disk type** to "Standard HDD", unless you desire a faster disk.
14. Select an **Encryption type** depending on the [Azure-level disk encryption](#) you prefer.
15. Select **Next: Networking >**. The **Networking** screen is displayed.

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 \* ⓘ testVNET   
[Create new](#)

Subnet \* ⓘ testSubnet (10.0.0.0/24)   
[Manage subnet configuration](#)

Public IP ⓘ (new) Tester-ip   
[Create new](#)

NIC network security group ⓘ   
☐ None  
☐ Basic  
☒ Advanced

 This VM image has preconfigured NSG rules

Configure network security group \* (new) Tester-nsg   
[Create new](#)

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](#)

Place this virtual machine behind an existing load balancing solution? ☐

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 deployments with Azure Dedicated HSMs, the virtual network must also include the Dedicated HSM instance, and the lunaclient host instance. The subnet must also include the lunaclient host instance; this subnet is referred to as the "Compute" subnet.

For a list of security groups/ports, refer to [Network Security Groups](#).

16. Select **Next: Management** >. The Management screen is displayed.
17. Select **Next: Advanced** > The Advanced screen is displayed.
18. If desired, you can apply [cloud-init configuration](#). Paste cloud-init configuration in the **Custom Data** field.

## Extensions

Extensions provide post-deployment configuration and automation.

Extensions ⓘ

[Select an extension to install](#)

## Custom data

Pass a script, configuration file, or other data into the virtual machine **while it is being provisioned**. The data will be saved on the VM in a known location. [Learn more about custom data for VMs](#) ↗

Custom data

```
#cloud-config
keysecure:
 initial-user:
 generate-password: true
```



19. Select **Next: Tags >**. The **Tags** screen is displayed.

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                           |
|----------------------|---------|------------------------------------|
| <input type="text"/> | :       | <input type="text"/> 12 selected ▼ |

20. Enter any desired tags.

21. 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.

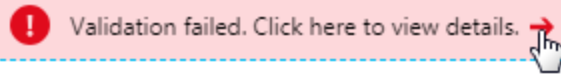
a. If the validation was not successful, a **Validation failed** message is displayed:

---

[Home](#) > [SafeNet KeySecure k170v \(BYOL\)](#) > [Create a virtual machine](#)

---

## 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:

[Home](#) > [SafeNet KeySecure k170v \(BYOL\)](#) > [Create a virtual machine](#)

## Create a virtual machine

✓ Validation passed

[Basics](#) [Disks](#) [Networking](#) [Management](#) [Guest config](#) [Tags](#) [Review + create](#)

### PRODUCT DETAILS

SafeNet Virtual KeySecure k170v

by Gemalto

[Terms of use](#) | [Privacy policy](#)

Not covered by credits ⓘ

0.0000 USD/hr

Standard A2

by Microsoft

[Terms of use](#) | [Privacy policy](#)

Subscription credits apply ⓘ

0.1200 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         | Free Trial                            |
| Resource group       | k170v-test                            |
| Virtual machine name | mycompany-keysecure                   |
| Region               | East US                               |
| Availability options | No infrastructure redundancy required |
| Authentication type  | SSH public key                        |
| Username             | ksadmin                               |

### DISKS

|                     |              |
|---------------------|--------------|
| OS disk type        | Standard HDD |
| Use unmanaged disks | No           |

### NETWORKING

Create

Previous

Next

[Download a template for automation](#)

22. Select **Create** to begin deployment of this VM. This screen is displayed indicating that deployment is in process.

Home > CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844 - Overview

### CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844 - Overview

Deployment

Search (Ctrl+/) « Delete Cancel Redeploy Refresh

**Overview**

Outputs

Inputs

Template

**... Your deployment is underway**

Check the status of your deployment, manage resources, or troubleshoot deployment issues. Pin this page to your dashboard to easily find it next time.

 Deployment name: CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844  
Subscription: [Free Trial](#)  
Resource group: [k170v-test](#)

DEPLOYMENT DETAILS [\(Download\)](#)

Start time: 03/10/2018, 09:33:37  
Duration: 0.55 seconds  
Correlation ID: f8df9e76-fa06-4662-9ae2-089a3986fa79

When deployment completes, this screen is displayed, providing access to all resources supporting the new VM.

Home > CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844 - Overview

### CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844 - Overview

Deployment

Search (Ctrl+/) « Delete Cancel Redeploy Refresh

**Overview**

Outputs

Inputs

Template

**✓ Your deployment is complete**

[Go to resource](#)

 Deployment name: CreateVm-gemalto-safenet.safenet-keysecure-k170v--20181003092844  
Subscription: [Free Trial](#)  
Resource group: [k170v-test](#)

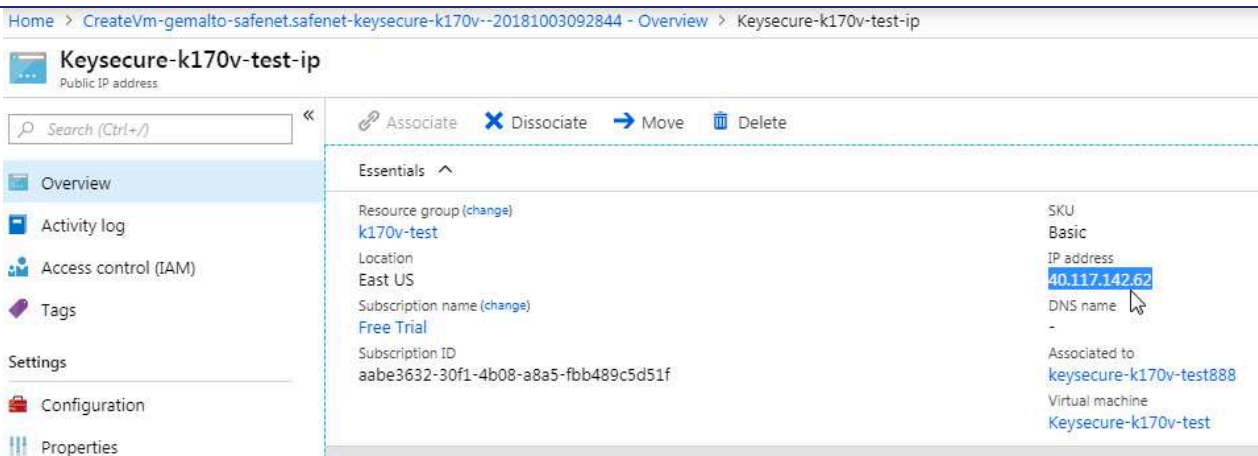
DEPLOYMENT DETAILS [\(Download\)](#)

Start time: 03/10/2018, 09:35:49  
Duration: 3 minutes 26 seconds  
Correlation ID: f8df9e76-fa06-4662-9ae2-089a3986fa79

| RESOURCE                                   | TYPE                           | STATUS  | OPERATION DETAILS                 |
|--------------------------------------------|--------------------------------|---------|-----------------------------------|
| ✓ <a href="#">Keysecure-k170v-test</a>     | Microsoft.Compute/virtualM...  | OK      | <a href="#">Operation details</a> |
| ✓ <a href="#">keysecure-k170v-test888</a>  | Microsoft.Network/networkI...  | Created | <a href="#">Operation details</a> |
| ✓ <a href="#">k170v-test-vnet</a>          | Microsoft.Network/virtualNe... | OK      | <a href="#">Operation details</a> |
| ✓ <a href="#">Keysecure-k170v-test-ip</a>  | Microsoft.Network/publicIp...  | OK      | <a href="#">Operation details</a> |
| ✓ <a href="#">Keysecure-k170v-test-nsg</a> | Microsoft.Network/network...   | OK      | <a href="#">Operation details</a> |
| ✓ <a href="#">k170vtestdiag</a>            | Microsoft.Storage/storageA...  | OK      | <a href="#">Operation details</a> |

23. Connect to the CipherTrust Manager Web Page.

- a. Select the resource with the IP address, in this example: **Keysecure-k170v-test-ip**.



- b. Browse to this IP address (in this example enter **https://40.117.142.62**). The **Log In** screen appears.



24. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



**Note**

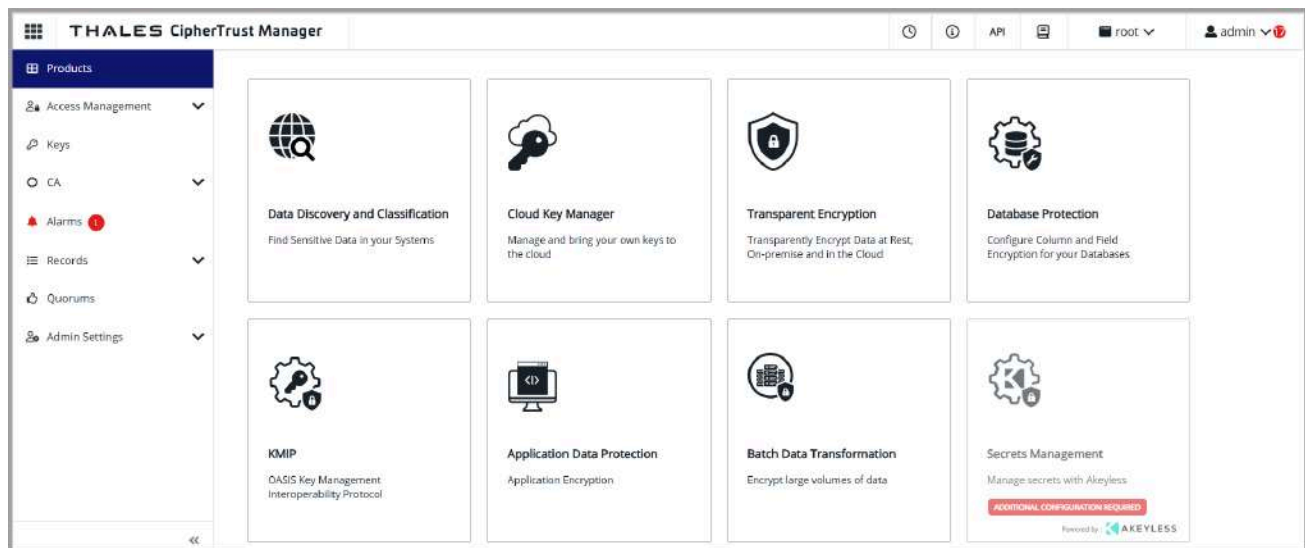
If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [changing the initial password](#).

25. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

26. Using your new password, log in again. The CipherTrust Manager Web Page appears.



27. At this point, it's strongly recommended to configure an NTP server.
- Navigate to **Admin Settings > NTP**.
  - Enter in an **NTP Server hostname**.
  - For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.



28. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
29. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                                 | Required Private Key Format |
|-------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

30. Login as `ksadmin` to an SSH session or cloud web console.
31. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

### Note

Virtual CipherTrust Manager includes a 90 day trial license. To activate your instance with a term or perpetual license, see [Licensing](#).

## To deploy versions back to 2.6 using the Azure CLI

1. List the Virtual CipherTrust Manager images available through the Azure CLI. Find the URN value for your desired version.

```
az vm image list --offer cm_k170v --all
```

### Example response:

```
[
 {
```

```

 "offer": "cm_k170v",
 "publisher": "thalesdiscplusainc1596561677238",
 "sku": "ciphertrust_manager",
 "urn": "thalesdiscplusainc1596561677238:cm_k170v:ciphertrust_manager:2.6.6
506",
 "version": "2.6.6506"
 },
 {
 "offer": "cm_k170v",
 "publisher": "thalesdiscplusainc1596561677238",
 "sku": "ciphertrust_manager",
 "urn": "thalesdiscplusainc1596561677238:cm_k170v:ciphertrust_manager:2.7.6
808",
 "version": "2.7.6808"
 }
]

```

2. Accept the terms for the desired Virtual CipherTrust Manager image version.

```
az vm image terms accept --urn <image_urn>
```

3. Create the VM image. You provide or specify the following values:

- The image URN.
- The name of an existing resource group. The image's region will be taken from this resource group.

#### Note

For deployments with Azure Dedicated HSMs, this must be the same resource group as for Luna client host VM and the Dedicated HSM.

#### Note

For Azure government, the region must be one of USGov Arizona, USGov Iowa, USGov Virginia, or USGov Texas.

- A new name for the VM.
- A desired size for the VM, that meets the [Minimum Requirements](#).

- The admin username set to `ksadmin`. This is required to supply an SSH key at launch time, and for SSH access. The only allowed value is `ksadmin`.
- A public IP SKU set to `Standard`
- A source for the SSH key. You can choose to generate a new SSH key pair (`--generate-ssh-keys`), enter a name for an existing key already stored in Azure (`--ssh-key-name <name_of_existing_ssh_key_in_azure>`), or upload an SSH key file (`--ssh-key-value <ssh_key_file>`). We support OpenSSH format for the public key, and OpenSSH, PKCS1, or PKCS8 format for the private key. RSA is the supported key algorithm. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security.

### 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.

- (Optional) If you wish to perform [cloud-init configuration](#) to set some initial values for CipherTrust Manager, you can use the `-d` or `--custom-data` flag to pass in the user data from `config.dat` file.

### Example syntax

```
az vm create --resource-group <resource_group> --name <desired_vm_name> --image <image_urn> --size <desired_image_size> --admin-username ksadmin {--generate-ssh-keys | --ssh-key-name <name_of_existing_ssh_key_in_azure> | --ssh-key-value <ssh_key_file>} [--custom-data <config.dat_file_path>] --public-ip-sku Standard
```

### Example command with Standard\_F4s\_V2 size and SSH key generation

```
az vm create --resource-group myURNVM --name myVM --image thalesdiscplusaincl596561677238:cm_k170v:ciphertrust_manager:2.6.6506 --admin-username ksadmin --size Standard_F4s_v2 --generate-ssh-keys --public-ip-sku Standard
```

### Example response

```
{
 "fqdns": "",
 "id": "/subscriptions/260ecbe7-777b-4d3c-84ea-887620498863/resourceGroups/myURNVM/providers/Microsoft.Compute/virtualMachines/myVM",
```

```

"location": "westus",
"macAddress": "00-00-00-00-00-00",
"powerState": "VM running",
"privateIpAddress": "1.1.1.1",
"publicIpAddress": "2.2.2.2",
"resourceGroup": "myURNVM",
"zones": ""
}

```

Note the public IP address returned in the VM creation response.

4. Open the 443 port to allow web browsing to the Virtual CipherTrust Manager instance.

```

az vm open-port -g <resource_group_name> -n <virtual_machine_name> --port '443'

```

5. Navigate to the `https://<VM_public_IP_address>` to connect to the CipherTrust Manager Web Page. The public IP address was returned when the VM was created. You can also view this value in the Azure portal **Virtual Machine** list.

The **Log In** screen appears.



The image shows the Thales CipherTrust Manager login interface. At the top is the Thales logo. Below it is a login form with two input fields: 'Username' and 'Password'. The 'Username' field has a placeholder text 'username or connection/username'. The 'Password' field has a placeholder text 'password'. Below the password field is a teal 'Log In' button. At the bottom of the form, there is a text line: 'By clicking Log In, you are accepting [this EULA](#)'.

6. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



#### Note

If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [changing the initial password](#).

7. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

8. Using your new password, log in again. The CipherTrust Manager Web Page appears.
9. At this point, it's strongly recommended to configure an NTP server.
  - a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

10. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
11. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                                 | Required Private Key Format |
|-------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

12. Login as `ksadmin` to an SSH session or cloud web console.

13. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

#### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or term or perpetual license, see [Licensing](#).

## Configuring the HSM Partition as the Root of Trust for Virtual CipherTrust Manager

To configure the HSM Partition as a root of trust for the CipherTrust Manager, you must first register the Virtual CipherTrust Manager as a client to the HSM partition. Then, you must provide some HSM partition authentication values and certificates to the Virtual CipherTrust Manager to establish a connection.

### To register the Virtual CipherTrust Manager as a client to the HSM

1. RDP to the Windows client.
2. Download a version 7.2 Luna client from the [Support Portal](#). Copy the client installer into the Windows remote desktop.
3. Fetch the `server.pem` certificate from the Dedicated HSM instance with SCP.

```
scp.exe tenantadmin@<dedicated_HSM_IP>:server.pem .
tenantadmin@<dedicated_HSM_IP>'s password:
server.pem 100% 1159 1.1KB/s
```

00:00

4. Use the VTL tool to create a client certificate and key for the Virtual CipherTrust Manager instance.

```
./vtl createcert -n <Virtual_CipherTrust_IP_address> -d 10 -v
```

The client certificate and key are available in the cert/client folder

5. Use `partition showinfo` on LunaCM to view the partition name and serial number.
6. Copy the client certificate to the HSM with SCP.

```
PS C:\Users\LunaAdmin\Desktop> scp client.pem tenantadmin@<dedicated_HSM_IP>:
tenantadmin@<dedicated_HSM_IP>'s password:

172.22.0.5.pem 100% 1184 1.2KB/s 0
0:00
```

7. SSH into the Dedicated HSM.
8. With the LunaSH utility, register the client to the HSM, setting a client name and providing the Virtual CipherTrust Manager IP address.

```
lunash:>client register -client <client_name> -ip <Virtual_CipherTrust_IP_addr
ess>
```

9. Assign the HSM partition to the client.

```
lunash:>client assignPartition -client <client_name> -partition <partition_nam
e>
```

10. Run `client show -c <client_name>` to verify that the partition has been assigned to the client.

## To add the HSM connection to the Virtual

1. Login to the Virtual CipherTrust Manager.
2. Navigate to **System> HSMs**.
3. Select **SafeNet Luna SA 5, 6, or 7 - Network HSM**
4. Enter the following values:

- Partition label

- 
- Partition Password: This is the Crypto Officer password created [above](#)
  - Partition Serial Number,
  - IP of the Dedicated HSM
  - HSM server certificate,
  - Client certificate,
  - Client certificate key.

5. Click **Next** to be taken to a validation page.

If the validation is successful the server will restart and you will be prompted to login again.

6. Login with the default credentials (username: admin , password: admin).
7. Change the password and login.
8. Verify that the HSM has been added on **System> HSMs**.

## Disk Resize Available with Disk Encryption

After you launch an unencrypted Virtual CipherTrust Manager and disable the cloud-init service, you cannot resize the disk. If you require a Virtual CipherTrust Manager with a larger or smaller disk size:

- [Encrypt the disk](#) for the Virtual CipherTrust Manager instance and then attempt re-size.
- [Take a backup](#) on the existing Virtual CipherTrust Manager, deploy a new Virtual CipherTrust Manager instance with the desired disk size, and restore the backup file on the new instance.

## Oracle Cloud Deployment

### Minimum Requirements

To deploy a Virtual CipherTrust Manager instance, the following minimum requirements apply:

- System volume: 300 GB
- Memory: 16 GB
- vCPUs:
  - 2 for unclustered, standalone Virtual CipherTrust Managers
  - 4 for [clustered](#) Virtual CipherTrust Managers
- NICs: 1



**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 Oracle Cloud

Steps for deploying a CipherTrust Manager instance in Oracle Cloud:

1. Contact [Thales customer support](#) to request a virtual CipherTrust Manager image for Oracle Cloud.

Customer support will provide an **Object Storage URL** needed to import the image.

2. Log in to [Oracle Cloud](#).
3. Click menu button on top left corner, then navigate to **Core Infrastructure > Compute > Custom Images**.
4. Under **List Scope**, select the desired **Compartment** to import the image into.
5. Click **Import Image**.

The **Import Image** dialog opens.

6. Provide details to import the Virtual CipherTrust Manager image.
  - a. Set a meaningful **Name**.
  - b. Set the **Operating System** to **Ubuntu**.
  - c. Select **Import from an Object Storage URL** and provide the **Object Storage URL** obtained in step 1.
  - d. Set the **Image Type** to **QCOW2**.
  - e. Set the **Launch Mode** to **Paravirtualized mode**.
  - f. Click **Import Image**.

7. On the **Custom Image** details page, find the **Work Requests** resource, and the **Create image** operation. Wait for the listed **State** to show as Succeeded and the listed **% Complete** to reach 100%.

At this point, the **Create Instance** button becomes enabled.

8. Click **Create Instance** to launch a Virtual CipherTrust Manager from the imported image.

The **Create Compute Image** page displays.

**Note**

Do not alter default settings except as indicated in the below steps. For example, **Boot Volume** and **Live Migration** settings should be left as default.

9. In the **Placement** section, select the desired **Availability Domain**.

**Note**

To learn more about Availability Domains, [click here](#).

10. Select the desired **Shape**.

**Note**

To learn more about Compute Shapes, [click here](#).

11. In the **Networking** section, specify/select the following settings:

- Virtual Cloud Network Compartment
- Select a Virtual Cloud Network
- Subnet Compartment
- Subnet
- Use Network Security Groups to control traffic (*optional*)

**Note**

It is recommended to use a Network Security Group to monitor/control the network traffic. To learn more about Network Security Groups, [click here](#).

- Depending on your requirements, select one of the following:
  - Assign a Public IP Address

---

To make this instance accessible through internet.

- Do Not Assign a Public IP Address

To make this instance accessible through private network only.

12. In the **Add SSH Keys** section, use one of the following options:

#### Note

We support OpenSSH format for the public key, and OpenSSH, PKCS1, or PKCS8 format for the private key. RSA is the supported key algorithm. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security.

- Generate SSH Keys

Use this option to generate the SSH keys on-the-go. Then download the keys using the links provided.

- Use SSH Key Files

If you already have SSH key files, then select this option and then upload the public key file.

- Paste SSH Keys

If you already have SSH key files, then paste the contents of the public key file in the field provided.

#### Note

You can add additional keys, if required. To learn how to generate a SSH Key pair, click [here](#).

#### Tip

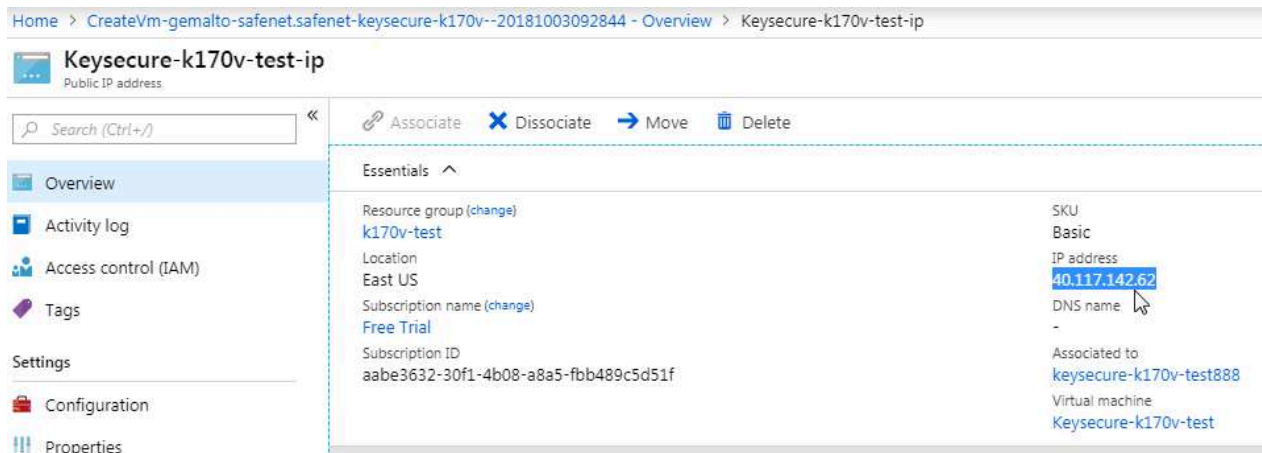
If it is desired to specify custom **Cloud-Init configuration**, then click **Show Advanced Options**, go to **Management** tab, and paste your Cloud-Init configuration script.

13. Click Create.

*The Oracle Cloud Instance will be created. Use its public/private IP to access CipherTrust Manager.*

14. Connect to the **CipherTrust Manager** Web Page.

- a. Select the resource with the IP address, in this example: **Keysecure-k170v-test-ip**.



- b. Browse to this IP address (in this example enter **https://40.117.142.62**). The **Log In** screen appears.



15. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



**Note**

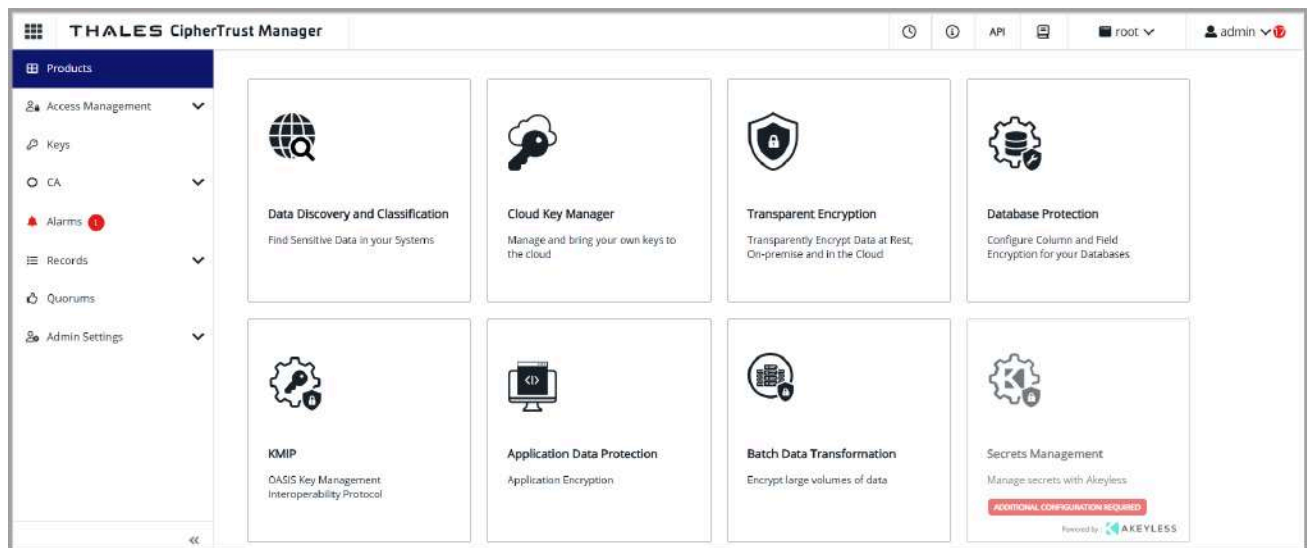
If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [changing the initial password](#).

16. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

17. Using your new password, log in again. The **CipherTrust Manager Web Page** appears.



18. At this point, it's strongly recommended to configure an NTP server.
  - a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

19. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
20. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                                 | Required Private Key Format |
|-------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

21. Login as `ksadmin` to an SSH session or cloud web console.
22. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

## Disk Resize Available with Disk Encryption

After you launch an unencrypted Virtual CipherTrust Manager and disable the cloud-init service, you cannot resize the disk. If you require a Virtual CipherTrust Manager with a larger or smaller disk size:

- [Encrypt the disk](#) for the Virtual CipherTrust Manager instance and then attempt re-size.
- [Take a backup](#) on the existing Virtual CipherTrust Manager, deploy a new Virtual CipherTrust Manager instance with the desired disk size, and restore the backup file on the new instance.

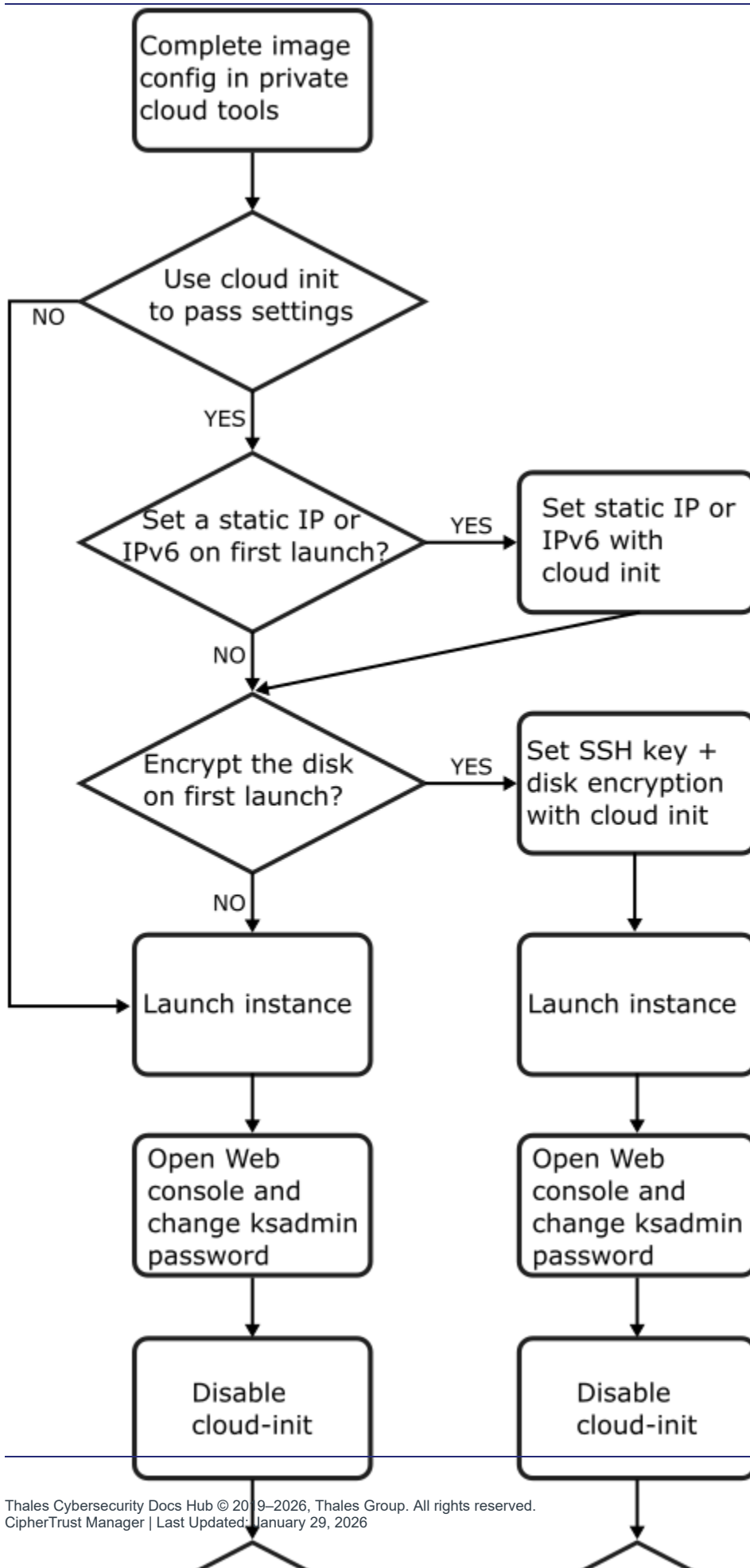
---

## Private Cloud Deployment

You can deploy a Virtual CipherTrust Manager instance in a private cloud. Support is currently provided for VMware vSphere, Microsoft Hyper-V, Nutanix AHV, and OpenStack, with file downloads available on the [customer support portal](#).

As part of private cloud deployment, you make decisions on disk encryption, setting a static IP, and disabling IPv6. For each of these options, you can apply them before launch, after launch, or not at all. In addition, you can use `nmcli` after first launch to set static routes, bond network interfaces, disable IPv6, or configure VLAN, if needed for your network.

The following flowchart demonstrates the decisions you make during virtual private cloud deployment.





## Minimum Requirements

To deploy a Virtual CipherTrust Manager instance, the following minimum requirements apply:

- System volume: 300 GB
- Memory: 16 GB
- vCPUs:
  - 2 for unclustered, standalone Virtual CipherTrust Managers
  - 4 for **clustered** Virtual CipherTrust Managers
- NICs: 1

### 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 CipherTrust Manager on VMware vSphere.

Refer to the *VMware vSphere documentation* for general information on launching a VM.

### Prerequisite

- OVA file: If your vSphere Client does not support deployment of OVA files directly, see [Decompressing an OVA file](#).

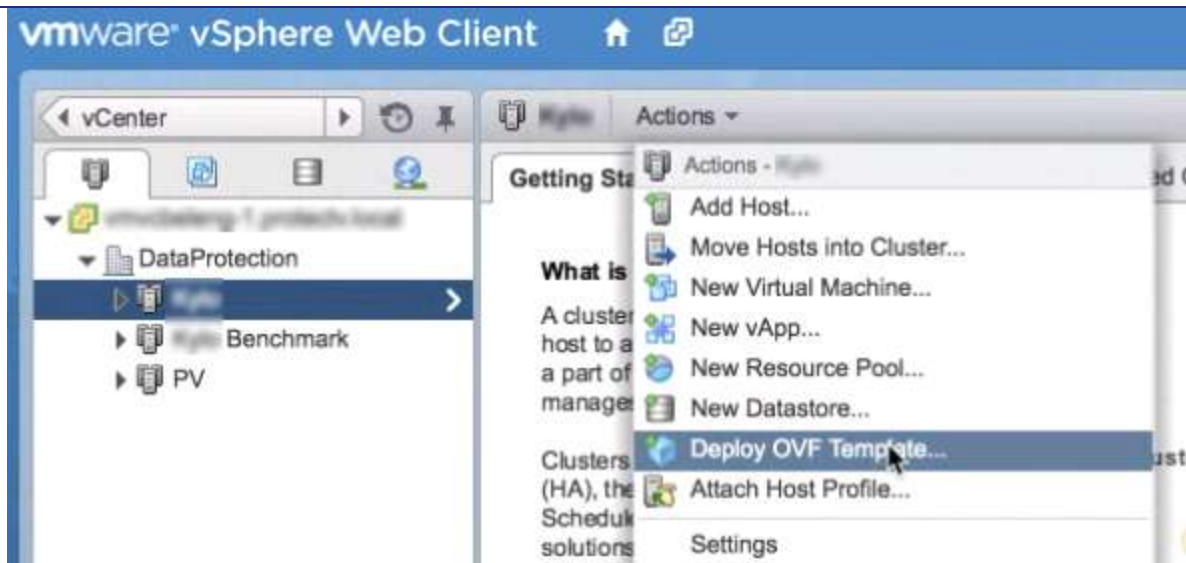
#### To launch the Virtual CipherTrust Manager

Virtual CipherTrust Manager supports VMware vSphere/ESXi v5.1 and higher versions. To launch Virtual CipherTrust on vSphere/ESXi v5.0 or lower versions, you need to change the virtual machine's hardware version from vmx-09 to vmx-08 or vmx-07. Use the VMware OVF Tool to make this change.

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 CipherTrust Manager appears under the inventory folder in the left pane.
3. In the left pane, select the newly launched Virtual CipherTrust Manager.
4. In the middle pane, select the **Summary** tab and click on the **Web console**.

The window opens in your browser. IP information is displayed along with the **ciphertrust login:** prompt.

#### Note

During the initial boot-up, the console may not display any IP information, or it might show outdated data before the system has collected the latest IP and other details.

To force a refresh and display the most current information, press **Enter** a few times on the console, or simply input a username and press **Ctrl+C** at the password prompt. This action will reload the console and update the displayed details.

5. At the **ciphertrust login:** prompt, enter "ksadmin" to log in and follow the prompts to create a secure password.

#### Note

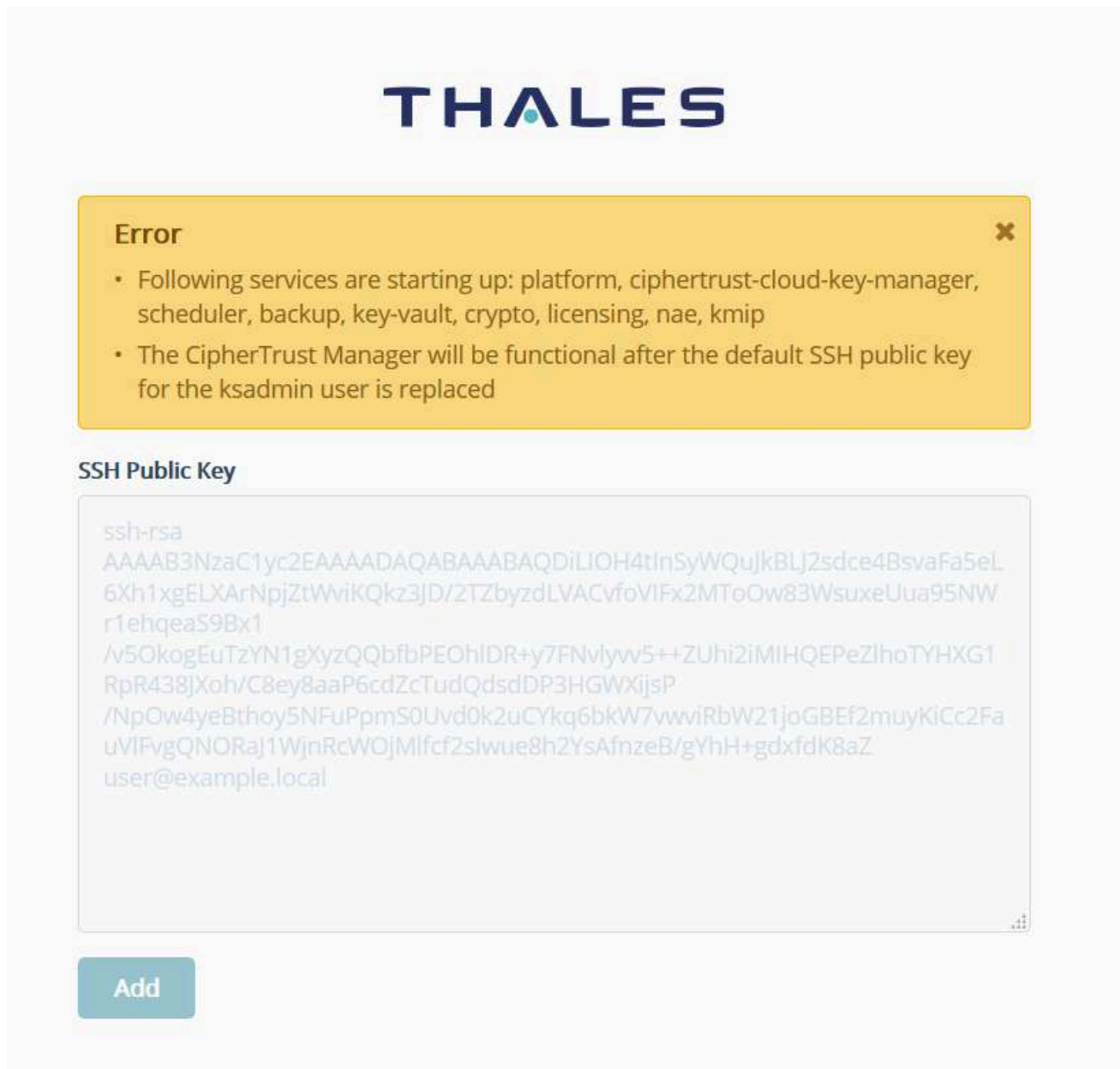
- If an IP address is not displayed, you likely do not have DHCP configured and you are not using Cloud-init configuration. If you do not have access to a static IP for the CipherTrust Manager GUI,

use the `nmcli` utility to configure a static IP address.

- `nmcli` can also be used to set static routes, bond network interfaces, disable IPv6, or configure VLAN, if needed for your network.

6. Access the GUI by browsing to the Virtual CipherTrust Manager's IP address in a web browser.

The initial **CipherTrust Manager** Web Page screen is displayed:



7. The Error displayed is normal and simply requires the default **SSH Public Key** to be replaced. As the initial KeySecure admin (ksadmin) you must paste in your **SSH Public Key** in the box provided and then select

---

**Add.****Note**

We support OpenSSH format for the public key, and OpenSSH, PKCS1, or PKCS8 format for the private key. The supported key algorithm is RSA. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security. You can generate this key using 'PuTTYgen' or similar utility. Save this SSH Public Key at a safe location. You will need this key for future SSH access.

After replacing the default SSH Public Key, the Log In screen appears. For more options to replace the default SSH Public Key, see [Starting Services After Deployment](#).

8. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:

**Password Change**

After successfully changing your password, you will be logged out and will have to log in again

**Note**

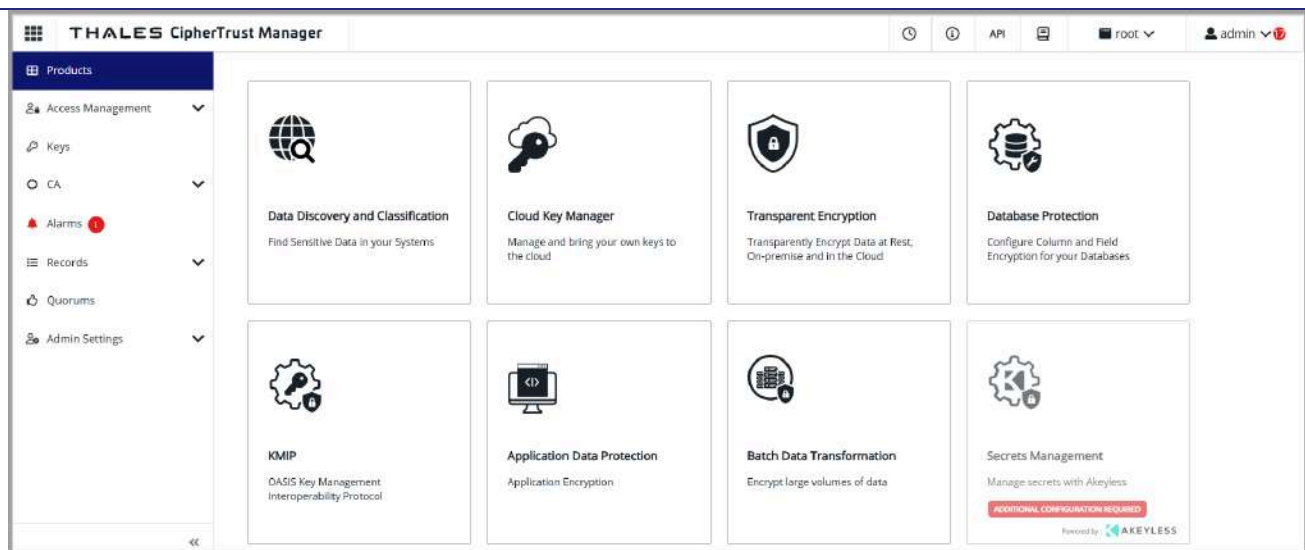
If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [Changing the Initial Password](#).

9. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

10. Using your new password, log in again. The **CipherTrust Manager** Web Page appears.



11. At this point, it's strongly recommended to configure an NTP server.
  - a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

Congratulations! You have successfully deployed your Virtual CipherTrust Manager.

12. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
13. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                            | Required Private Key Format |
|-------------------------|---------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

14. Login as `ksadmin` to an SSH session or cloud web console.
15. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

16. If you did not apply disk encryption with cloud-init, it is available [after first launch with ksctl](#). Because installation specific secrets are generated the first time a Virtual CipherTrust Manager instance is launched, it is recommended that the instance be encrypted at launch time to ensure these secrets are never exposed.

### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

## Decompressing an OVA File

The Virtual CipherTrust Manager package includes an Open Virtual Appliance (OVA) file for launching Virtual CipherTrust Manager on VMware vSphere. Some versions of the vSphere Client do not support deployment of OVA files directly. In this case it is recommended to extract the `.ovf`, `.vmdk`, and `.mf` files using the VMware OVF Tool. You can then launch Virtual CipherTrust Managers using these files.

**Example: To decompress the `.ovf` file, execute this operation:**

```
ovftool.exe --lax <source_OVA_file> <destination_OVF_file>
```

`<source_OVA_file>`: represents the OVA file included in the Virtual CipherTrust Manager package.

`<destination_OVF_file>`: represents a name for the OVF file.

## Using Cloud-init with VMware vSphere

Virtual CipherTrust Manager uses Cloud-init to inject configuration information prior to first boot. Cloud-init is a standard configuration mechanism that is supported in vSphere.

Below is an example of setting a static IP address using Cloud-init and vSphere.

### To setup a static IP address using Cloud-init

1. Create your Cloud-init configuration file. This is a text file containing the specific configuration you want to use. Refer to ["Plan Configuration Settings for Deployment with Cloud Init"](#) for other available configuration parameters. Here is a simple configuration file for a static IP:

```
#cloud-config
keysecure:
```

```
netcfg:
 iface:
 name: ens32
 type: static
 address: 10.121.107.103
 netmask: 255.255.252.0
 gateway: 10.121.104.1
 dns1: 172.16.2.12
```

The configuration parameters that save deployment steps after first launch are disk encryption, static IP, and disabling IPv6. Because installation specific secrets are generated the first time a Virtual CipherTrust Manager instance is launched, it is recommended that the instance be encrypted at launch time to ensure these secrets are never exposed.

2. VMware requires this file to be converted to base64. To convert to base64, use the `openssl` command:

```
openssl base64 -in <infile> -out <outfile>
```

The above file converted to base64 is below:

```
I2Nsb3VklWNvbmZpZw0Ka2V5c2VjdXJlOg0KICBuZXRjZmc6DQogICAgZWZhY2U6DQogICAgICBuYW
1lOiBlOGgwDQogICAgICB0eXBldGF0aWMNCiAgICAgIGFkZlJlc3M6IDEwLjEyMS4xMDcuMTAz
DQogICAgICBuZXRtYXNrOiAyNTUuMjU1LjI1Mi4wDQogICAgICBnYXRld2F5OjE6MC4xMjEuMTA0Lj
ENCiAgICAgIGRuczE6IDE3Mi4xNi4yLjEy
```

3. Create a Virtual CipherTrust Manager in VMware, but do not boot it.
4. Configure the VM with the following:

#### Note

The vSphere web client should be used since some configuration parameters are not available with the Windows client.

- a. Add an empty CD/DVD drive to the VM, no configuration of it is necessary.
  - b. Under **vApp Options > OVF Settings > OVF Environment Transport**, select ISO Image.
  - c. Under **vApp Options > Properties** create a property with the Key "user-data", and a default value of the base64 encoded file.
5. Boot up your system.

**Note**

Due to a bug, some versions of Virtual CipherTrust Manager may not apply the static IP on the initial boot. After a reboot, the IP address will be applied correctly.

- If you did not apply disk encryption with cloud-init, it is available [after first launch with ksctl](#). Because installation specific secrets are generated the first time a Virtual CipherTrust Manager instance is launched, it is recommended that the instance be encrypted at launch time to ensure these secrets are never exposed.

## Deploying in Hyper-V

The CipherTrust Manager Private Cloud Image supports Microsoft Hyper-V in the form of a Virtual Hard Disk (VHDX) image. This image has been tested as a Generation 1 Hyper-V virtual machine.

Refer to the *Microsoft Hyper-V Manager* documentation for general information on launching a VM using an existing disk on Hyper-V.

This section guides you through the steps needed to deploy a Virtual CipherTrust Manager on Microsoft Hyper-V.

### Prerequisites

- Virtual CipherTrust Manager Virtual Hard Disk (VHDX) image.

#### To decompress the zipped image file you received from Thales Sales

- Locate the image file you received from Thales sales; e.g. **k170v-1.x.x.xxxx.vhdx.zip**.
- Decompress this file by right-clicking on the file and selecting **Extract All...**

This is a large file and will take some time. A single decompressed file will result, which is a Hard Disk Image File; e.g. **k170v-1.x.x.xxxx.vhdx**.

- Note the location of this file as you will need it below when creating the virtual machine.

#### Launch the Hyper-V Manager on your Windows System

- If needed, install Hyper-V on your Windows system.

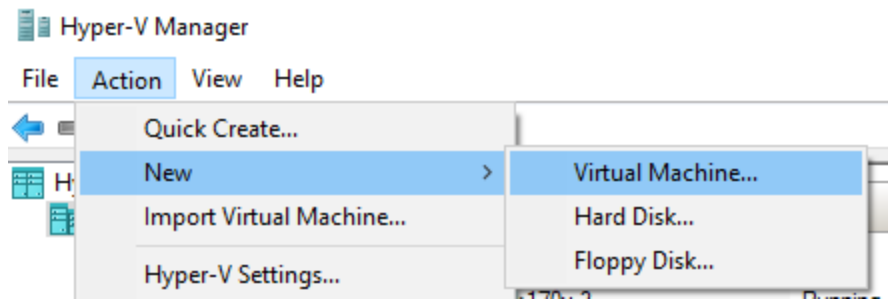
For example, for Windows 10, refer to: [Install Hyper-V on Windows 10](#)

- Open Hyper-V Manager on your Windows system.

#### Create a Virtual CipherTrust Manager instance

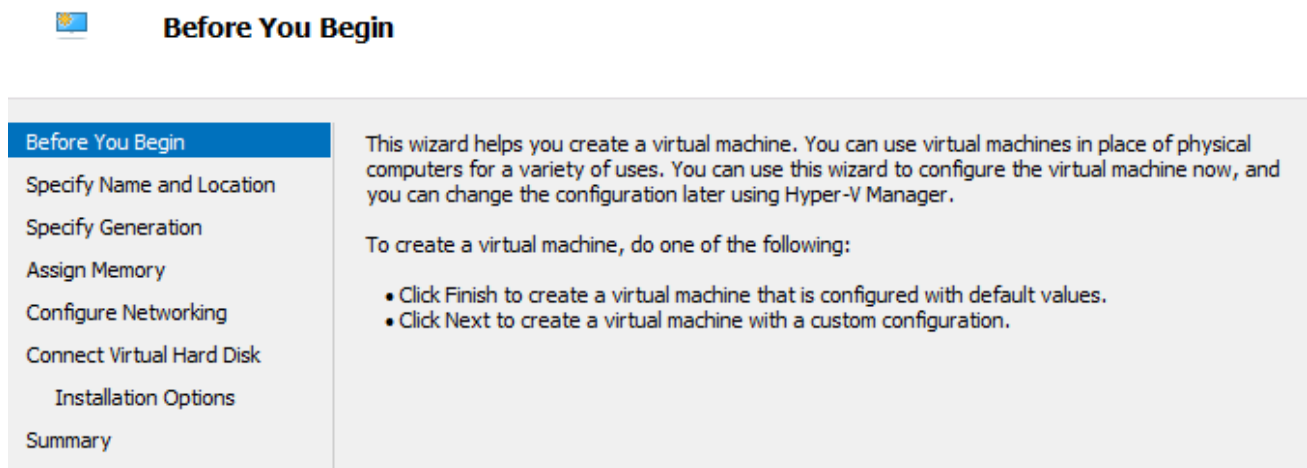
- In the Hyper-V Manager UI, select **Action > New > Virtual Machine**.





The 'New Virtual Machine Wizard' begins with the 'Before You Begin' screen.

#### New Virtual Machine Wizard



- Here you have the choice to either 'create a virtual machine with default values' and then modify them later, or you can 'create a virtual machine with custom configuration'. This procedure uses the option to 'create a virtual machine with custom configuration'. Select **Next**.
- In the 'Select Name and Location' screen,

## New Virtual Machine Wizard



## Specify Name and Location

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Before You Begin                 | <p>Choose a name and location for this virtual machine.</p> <p>The name is displayed in Hyper-V Manager. We recommend that you use a name that helps you easily identify this virtual machine, such as the name of the guest operating system or workload.</p> <p>Name: <input type="text" value="k170v_test"/></p> <p>You can create a folder or use an existing folder to store the virtual machine. If you don't select a folder, the virtual machine is stored in the default folder configured for this server.</p> <p><input type="checkbox"/> Store the virtual machine in a different location</p> <p>Location: <input type="text" value="C:\ProgramData\Microsoft\Windows\Hyper-V\"/> <input data-bbox="1323 640 1458 682" type="button" value="Browse..."/></p> <p> If you plan to take checkpoints of this virtual machine, select a location that has enough free space. Checkpoints include virtual machine data and may require a large amount of space.</p> |
| <b>Specify Name and Location</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Specify Generation               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Assign Memory                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Configure Networking             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Connect Virtual Hard Disk        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Installation Options             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Summary                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

- Enter a *descriptive name* of the virtual machine you are creating.
  - Enter the *folder path* of where the virtual machine will reside, or use the specified default folder path.
  - Select **Next**.
- In the 'Specify Generation' screen, select **Generation 1** and then select **Next**.

## New Virtual Machine Wizard



## Specify Generation

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Before You Begin          | <p>Choose the generation of this virtual machine.</p> <p><input checked="" type="radio"/> Generation 1<br/>This virtual machine generation supports 32-bit and 64-bit guest operating systems and provides virtual hardware which has been available in all previous versions of Hyper-V.</p> <p><input type="radio"/> Generation 2<br/>This virtual machine generation provides support for newer virtualization features, has UEFI-based firmware, and requires a supported 64-bit guest operating system.</p> <p> Once a virtual machine has been created, you cannot change its generation.</p> |
| Specify Name and Location |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Specify Generation</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Assign Memory             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Configure Networking      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Connect Virtual Hard Disk |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Installation Options      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Summary                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

- In the 'Assign Memory' screen, select the desired amount of memory for you new virtual machine. Note the Private Cloud minimum requirement for this setting in [Minimum Requirements](#).

## New Virtual Machine Wizard



## Assign Memory

Before You Begin

Specify Name and Location

Specify Generation

Assign Memory

Configure Networking

Connect Virtual Hard Disk

Installation Options

Summary

Specify the amount of memory to allocate to this virtual machine. You can specify an amount from 32 MB through 12582912 MB. To improve performance, specify more than the minimum amount recommended for the operating system.

Startup memory:  MB
☒ Use Dynamic Memory for this virtual machine.

**i** When you decide how much memory to assign to a virtual machine, consider how you intend to use the virtual machine and the operating system that it will run.

- a. Enter the *value*.
  - b. If you wish to 'use Dynamic Memory' for this virtual machine, **leave the box checked** (default).
  - c. Select **Next**.
6. In the 'Configure Networking' screen drop down box, choose **Default Switch** and then select **Next**.

## New Virtual Machine Wizard



## Configure Networking

Before You Begin

Specify Name and Location

Specify Generation

Assign Memory

Configure Networking

Connect Virtual Hard Disk

Installation Options

Summary

Each new virtual machine includes a network adapter. You can configure the network adapter to use a virtual switch, or it can remain disconnected.

Connection:  ▼

7. In the 'Connect Virtual Hard Disk' screen,

New Virtual Machine Wizard

## Connect Virtual Hard Disk

Before You Begin  
Specify Name and Location  
Specify Generation  
Assign Memory  
Configure Networking  
**Connect Virtual Hard Disk**  
Summary

A virtual machine requires storage so that you can install an operating system. You can specify the storage now or configure it later by modifying the virtual machine's properties.

☐ Create a virtual hard disk  
Use this option to create a VHDX dynamically expanding virtual hard disk.

Name:   
Location:    
Size:  GB (Maximum: 64 TB)

☒ Use an existing virtual hard disk  
Use this option to attach an existing virtual hard disk, either VHD or VHDX format.

Location:

☐ Attach a virtual hard disk later  
Use this option to skip this step now and attach an existing virtual hard disk later.

< Previous   **Next >**   Finish   Cancel

- a. Select the button for **Use an existing virtual hard disk**,
  - b. Enter the location of the 'vhdx' file you decompresses earlier, e.g. **c:/k170v-1.x.x.xxxx.vhdx**
  - c. Select **Next**.
8. In the 'Completing the New Virtual Machine Wizard screen,



## Completing the New Virtual Machine Wizard

Before You Begin

Specify Name and Location

Specify Generation

Assign Memory

Configure Networking

Connect Virtual Hard Disk

**Summary**

You have successfully completed the New Virtual Machine Wizard. You are about to create the following virtual machine.

Description:

Name: k170v\_test

Generation: Generation 1

Memory: 4096 MB

Network: Default Switch

Hard Disk: C:\k170v-1.8.0.3403.vhdx (VHDX, dynamically expanding)

To create the virtual machine and close the wizard, click Finish.

< Previous

Next >

**Finish**

Cancel

- Review the Summary description of the virtual machine your are about to create.
- To make a change, select 'Previous' to go back to change setting(s).
- When ready, select **Finish**.

Your new virtual machine is created within a few seconds.

### Connect to you new virtual machine

- In the Hyper-V Manager, right-click on the new virtual machine and select **Connect** to connect to the new Virtual CipherTrust Manager instance.

The initial screen shows that the VM is turned off.

- Click on **Start**.
- Access the **Console Window** of you newly created virtual machine,

The window opens in your browser. IP information is displayed along with the **ciphertrust login:** prompt.

- At the **ciphertrust login:** prompt, enter "ksadmin" to log in and follow the prompts to create a secure

password.

5. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
6. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                                 | Required Private Key Format |
|-------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

7. Login as `ksadmin` to an SSH session or cloud web console.
8. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

9. If an IP address is not displayed, you likely do not have DHCP configured and you are not using [Cloud-init configuration](#). You do not have access to a static IP for the CipherTrust Manager GUI. In this case, use the [nmcli utility to configure a static IP address](#). `nmcli` can also be used to set static routes, bond network interfaces, disable IPv6, or configure VLAN, if needed for your network.
10. Access the GUI by browsing to the Virtual CipherTrust Manager's IP address in a web browser.

The initial **CipherTrust Manager** Web Page screen is displayed:

# THALES

## Error



- Following services are starting up: platform, ciphertrust-cloud-key-manager, scheduler, backup, key-vault, crypto, licensing, nae, kmip
- The CipherTrust Manager will be functional after the default SSH public key for the ksadmin user is replaced

## SSH Public Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQBDiLlOH4tInSyWQuJkBLJ2sdce4BsvaFa5eL
6Xh1xgELXArNpjZtWvikQkz3JD/2TZbyzdLVACvfoVIFx2MToOw83WsuxeUua95NW
r1ehqeaS9Bx1
/v5OkogEuTzYN1gXyzQQbfbPEOHIDR+y7FNvlyw5++ZUhi2iMIHQEPeZlhoTYHXG1
RpR438JXoh/C8ey8aaP6cdZcTudQdsdDP3HGWXjsP
/NpOw4yeBthoy5NFuPpmS0Uvd0k2uCYkq6bkW7wwiRbW21joGBEf2muyKiCc2Fa
uVIFvgQNORaj1WjnRcWOjMlfcf2slwue8h2YsAfnzeB/gYhH+gdxfdK8aZ
user@example.local
```

Add

11. The Error displayed is normal and simply requires the default **SSH Public Key** to be replaced. As the initial KeySecure admin (ksadmin) you must paste in your **SSH Public Key** in the box provided and then select **Add**.

## Note

We support OpenSSH format for the public key, and OpenSSH, PKCS1, or PKCS8 format for the private key. RSA is the supported key algorithm. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security. You can generate this key pair using 'PuTTYgen' or similar utility. Save the SSH public and private keys. The SSH Private key, used to access the System Administrative

account "ksadmin", is extremely sensitive and should be kept in a secure environment.

After replacing the default SSH Public Key, the Log In screen appears. For more options to replace the default SSH Public Key, see [Starting Services After Deployment](#).

12. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



### Password Change

After successfully changing your password, you will be logged out and will have to log in again

### Note

If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [Changing the Initial Password](#).

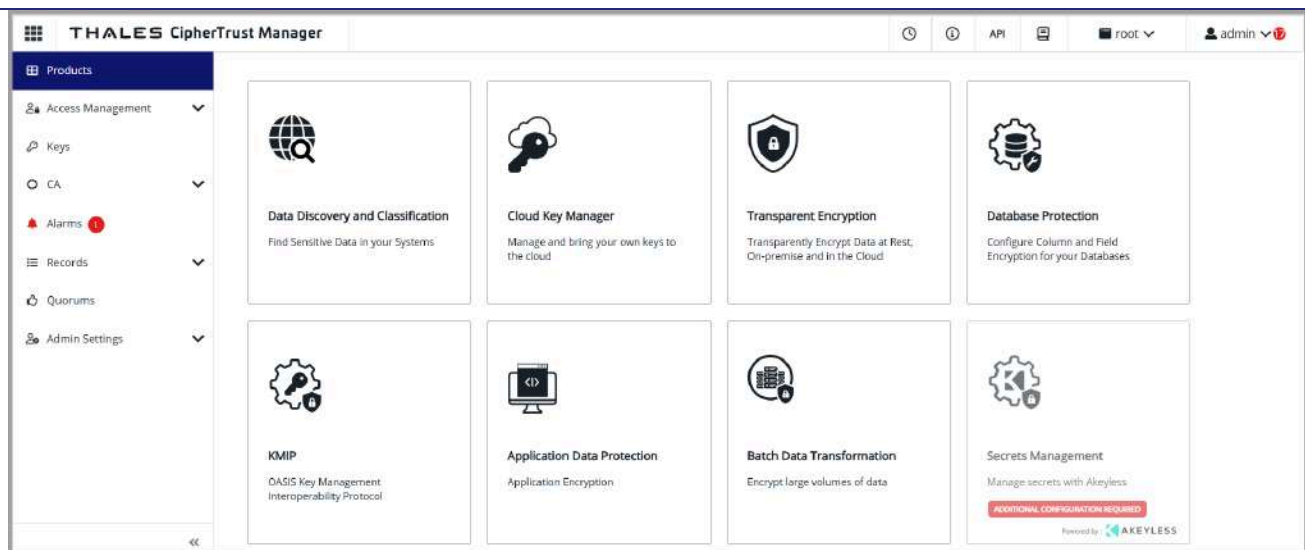
13. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

14. Using your new password, log in again. The **CipherTrust Manager** Web Page appears.





Congratulations! You have successfully deployed your CipherTrust Manager virtual machine.

15. At this point, it's strongly recommended to configure an NTP server.
  - a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

16. If you did not apply disk encryption with cloud-init, it is available [after first launch with ksctl](#). Because installation specific secrets are generated the first time a Virtual CipherTrust Manager instance is launched, it is recommended that the instance be encrypted at launch time to ensure these secrets are never exposed.

### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

## Using Cloud-init with Hyper-V

The Virtual CipherTrust Manager uses cloud-init to inject configuration information prior to first boot. Cloud-init is a standard configuration mechanism that is supported in Hyper-V. Cloud-init is supported as an ISO image for Hyper-V, mounted as media to use with the virtual CD/DVD drive. The ISO image must have two text files named **user-data** and **meta-data**.

The configuration parameters that save deployment steps after first launch are disk encryption, static IP, and

disabling IPv6. Because installation specific secrets are generated the first time a Virtual CipherTrust Manager instance is launched, it is recommended that the instance be encrypted at launch time to ensure these secrets are never exposed.

Below is an example of setting up disk encryption and the user's ssh key using cloud-init and Hyper-V.

### Example: To setup disk encryption and the user's SSH key

1. Prepare the **user-data** file. This is a text file containing the specific configuration you want to use. Refer to ["Plan Configuration Settings for Deployment with Cloud Init"](#) for other available configuration parameters. Here is a simple **user-data** configuration file for setting up disk encryption and the user's SSH key.

```
#cloud-config
diskenc:
 encrypt: true
ssh_authorized_keys:
 - <replace with your OpenSSH format ssh public key>
```

#### Note

You must provide the SSH key in OpenSSH format. The corresponding private key can be OpenSSH, PKCS1, or PKCS8 format. RSA is the supported key algorithm. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security.

2. Create the **meta-data** file. This is a text file containing user provided instance parameters, for example:

```
instance-id: <some instance id>`
local-hostname: <host name for the instance>
```

3. Create the ISO image file:
  - a. Make sure **genisoimage** utility is installed.
  - b. Create the ISO file:

```
genisoimage -o config.iso -volid cidata -joliet -rock user-data meta-data
```

4. Create a Virtual CipherTrust Manager in Hyper-V, but do not boot it.
5. In Hyper-V manager:
  - a. Right click on the instance and go to "Settings...".
  - b. Under **IDE Controller 1 > DVD Drive > Specify** the media to use with your virtual CD/DVD drive, select

---

#### Image File

- c. Browse to the **config.iso** file generated in step 3 above.
  - d. Select OK.
6. Boot up your Virtual CipherTrust Manager.

## Deploying on the Nutanix AHV

This section guides you through the steps needed to deploy a Virtual CipherTrust Manager on Nutanix AHV.

### Prerequisites

- OVA file: It is mandatory to decompress the OVA file before deployment to generate a VMDK file. For steps, see [Decompressing an OVA file](#)

## Creating a CipherTrust Manager Virtual Machine

Perform the following steps to create a CipherTrust Manager virtual machine on a Nutanix Cluster:

#### Caution

Do not host a CipherTrust Manager virtual machine on the encrypted cluster where it is being used. This can lead to complete data loss if there is an issue with the VM while it is hosted in that cluster.

1. Go to the Prism Element.
2. Create a Storage Container.
  - a. Go to the **Storage** page and click **Storage Container**.

The Create Storage Container window appears.
  - b. Specify the **Name** and select a **Storage Pool**.
  - c. (Optional Step) If required, click **Advanced Settings** for the settings related to compression, cache, duplication, and so on.
  - d. Click **Save**.

The New Storage Container will be added.
3. Create/Upload an image on the Nutanix Cluster.
  - a. Go to **Settings > Image Configuration**. Click **Upload Image**.

---

The Create Image page appears.

b. Specify/select the following fields/options:

- Name
- Annotations
- Image Type
- Select the Storage Container
- Image Source

c. Click **Save**.

The application will start uploading the image. This can take some time depending on the network speed. After the uploading process is complete, the new image is created, and the image activation process is initiated. You can use the image after it becomes active. This image can be accessed from **Settings > Image Configuration**.

4. Go to the VM page and click **Create VM**.

5. Specify/select the following fields/options:

a. General Configuration

- Name
- Description
- Timezone

b. Minimum Requirements

Refer to [Minimum Requirements](#).

c. Disks

Remove any existing disk and click **Add New Disk**. Specify the following fields:

- **Type:** DISK
- **Operation:** Clone from Image Service
- **Bus Type:** SCSI
- **Image:** Select image created in the previous step
- **Index:** Next Available

---

Click **Add**. The disk will be added.

d. Set **Boot configuration** to **Legacy BIOS**.

e. Network Adapter

Click **Add New NIC** and specify the following details:

- **Network Name**: Select a pre-configured network
- **VLAN ID**: Specify the required value
- **Network Connection State**: Set to Connected
- **Network Address/Prefix**: Set to the required value

Click **Add** to add the Network Adapter.

f. Go to **VM Host Affinity** and select the Host(s) as required. Click **Save**.

6. Click **Save** after completing all configurations.

The new VM will be created.

## Booting the CipherTrust Manager VM

Go to the VM page and locate the newly added VM in the VM Name column. You can use search option, if required.

1. Click on the VM name and scroll down.
2. Click **Power On**.
3. Access the CipherTrust Manager in a web browser using its IP Address.
4. Access the GUI by browsing to the Virtual CipherTrust Manager's IP address in a web browser.

The initial **CipherTrust Manager** Web Page screen is displayed:

# THALES

## Error



- Following services are starting up: platform, ciphertrust-cloud-key-manager, scheduler, backup, key-vault, crypto, licensing, nae, kmip
- The CipherTrust Manager will be functional after the default SSH public key for the ksadmin user is replaced

## SSH Public Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQADiLlOH4tInSyWQuJkBLJ2sdce4BsvaFa5eL
6Xh1xgELXArNpjZtWvikQkz3JD/2TZbyzdLVACvfoVIFx2MToOw83WsuxeUua95NW
r1ehqeaS9Bx1
/v5OkogEuTzYN1gXyzQQbfbPEOHIDR+y7FNvlyw5++ZUhi2iMIHQEPeZlhoTYHXG1
RpR438JXoh/C8ey8aaP6cdZcTudQdsdDP3HGWXjsP
/NpOw4yeBthoy5NFuPpmS0Uvd0k2uCYkq6bkW7wwiRbW21joGBEf2muyKiCc2Fa
uVIFvgQNORaj1WjnRcWOjMlfcf2slwue8h2YsAfnzeB/gYhH+gdxfdK8aZ
user@example.local
```

Add

5. The Error displayed is normal and simply requires the default **SSH Public Key** to be replaced. As the initial KeySecure admin (ksadmin) you must paste in your **SSH Public Key** in the box provided and then select **Add**.

## Note

We support OpenSSH for the public key format. The corresponding private key can be OpenSSH, PKCS1, or PKCS8 format. You can generate this key pair using 'PuTTYgen' or similar utility. Save this SSH Public Key at a safe location. You will need this key for future SSH access.

After replacing the default SSH Public Key, the Log In screen appears. For more options to replace the default

SSH Public Key, see [Starting Services After Deployment](#).

- Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



### Note

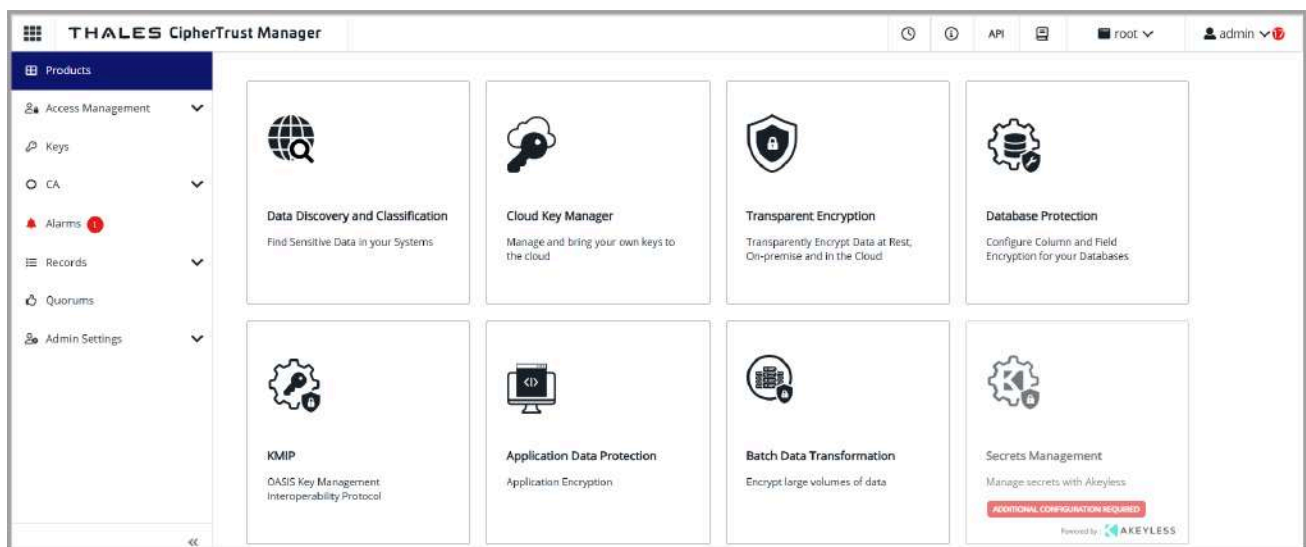
If the default credentials do not work, you may need to retrieve an auto-generated password, as described in [Changing the Initial Password](#).

- Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

- Using your new password, log in again. The **CipherTrust Manager** Web Page appears.



9. At this point, it's strongly recommended to configure an NTP server.
  - a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

10. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
11. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                                 | Required Private Key Format |
|-------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

12. Login as `ksadmin` to an SSH session or cloud web console.
13. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

Congratulations! You have successfully deployed your Virtual CipherTrust Manager.

#### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).



---

## Deploying on Red Hat OpenShift

To deploy on Red Hat OpenShift, there are four stages:

1. Satisfy the [prerequisites](#)
2. [Launch the Virtual CipherTrust Manager Instance.](#)
3. [Set up networking for the launched Virtual CipherTrust Manager.](#)
4. [Perform initial configuration on the Virtual CipherTrust Manager](#)

### Prerequisites

- Red Hat OpenShift cluster version 4.18.0

#### Caution

Other versions have not been tested and might not work.

- Open Shift Virtualization operator.
- QCOW2 image for desired CipherTrust Manager version

### Launch the Virtual CipherTrust Manager Instance on Red Hat OpenShift

1. Launch OpenShift Virtualization.
2. Navigate to your namespace/project, **Virtualization** and click **VirtualMachines**.
3. Click **Create** in the upper right corner, and select **From template**.
4. Select **Red Hat Enterprise Linux 8 VM** from the **Catalog**.
5. Select **CPU|Memory** settings appropriate for your desired load, at or above the [minimum requirements](#).
6. Under **Disk source** select **Upload (Upload a new file to a PVC)**.
7. Navigate to and select the QCOW2 image on your local system.
8. Select **Disk space** settings appropriate for your desired load, at or above the [minimum requirements](#).
9. Provide a **VirtualMachine name**.
10. Click **Quick create VirtualMachine**.

You can monitor the progress bar for QCOW2 file upload and virtual machine launch. Once the launch

---

completes, the virtual machine is visible at **Virtualization > VirtualMachines**.

11. Proceed to **Set Up Networking and Access the New Virtual CipherTrust Manager**.

## Set up Networking and Access the New Virtual CipherTrust Manager

You need to provide a hostname and expose access to the 443 port to make CipherTrust Manager available for users. You do this by creating a Nodeport service and a route.

1. Navigate to **Networking > Service**.
2. Click **Create**.
3. In the **yaml** text field, paste the following text, and edit the `namespace` and `selector` values:

```
apiVersion: v1
kind: Service
metadata:
 name: vm-nodeport-service
 namespace: openshift-cnv # Replace with your namespace
spec:
 type: NodePort
 selector:
 kubevirt.io/domain: ciphertrustmanager-vm # Replace with your VM's name
 ports:
 - protocol: TCP
 port: 443 # Port exposed by the service
 targetPort: 443 # Port on the VM
 nodePort: 31080 # Optional: specific NodePort (must be between 30000-32767)
```

4. Click **Create Service**.
5. Navigate to **Networking > Route**.
6. Click **Create Route**.
7. Select **yaml view**, paste the following text, and edit the `name`, `namespace`, and `host` values:

```
kind: Route
apiVersion: route.openshift.io/v1
metadata:
 name: route-cm-web #name of the route
 namespace: openshift-cnv # Replace with your namespace
 uid: 0ada78fd-b6e8-460e-b797-396497b35390
```

```

resourceVersion: '26083717'
creationTimestamp: '2025-06-05T18:18:36Z'
managedFields:
 - manager: openshift-router
 operation: Update
 apiVersion: route.openshift.io/v1
 time: '2025-06-05T18:18:36Z'
 fieldsType: FieldsV1
 fieldsV1:
 'f:status':
 'f:ingress': {}
 subresource: status
 - manager: Mozilla
 operation: Update
 apiVersion: route.openshift.io/v1
 time: '2025-06-05T20:16:19Z'
 fieldsType: FieldsV1
 fieldsV1:
 'f:spec':
 'f:port':
 .: {}
 'f:targetPort': {}
 'f:tls':
 .: {}
 'f:termination': {}
 'f:to':
 'f:kind': {}
 'f:name': {}
 'f:weight': {}
 'f:wildcardPolicy': {}
spec:
 host: www.ciphertrustmanager.com # provide the hostname through which you
 can access the CipherTrust webUI
 to:
 kind: Service
 name: vm-nodeport-service # provide the service name to get it attache
d
 weight: 100
 port:
 targetPort: 443 #provide port number
 tls:
 termination: edge
 wildcardPolicy: None
status:

```

```
ingress:
 - host: ciphertrustmanager
 routerName: default
 conditions:
 - type: Admitted
 status: 'True'
 lastTransitionTime: '2025-06-05T18:18:36Z'
 wildcardPolicy: None
 routerCanonicalHostname: router-default.apps.cw.gemalto.com
```

8. Attach the new service to the Virtual CipherTrust Manager.
  - a. Navigate to **Virtualization > VirtualMachines**.
  - b. Click the name of the Virtual CipherTrust Manager.
  - c. Click **Services**, and select the **Edit Service** option.
  - d. Provide the **Service Name**, `vm-nodeport-service`.
  - e. Click **Save**.
9. Access the CipherTrust WebUI by browsing to the Nodeport host specified in the route in step 7, and the Nodeport port number specified in the service in step 3. Following the example configuration, the URL would be `https://ciphertrustmanager.com:31080`

The initial **CipherTrust Manager** Web Page screen is displayed:



### Error



- Following services are starting up: platform, ciphertrust-cloud-key-manager, scheduler, backup, key-vault, crypto, licensing, nae, kmip
- The CipherTrust Manager will be functional after the default SSH public key for the ksadmin user is replaced

### SSH Public Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQBDiL1OH4tInSyWQuJkBLJ2sdce4BsvaFa5eL
6Xh1xgELXArNpjZtWvikQkz3JD/2TZbyzdLVACvfoVIFx2MToOw83WsuxeUua95NW
r1ehqeaS9Bx1
/v5OkogEuTzYN1gXyzQQbfbPEOhIDR+y7FNvlyw5++ZUhi2iMIHQEPeZlhoTYHXG1
RpR438JXoh/C8ey8aaP6cdZcTudQdsdDP3HGWXjsP
/NpOw4yeBthoy5NFuPpmS0Uvd0k2uCYkq6bkW7vwviRbW21joGBEf2muyKiCc2Fa
uVIFvgQNORaj1WjnRcWOjMlf2slwue8h2YsAfnzeB/gYhH+gdxfdK8aZ
user@example.local
```

Add

## 10. Proceed to Initial Setup on Virtual CipherTrust Manager

### Initial Configuration on Virtual CipherTrust Manager

1. As the initial KeySecure admin (ksadmin) you must paste in your **SSH Public Key** in the box provided and then select **Add**.

#### Note

We support OpenSSH for the public key format. The corresponding private key can be OpenSSH,

PKCS1, or PKCS8 format. You can generate this key pair using 'PuTTYgen' or similar utility. Save this SSH Public Key at a safe location. You will need this key for future SSH access.

After replacing the default SSH Public Key, the Log In screen appears. For more options to replace the default SSH Public Key, see [Starting Services After Deployment](#).

2. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



#### Note

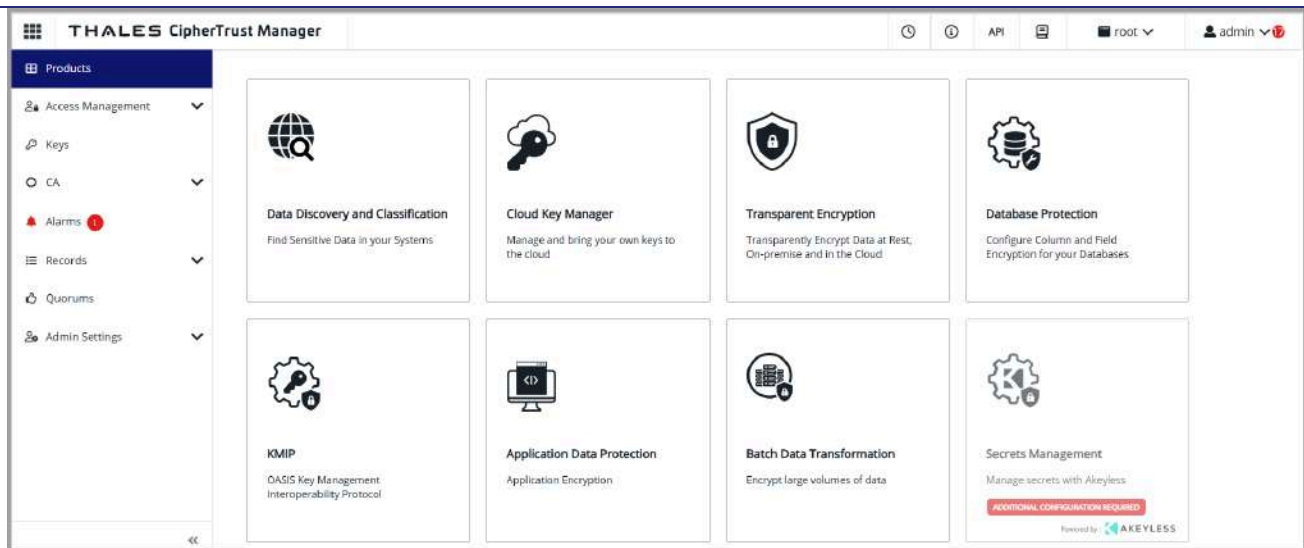
If the default credentials do not work, you may need to retrieve an auto-generated password, as described in [Changing the Initial Password](#).

3. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new Login screen appears.

4. Using your new password, log in again. The **CipherTrust Manager** Web Page appears.



5. At this point, it's strongly recommended to configure an NTP server.
  - a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

6. Obtain the `ks_patch_disable_cloud_init.tar.gz.gpg` archive file from Thales support portal (refer to [KB0029319](#)). This archive file is required to disable the cloud-init service after first launch, to prevent unauthorized usage of cloud-init.
7. Transfer the archive file to the CipherTrust Manager. You require the private SSH key associated with the `ksadmin` account. The command and supported private key format depends on the operating system you are transferring the archive file from.

| Source Operating System | Example Command Syntax                                                                            | Required Private Key Format |
|-------------------------|---------------------------------------------------------------------------------------------------|-----------------------------|
| Linux                   | <pre>scp -i &lt;path_to_private_SSH_key&gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre>  | OpenSSH format              |
| Windows                 | <pre>pscp -i &lt;path_to_private_SSH_key&gt; &lt;archive_file_name&gt; ksadmin@&lt;ip&gt;:.</pre> | PuTTY PPK format            |

8. Login as `ksadmin` to an SSH session or cloud web console.

9. Apply the archive file to disable cloud-init.

```
sudo /opt/keysecure/ks_upgrade.sh -f <archive_file_path>
```

Congratulations! You have successfully deployed your Virtual CipherTrust Manager.

#### Note

Virtual CipherTrust Manager launches in Community Edition, with some restrictions on functionality. You can activate a 90 day trial evaluation for full functionality. To activate your instance with a trial evaluation, or a term or perpetual license, see [Licensing](#).

## Disk Resize Available with Disk Encryption

After you launch an unencrypted Virtual CipherTrust Manager and disable the cloud-init service, you cannot resize the disk. If you require a Virtual CipherTrust Manager with a larger or smaller disk size:

- [Encrypt the disk](#) for the Virtual CipherTrust Manager instance and then attempt re-size.
- [Take a backup](#) on the existing Virtual CipherTrust Manager, deploy a new Virtual CipherTrust Manager instance with the desired disk size, and restore the backup file on the new instance.

## Resizing the Virtual Hard Disk After Deployment

You can use the cloud provider's tools to resize a virtual CipherTrust Manager instance's hard disk size after deployment.

## Disk Resize Available with Disk Encryption

After you launch an unencrypted Virtual CipherTrust Manager and disable the cloud-init service, you cannot resize the disk. If you require a Virtual CipherTrust Manager with a larger or smaller disk size:

- [Encrypt the disk](#) for the Virtual CipherTrust Manager instance and then attempt re-size.
- [Take a backup](#) on the existing Virtual CipherTrust Manager, deploy a new Virtual CipherTrust Manager instance with the desired disk size, and restore the backup file on the new instance.

If you have applied [disk encryption](#), you need a [ksctl connection](#) to unlock the instance after resizing.

## Resizing an instance

1. Shut down the Guest OS or power off the virtual CipherTrust Manager instance.
2. Use the cloud provider's tooling to resize the hard disk.
  - For Amazon Web Services, under **Actions**, select **Modify Volume**, edit the **Size** field, and select **Modify**.



---

Check [Amazon EC2 documentation](#) for more details.

- For Google cloud, follow [Google Cloud documentation](#).
- For Microsoft Azure, consult [Microsoft Azure documentation](#).
- For Microsoft Hyper-V, consult [Microsoft Hyper-V documentation](#).
- For Nutanix AHV, consult [Nutanix documentation](#).
- For Oracle Cloud, consult [Oracle Documentation](#).
- In VMware vSphere, right click the virtual CipherTrust Manager instance, select **Edit Settings**, and adjust the **Hard disk** size. Consult [VMware vSphere documentation](#) for more details.

#### Note

The resulting disk size will be slightly smaller (within 2 GB) of the setting you apply here.

3. Power the device back on.
4. If the disk was encrypted, booting pauses until you secure boot with the following command:

```
$ ksctl diskenc secureboot -i <private ssh key for the instance> -u https://<instance dns name>
```

The instance resumes booting.

5. To confirm the disk resize:
  - a. Use the **Web Console** for SSH access to the instance.
  - b. Login the `ksadmin` user.
  - c. Run the following command to display the disk size used `df -h /`.

## Install Physical CipherTrust Manager Appliance

This section describes the installation tasks you perform in your data center to set up a **k470** or **k570** CipherTrust Manager appliance.

**Note**

**k160** set up instructions are available [here](#).

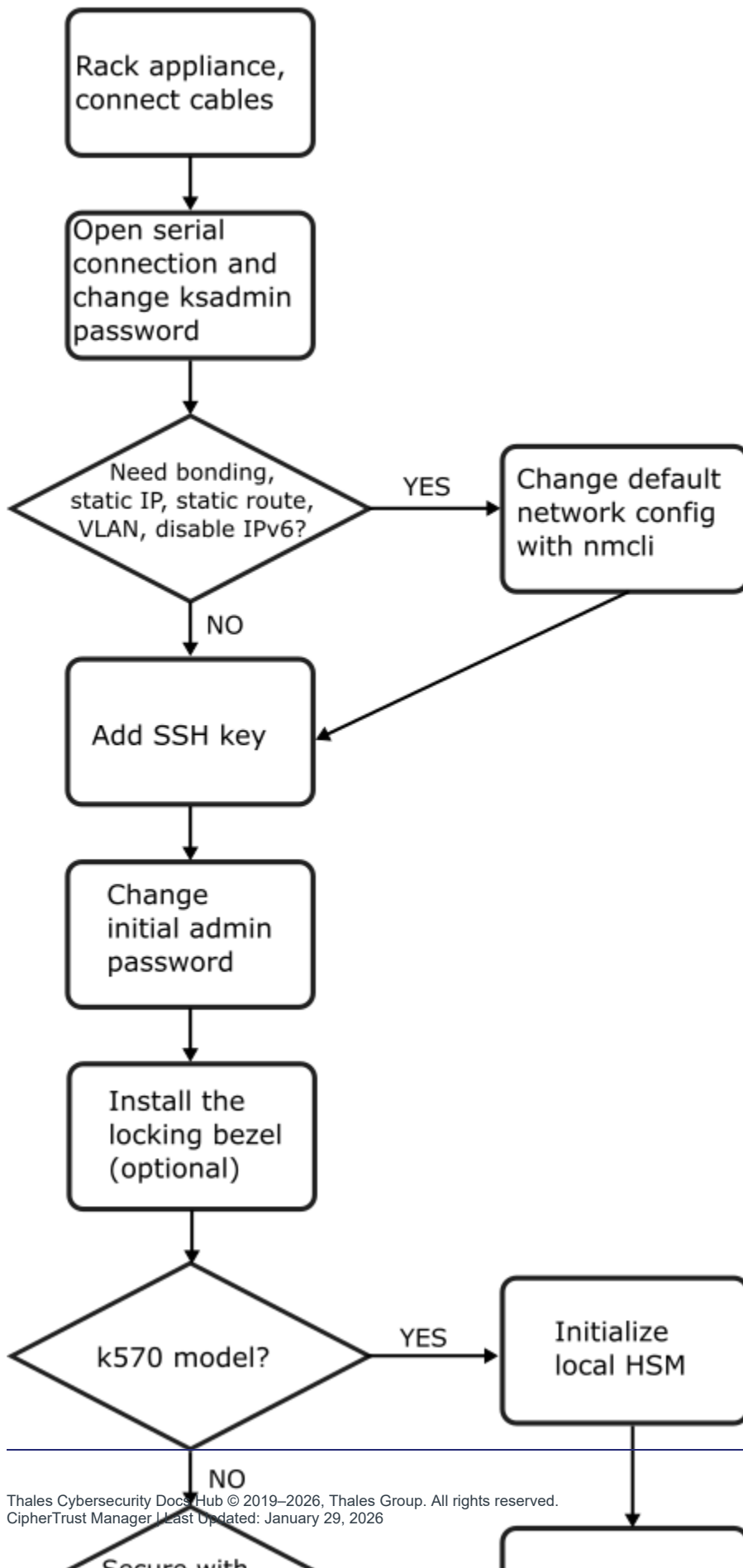
Set up tasks include:

- Unpacking and physically installing the appliance into a rack.
- Changing default passwords for the System Administrator (ksadmin) and the initial admin user using a local serial connection and the initial DHCP assigned IP address.
- For PED authenticated k570 models, initializing the embedded Luna PCIe HSM and PED key roles.

The basic workflow for deploying a physical appliance is shown in the below diagram.

---

## Physical Appliance Deployment



---

## k470 and k570 Prerequisites

The following items are required to install the CipherTrust Manager appliance in a rack, and deploy the CipherTrust Manager firmware.

- Supported Appliances: CipherTrust Manager k470 or k570
- #2 Phillips screwdriver
- hydraulic equipment lift
- Terminal server, dumb terminal, PC, or laptop to establish an install serial connection to the appliance using the supplied USB to RJ45 adapter.
- Firmware: CipherTrust Manager release 2.0 or later. (Contact [Technical Support](#))

### For PED authenticated k570 model only:

- PIN Entry Device (PED)
- Three PED keys (blue, red and black)

## Verify the Integrity of Your k470 or k570 Shipment

### Caution

Thales employs a number of security measures to allow you to verify that your new hardware was not intercepted in transit or otherwise tampered with before you received it. To verify the authenticity and handling history of your received items, review the following checklist before you unpack your new hardware, and then follow the checklist as you unpack each received item.

1. Do the items received (individual items, part numbers) match those listed in the enclosed packing list? If yes, go to the next step. If no, contact Thales support.
2. Before you received the product, did you receive an advanced shipping notification providing details regarding the shipment (part numbers and serial numbers for the product and tamper-evident bags)? If yes, go to the next step. If no, contact Thales support.
3. Are all of the tamper-evident bag serial numbers and tamper-evident label serial numbers listed in the advanced shipping notification present, and do they match the actual tamper-evident bag/label serial numbers received? If yes, go to the next step. If no, contact Thales support.
4. Did you receive any tamper-evident bag/label serial numbers that are not listed on the advance shipping notification? If yes, contact Thales support. If no, go to the next step.
5. Are there any signs of physical tampering? If tamper-evident labels are affixed to the received product, have any of these labels been damaged? Have the tamper evident bags been damaged in any way? The tamper

seals on the sides indicate tampering if they show the ALERT markings as illustrated below. If yes, contact Thales support. If no, go to the next step.



## Check Received k470 or k570 Items

This section provides a list of the components you should have received with your CipherTrust Manager k470 or k570 order. If you ordered a PED-authenticated k570 model, some additional components are included as described in [PED related order items](#).

### Basic Order Items

- **1 CipherTrust Manager Appliance:** Fits any standard 19-inch server rack.

#### Note

You can use the part number on the product label to verify the if the appliance's model is k470 or k570. The part number for k570 models also indicates whether the appliance is password-authenticated or PED authenticated.



- **2 power supply cords:** One for each power supply, with connectors appropriate to your region of operation.



- **1 Adapter Cable: RJ45 to USB with a standard eight-pin, eight connector (8P8C) modular connector:**  
Used to connect a console terminal to the appliance during initial configuration.



- **1 Front Ear Bracket Set:** Set includes 2 front ear brackets and 4 bracket screws



- **1 Friction Rail Mounting Bracket Set:** See [Using the Supplied Mounting Brackets](#) for installation instructions.  
Set includes: 2 side rails, 8 side rail screws, 2 sliding rear brackets (fit into the rails for rear support adjustable positioning)



### Caution

The included mounting hardware is meant for static positioning of the appliance. The long tab that slides into the bracket, applied to each side of the appliance, is adjustable for fitting the appliance into racks of varying depth - it must not be used to extend the appliance out of the rack.

Optional gliding rails with rolling bearings are available for situations where rolling excursion of the appliance, while attached to the rack, is required for maintenance. See [Optional Items](#).

- **1 Friction Rail Rack Mounting Screws/Cage Nuts:** Set includes 8 M5 cage nuts and 8 M5x14 rack screws. If you did not receive this set, you can request one from Thales Group (part number: 216-000035-001) or obtain your own suitable screws/nuts.



- **1 Secure Locking Bezel:** For maximum physical security, this faceplate bezel can restrict access to the CipherTrust Manager front-facing inputs. Some security standards require the use of this bezel. Leaving the appliance uncovered for ease of access might compromise physical security. Includes set of three (3) keys for each lock (locks are keyed differently).



## Optional Items

You may have also ordered one or more of these optional items:

- **1 Sliding Rail Mounting Bracket Set:** The CipherTrust Manager will fit into any standard 19-inch server rack. The optional sliding rail mounts allow for easy removal and access to the rear face of the appliance. See Using the Optional Sliding Rail System for installation instructions. The set includes 2 sliding rail mounts with removable side rails, 2 transformer brackets, 6 rail screws.



- **1 Sliding Rail Rack Mounting Screws Set:** Set includes 8 M5x8 flat-headed screws. If you did not receive this set, you can request one from Thales Group (part number: 216-000034-001) or obtain your own suitable screws. If you do not use the screws included in this kit, ensure that the screw heads are flat enough so as not to interfere with the locking bezel.





- **SFP 10 Gbps Optical Ethernet transceiver modules:** If you ordered the model with 2X10Gbps ports and 2X1Gbps ports, you should have received two SFP 10 Gbps Optical Ethernet transceiver modules, packed separately. To install:
  - a. Locate the two 2X10Gbps ports on the [appliance rear panel](#). These ports are protected by plastic dust covers during shipment.
  - b. Remove these dust covers and insert a transceiver module in each port.



## PED related order items for k570 models

If you ordered a PED-authenticated k570 model, you should have received some combination of the following items in addition to the basic order items.

- **1 PED device:** This device is needed for authentication to the on-board PCIe HSM.



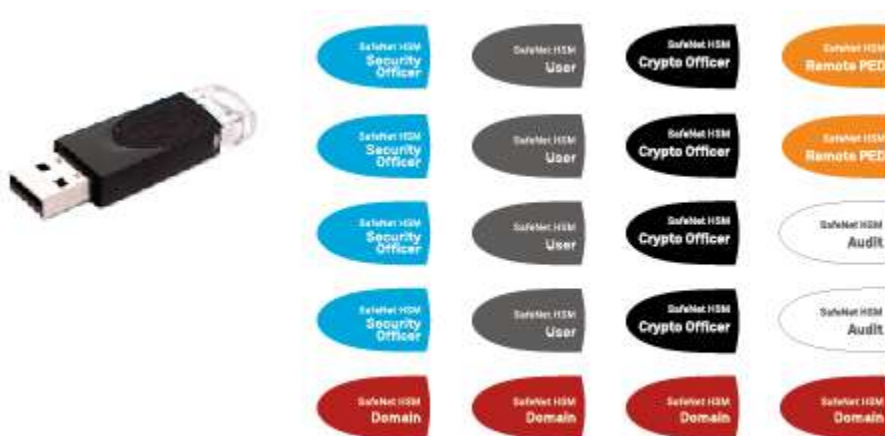
- **1 PED cable:** This is a Type A to Mini B USB cable used to connect the PED device to your CipherTrust Manager.



- **Luna PED Power Supply Kit:** If you ordered a Luna PED, your order should also include a Luna PED power supply kit with the appropriate power connection for your region. The power supply is auto-sensing and includes replaceable mains plug modules for international use.



- **Set of PED Keys and Labels:** Your order should include a set of iKey PED keys and peel-and-stick labels.



## Rack Mounting k470 or k570

If you intend to mount the CipherTrust Manager in a standard equipment rack, front ear brackets, side rails, rear slider brackets, and the necessary screws are packed separately in the carton. You may also have ordered the optional sliding rail mounting system. See [Received Items](#) for details. Instructions for installing both systems are provided below:

- If you intend to use the supplied mounting brackets, see [Using the Supplied Mounting Brackets](#).
- If your order included the optional sliding rail mounting system, see [Using the Optional Sliding Rail System](#). The sliding rails are recommended for ease of installation and maintenance.

**Caution**

Do not attempt to mount the appliance using only the front brackets – damage can occur.

## Using the Supplied Mounting Brackets

Install and adjust the rails and brackets to suit your equipment rack. The standard mounting bracket set is designed for use in racks with a maximum depth of 27 inches (686 mm). For racks larger than 27 inches, a mounting tray or shelf is recommended.

**Caution**

The included mounting hardware is meant for static positioning of the appliance. The long tab that slides into the bracket, applied to each side of the appliance, is adjustable for fitting the appliance into racks of varying depth - it must not be used to extend the appliance out of the rack.

Optional gliding rails with rolling bearings are available for situations where rolling excursion of the appliance, while attached to the rack, is required for maintenance. See [Using the Optional Sliding Rail System](#).

Ensure you have all the necessary components before proceeding. In addition to the supplied components, you will need:

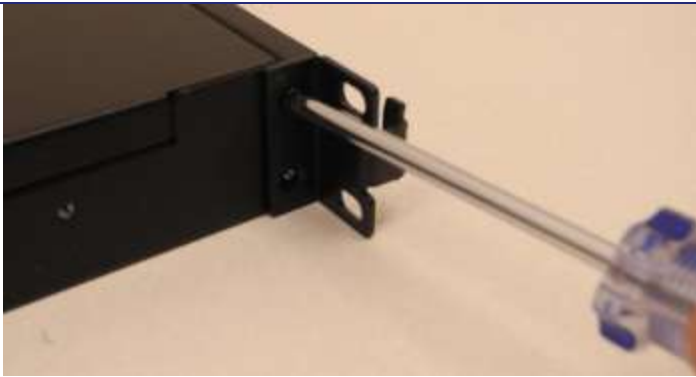
- #2 Philips screwdriver
- hydraulic equipment lift

**Caution**

If you are installing the appliance in a rack without a mounting tray or shelf, ensure that the appliance is supported at all times or damage may occur. Use of a hydraulic equipment lift is strongly recommended. If you do not have access to a lift, you will need at least one assistant to mount the appliance.

### To mount the appliance

1. Install the two front ear mounting brackets on the appliance using the included screws and a #2 Phillips screwdriver.

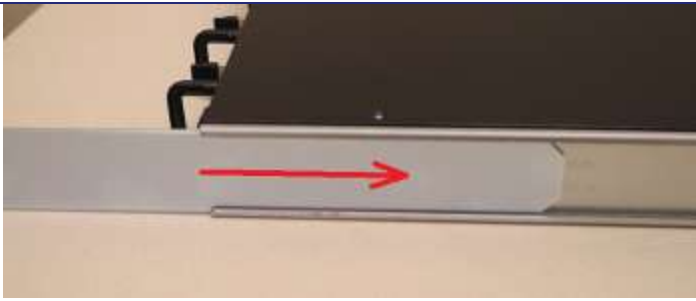


2. Fit eight cage nuts into the rack space where you want to install the appliance. Ensure that they are spaced correctly.



3. Install the two side rails on either side of the appliance, using the included screws and a Phillips screwdriver. Note how the sliding rear brackets fit into the side rails.





4. Install the two sliding rear brackets in your equipment rack using four rack mounting screws.

**Note**

While any standard equipment rack screws should fit the brackets, certain large-headed screws may interfere with the operation of the secure locking bezel.



5. Using a hydraulic lift, raise the appliance to the level of the brackets and extend the lift into the rack.

**Caution**

Perform the next step from the rear of the server rack. Do not push the appliance off the lift without supporting its rear end.

6. From the rear of the server rack, pull the appliance back towards you until the sliding rear brackets fit into the side rails. Pull the appliance back onto the rear brackets until the front ear brackets meet the equipment rack.

**Caution**

Support the weight of the appliance with the hydraulic lift until all four brackets are secured.



7. Secure the front ear brackets using rack mounting screws.



8. Continue to Establish a Connection and Change Default Passwords.

---

## Using the Optional Sliding Rail System

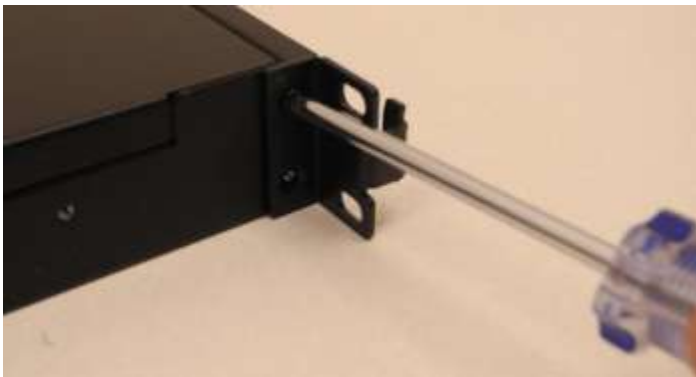
The optional sliding rail system allows for the appliance to be extended out in front of the equipment rack, possibly easing access to other racked appliances. This is rarely necessary.

The sliding rail mounts fit into any standard 19" equipment rack.

Ensure you have all the necessary components before proceeding. In addition to the supplied components, you will need a #2 Philips screwdriver.

### To mount the appliance

1. Install the two front ear mounting brackets on the appliance using the included screws and a #2 Phillips screwdriver.

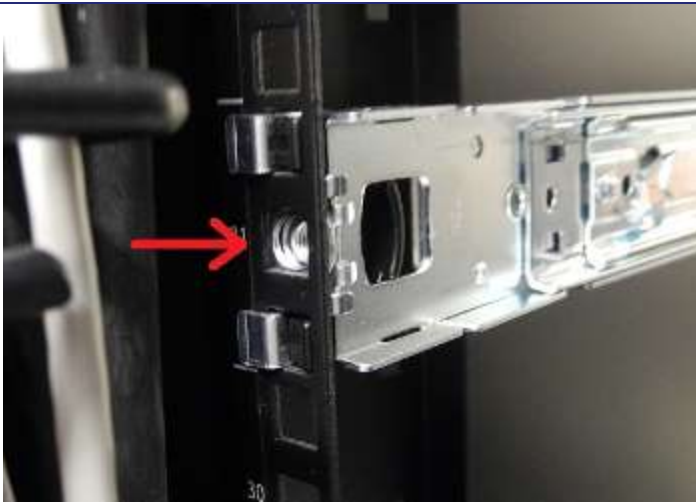


2. Fit the front end of each mount into either side of the rack and pull the spring-loaded latch at the rear to snap it in place.

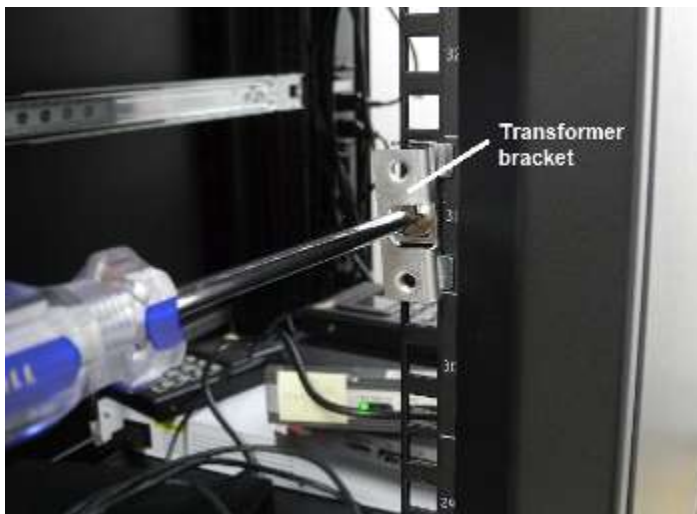


3. Secure the rear end of each mount to the rack with two wide flat-headed screws.





4. Fasten the transformer bracket to each sliding mount with two wide flat-headed screws.



5. Loosely thread two small flat-headed screws into each side of the appliance. Fit each sliding rail over the screw heads and slide it forward into place before tightening the screws. Fasten each sliding rail with a third screw where it lines up with the hole on the appliance.



6. Fit the sliding rails onto the rack mounts until they lock into place.



The appliance now moves smoothly and securely on the rails.

7. Push the appliance all the way back and secure it to the transformer bracket with four rack screws.

#### Note

Screws with heads that are too large can prevent the locking bezel from fitting to the faceplate. Use the screws included with the appliance, or other screws with suitable heads.



8. Continue to [Establish a Connection and Change Default Passwords](#).

## Establish a Connection and Change Default Passwords for k470 or k570

After you have rack-mounted the appliance, you must log in to the console to create a secure password for the kadmin user, and then log in to the GUI to change the default SSH key and admin user password. Changing these defaults ensures the security of CipherTrust Manager and is required before beginning to create keys or engage in any other cryptographic usecases.

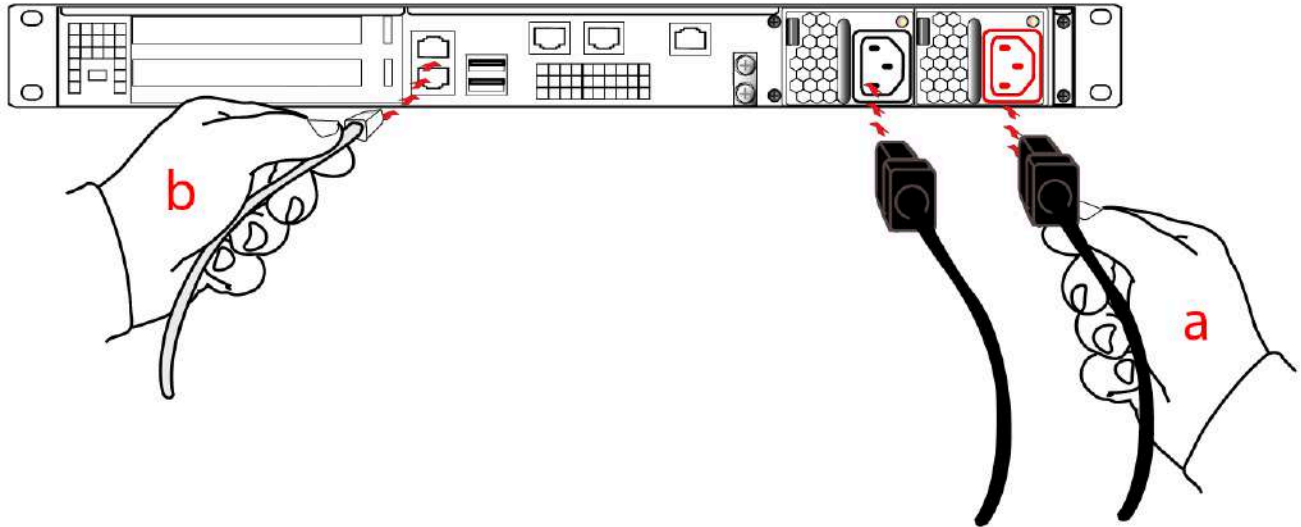
### Connecting network cables

1. Insert the power (a) and network cables (b) at the rear panel.

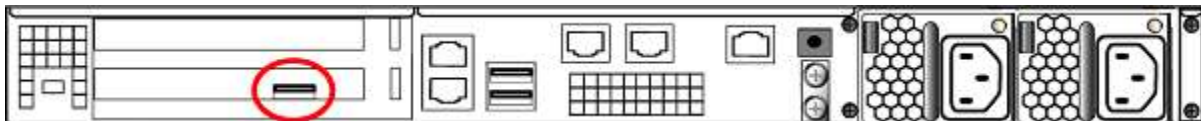
**Note**

The physical location of the network ports (Eth0, Eth1, Eth2 and Eth3) are dependent on the appliance model. Correct locations for your model are printed on the rear panel.

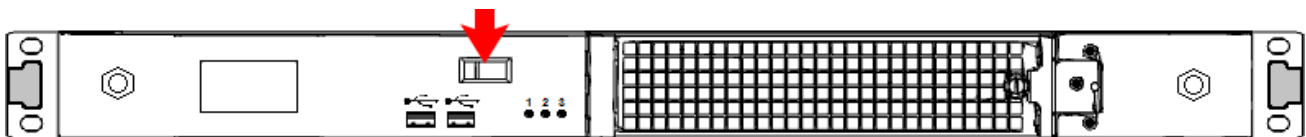
For proper redundancy and best reliability, the power cables should connect to two completely independent power sources.



2. If you have a PED-authenticated k570 appliance, connect the PED directly to the appliance's USB port (on the rear panel's left side), using the included USB-to-MiniUSB PED cable.



3. Press and release the Start/Stop switch on the front panel to power up the appliance.



## Connecting the Appliance to a Console Device for k470 or k570

From the console you log in as the System Administrator ('ksadmin' user), create a secure password, start-up the system and access the IP address of the Graphical User Interface (GUI). You can connect a computer directly to the console port of the CipherTrust Manager Appliance using a serial connection.

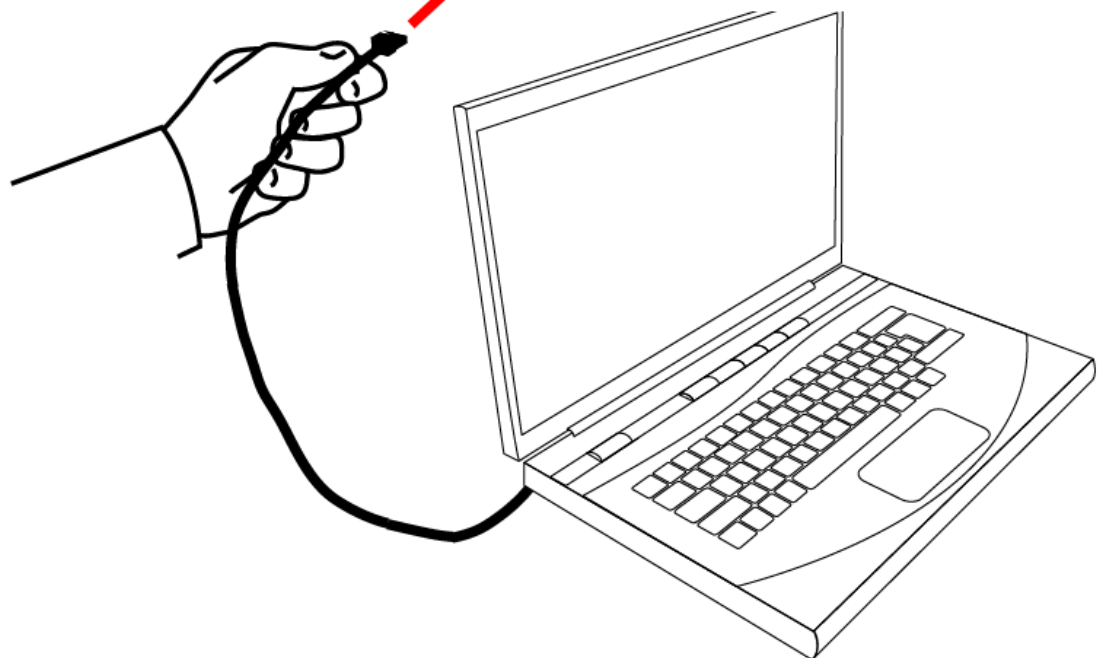
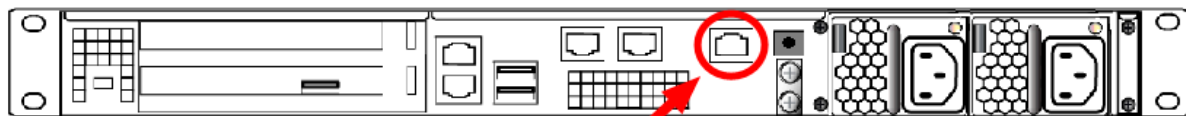
Direct administration connection to the console via serial terminal is required for these reasons:

- During initial configuration, you do not yet know the IP address dynamically allocated by your DHCP server.

- After deployment, if you re-configure network settings (change the IP address) via SSH, you will lose the old IP address connection.

### To open a serial connection

1. Connect the serial port on the appliance's rear panel to a terminal server, dumb terminal, PC, or laptop, using the supplied Prolific Technology Inc. USB to RJ45 (with 8P8C connector) adapter.



2. If the driver for the Prolific Technology Inc. USB to RJ45 (with 8P8C connector) adapter did not download and install automatically, download the [the PL2303 USB-to-Serial Windows driver](#).
3. Open Device Manager (**Control Panel > Hardware > Device Manager**) and expand the **Ports (COM and**

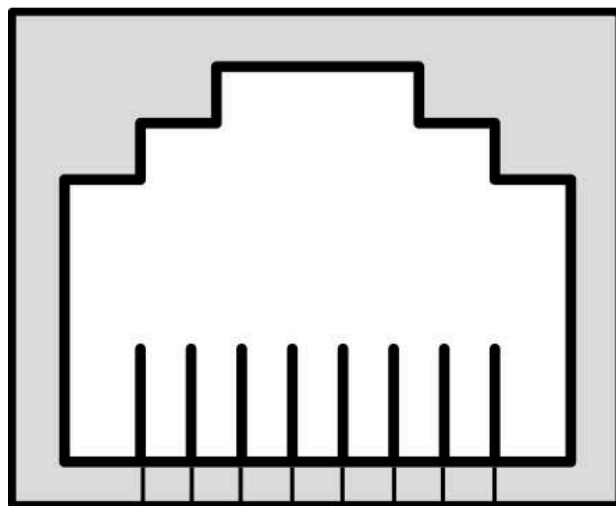
**LPT)** folder. If the driver installed successfully, an entry is displayed for the Prolific USB-to-Serial Comm Port, followed by the port associated with the adapter. For example: Prolific USB-to-Serial Comm Port (COM4)

4. Record the COM port (COM4 in this example) associated with the adapter. You will need this port number when you open a serial connection.
5. Use a terminal emulation package, such as PuTTY, to open a serial connection to the COM port associated with your Prolific USB-to-Serial adapter. Set the serial connection parameters as follows:
  - Baud rate: 19200
  - Data bits: 8
  - Parity: None
  - Stop bits: 1

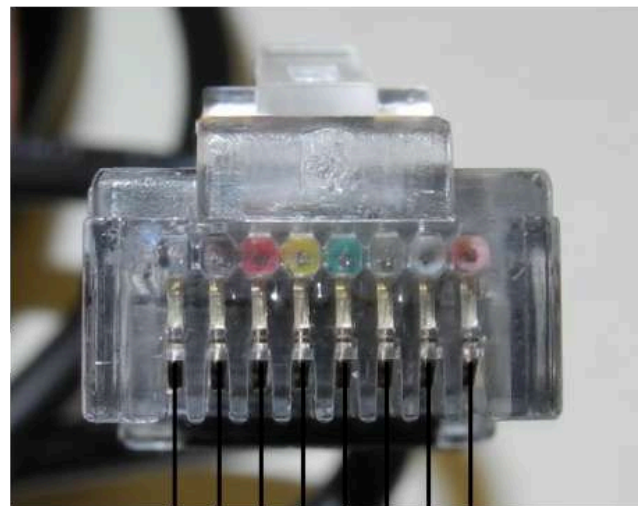
### Serial Port Pinout

The serial port uses a configuration equivalent to the Cisco Terminal Console. The Prolific Technologies Inc. RJ45-to-USB serial adapter cable uses a standard RJ45 pinout configuration:

#### HSM Serial Port



#### RJ45-to-USB serial cable



- 1 RTS
- 2 DTR
- 3 TxD
- 4 GND
- 5 GND
- 6 RxD
- 7 DSR
- 8 CTS

- When the connection is made, the appliance log in prompt is displayed: **ciphertrust login:**

**Note**

You may need to press **ENTER** several times to initiate the session.

**Note**

Windows 10 occasionally crashes when trying to detect a serial port. This is a known issue with the Windows 10 PL2303 drivers. If you experience trouble opening a serial connection using Windows 10, use another supported operating system.

- Type "Ctrl+C" to display the IP address. Retain this value to connect to the GUI for the first time.
- As the System Administrator, enter "ksadmin" to log in and follow the prompts to create a secure password.

**Caution**

Be sure to retain this password - it will be required to access the system in case of network connectivity problems.

The system starts up, which can take several minutes.

## Connecting to the GUI for the First Time

- After the system starts up, in the **Console Window**, choose the KeySecure IP address for your network. Use this address to browse to the CipherTrust Manager GUI.

**Note**

The initial IP address is set via DHCP, and is accessible through the Console Window. If you need to set a static IP address, you can set it from the console using the `nmcli` tool. For details, refer to the [Network Configuration Tutorial](#). `nmcli` can also be used to set static routes, bond network interfaces, disable IPv6, or configure VLAN, if needed for your network.

The initial CipherTrust Manager GUI screen is displayed:



# THALES

## Error



- Following services are starting up: platform, ciphertrust-cloud-key-manager, scheduler, backup, key-vault, crypto, licensing, nae, kmip
- The CipherTrust Manager will be functional after the default SSH public key for the ksadmin user is replaced

## SSH Public Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQBDiLlOH4tInSyWQuJkBLJ2sdce4BsvaFa5eL
6Xh1xgELXArNpjZtWvikQkz3JD/2TZbyzdLVACvfoVIFx2MToOw83WsuxeUua95NW
r1ehqeaS9Bx1
/v5OkogEuTzYN1gXyzQQbfbPEOhIDR+y7FNvlyw5++ZUhi2iMIHQEPeZlhoTYHXG1
RpR438JXoh/C8ey8aaP6cdZcTudQdsdDP3HGWXjsP
/NpOw4yeBthoy5NFuPpmS0Uvd0k2uCYkq6bkW7wwiRbW21joGBEf2muyKiCc2Fa
uVIFvgQNORaj1WjnRcWOjMlfcf2slwue8h2YsAfnzeB/gYhH+gdxfdK8aZ
user@example.local
```

Add

The error displayed is normal and simply requires the default SSH Public Key to be replaced.

- As the System Administrator (ksadmin), paste in your **SSH Public Key** in the box provided and then select **Add**.

## Note

The SSH Public Key must be a 'PEM-formatted RSA key'. You can generate this key using 'PuTTYgen' or similar utility. Save this SSH Public Key at a safe location. You will need this key for future SSH access.

After replacing the default SSH Public Key, the **Log In** screen appears. For more options to replace the default SSH Public Key, see [Starting Services After Deployment](#)

### Warning

Be sure to store and securely protect the associated private SSH key, as this key will be required to SSH to the appliance from this point on.

The initial Application Administrator can now log in.

3. Log in using the initial default credentials: Username = admin, Password = admin

The following notice is displayed:



### Password Change

After successfully changing your password, you will be logged out and will have to log in again

### Note

If the default credentials do not work, you may need to retrieve an autogenerated password, as described in [Changing the Initial Password](#).

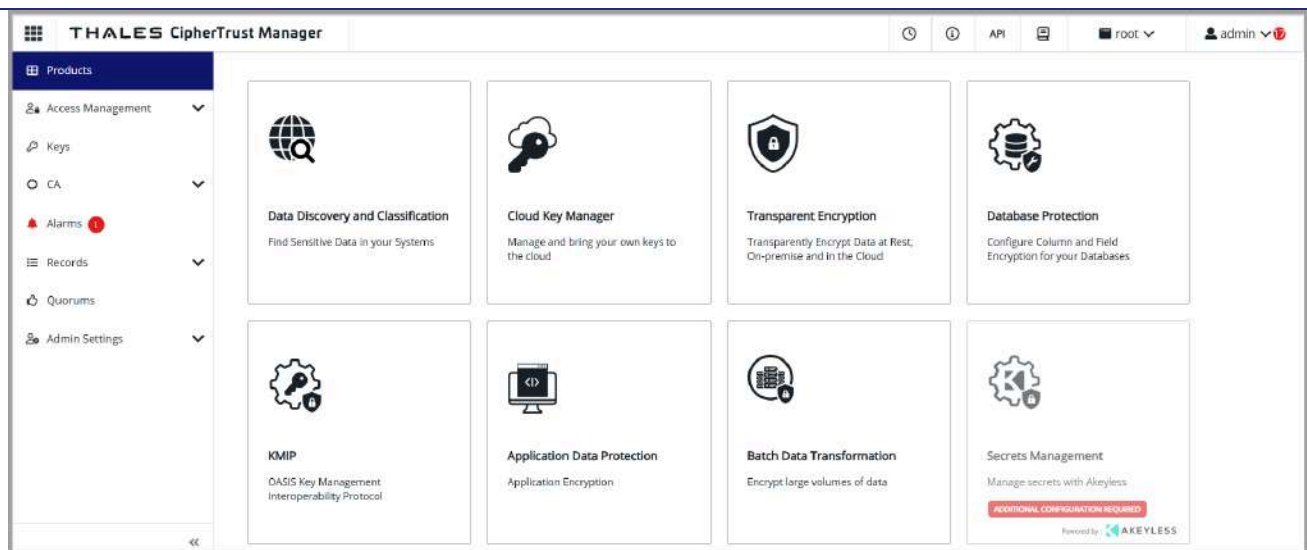
4. Enter a new password using this default Password Policy:

```
Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1
```

A new **Login** screen appears.

5. Using your new password, log in again. The **CipherTrust Manager** Web Page appears.





6. At this point, it's strongly recommended to configure an NTP server.
  - a. Navigate to **Admin Settings > NTP**.
  - b. Enter in an **NTP Server hostname**.
  - c. For an authenticated NTP Server, enter in a symmetric key value in the **NTP Key** field.
  - d. Click the **+Add NTP Server** button.

See [Network Time Protocol Server Configuration](#) for more details.

## Installing the Locking Bezel for k570 or k470

For maximum physical access security, after you have powered on the appliance, fit the locking bezel over the appliance's faceplate. Certain security standards require the use of these physical access measures. The locks fit over the posts highlighted below.

Turn the keys to the vertical position to lock the bezel. The keys cannot be removed if the bezel is unlocked. The two locks are keyed differently, so the keys can be issued to different security personnel and kept in secure, separate locations.

### Note

Leaving the keys in the bezel could interfere with closing the rack door, and compromise security.

## Local HSM Configuration for k570

HSM configuration must be performed for k570 devices before beginning to create keys or engage in any other cryptographic usecases.

For PED-authenticated HSMs, you can also [configure the HSM remotely](#). That configuration requires remote PED

access from another site.

## Thales k570 Configuration

The Thales CipherTrust Manager k570 differs from the [Thales Cyber Technologies \(TCT\) CipherTrust Manager k570](#) in the HSM initialization steps. To confirm that you have a Thales CipherTrust Manager k570, you may check the front bezel label, view the order summary, or examine the terminal response for the initial LunaCM command.

### Tip

Retain the 'partition-label' and the 'partition challenge' created during the HSM initialization procedure for later root-of-trust configuration.

### Note

If you are using PED authentication, and using the multifactor quorum feature (previously known as M of N) to split secrets across multiple PED keys, the default PED timeout value might not be long enough to imprint all PED keys. If you notice the error `CKR_Timeout`, [contact client services](#) to increase the PED timeout values through the [Support CLI](#).

1. As the System Administrator (ksadmin) SSH in to the appliance (or connect via serial port using your password) and execute `"/usr/safenet/lunaclient/bin/lunacm"` utility.

The displayed `Model` should be `Luna K7`. If the `Model` value is `Luna T7` follow the [configuration for the Trusted Cyber Technologies k570 appliance](#).

The utility displays information on the detected HSM card and allows you to execute various HSM management commands.

Refer to the [Luna PCIe HSM documentation](#) for more details on these HSM commands.

2. Make sure an HSM admin slot is selected.

To see the available slots, enter:

```
lunacm:> slot list
```

Look for a slot with description "Admin Token Slot".

To select the active slot, enter:

```
lunacm:> slot set -slot <number>
```

---

### 3. Re-initialize the HSM.

```
lunacm:> hsm factoryReset
lunacm:> hsm init -label <admin token slot label>
```

#### Note

At this point, you can use `slot list` to see that the slot with description "Admin Token Slot" now has a label.

### 4. Initialize the Security Officer (SO) role:

```
lunacm:> role login -n so
```

For PED-authenticated devices, you are asked to present a blue HSM SO key to the PED for the SO `role login`. You can re-use an existing key or imprint a new key. Prompts on the PED screen guide you through these options.

### 5. Decide if you wish to operate the HSM in a mode compliant with the Federal Information Processing Standard (FIPS) 140-2. By default the HSM is **not** in FIPS compliant mode.

If you don't need this mode, skip to the next step.

If you do need this mode, change the HSM policy 12 to off. Refer to [Luna PCIe documentation](#) or contact [customer support](#) for more details.

```
lunacm:> hsm changehsmpolicy -policy 12 -value 0
```

#### Warning

This policy is destructive, meaning that HSM partitions and root keys are deleted when the policy is changed. It is strongly recommended that you decide on this configuration now, and do not change it later. If you wish to change the policy after creating user keys, you must backup your user keys in a cluster or risk losing access to user keys. Contact [customer support](#) for guidance for changing FIPS mode after the initial setup.

### 6. Create the first partition:

```
lunacm:> partition create
lunacm:> slot list
```

Notice the slot with the slot description "User Token Slot". Remember the slot ID of this slot as this will be used in the next step.

```
lunacm:> role logout
```

7. Initialize partition and the partition SO role:

```
lunacm:> slot set -slot <slot number of user token slot created above>
lunacm:> partition init -label <new partition label>
```

8. If your k570 is password authenticated, skip this step. If your k570 is PED-authenticated:
  - a. Respond to PED prompts to create the partition. You create PED keys for the partition SO token (blue) and the partition cloning domain token (red).
  - b. As the Partition SO, activate the partition.

#### Note

This instructs the HSM to cache PED credentials and allows the k570 appliance to authenticate to the HSM using only the challenge secret (password) without requiring the black PED key to always be connected to the HSM. However, in the event of a power outage of more than 2 hours, the HSM cached PED credentials will expire and the k570 appliance will fail to run its services. In this case, reboot the CipherTrust Manager, disconnect and reconnect the PED USB cable, and re-authenticate with the HSM using the black PED key when prompted. With auto-activation enabled, the HSM re-caches the PED credentials.. You can also [configure remote PED access](#).

```
lunacm:> role login -name Partition SO lunacm:> partition changepolicy -policy 22 -value 1
lunacm:> partition changepolicy -policy 23 -value 1
```

9. As the Partition SO, initialize the Crypto officer role:

- a. Enter the command to initialize.

```
lunacm:> role init -name Crypto Officer
```

- b. Respond to prompts on the terminal and PED to create the initial Crypto Officer credential.

**Caution**

The Crypto Officer PED key or password is valid for the initial login only. The Crypto Officer must change this initial credential using the command `role changepw` immediately. Failing to change the credential results in a `CKR_PIN_EXPIRED` error when accessing the partition.

- c. If using PED authentication, create an initial Crypto Officer challenge secret. As with the PED key, it is valid for the first Crypto Officer login only and must be changed immediately.

```
lunacm:>role createchallenge -name Crypto Officer
```

10. Reset the Crypto Officer's credentials.

- a. Log in the Crypto Officer. When prompted for the password, provide the initial password (password authentication) or challenge secret (PED authentication).

```
lunacm:> role login -name Crypto Officer
```

- b. Run the following command, which will reset the Crypto Officer PED key secret or initial password. Respond to the PED and terminal prompts.

```
lunacm:> role changePw -name Crypto Officer
```

- c. For PED authenticated HSM, change the initial challenge password. The passwords are not masked.

```
lunacm:> role changePw -name Crypto Officer -old <existing challenge secret> -newpw <new challenge secret>
```

- d. Log in again to activate/cache the new Crypto Officer credentials.

```
lunacm:> role login -name Crypto Officer
```

11. Exit the `lunacm` utility.
12. Proceed to [HSM root of trust configuration](#).

## Trusted Cyber Technologies k570 Local HSM Configuration

The Trusted Cyber Technologies CipherTrust Manager k570's HSM has different steps to initialize than the Thales CipherTrust Manager k570 appliance's HSM.

This appliance has the label "Trusted Cyber Technologies CipherTrust k570" on the front bezel and on the order summary. In addition, the initial LunaCM response contains the SafeNet Assured Technologies, LLC and indicates an HSM Model of Luna-T7.

You can configure the HSM to use [password authentication](#) or [PED authentication](#).

## Configure for Password Authentication

1. As the System Administrator (ksadmin) SSH in to the appliance (or connect via serial port using your password) and execute `/usr/safenet/lunaclient/bin/lunacm` utility. The initial response looks like the following for TCT k570s, including the SafeNet Assured Technologies, LLC heading, and the Luna-T7 model:

```
lunacm (64-bit) v7.11.1-5 (7.11.1-5-ga24a9e8). Copyright (c) 2020 SafeNet Assured Technologies, LLC. All rights reserved.
```

```
Available HSMs:
```

```
Slot Id -> 0
HSM Label -> no label
HSM Serial Number -> XXXXXX
HSM Model -> Luna-T7
HSM Firmware Version -> 7.11.0
HSM Configuration -> Luna PCI (PED) Undefined Mode / Uninitialized
HSM Status -> Zeroized
```

```
Current Slot Id: 0
```

If the LunaCM response does not include these elements, you likely have a [Thales CipherTrust Manager k570 device](#)

The utility displays information on the detected HSM card and allows you to execute various HSM management commands.

Refer to the latest *Luna T-Series Documentation* for more details on these HSM commands. To access this documentation, login to the [TCT Customer Support Portal](#), and navigate to **Knowledge Base > Luna T-series**.

2. Re-initialize the HSM. You must set an HSM label, a password for the HSM Security Officer, and a string for the Security Officer domain.

The Security Officer password is needed for the initial HSM configuration.

Retain the HSM label value. This value is needed later to set the HSM as [root of trust](#).

```
lunacm:> hsm init -initwithpwd -label <HSM label> -password <Security Officer
```

```
password> -domain <Security Officer domain name>
```

3. Login to the HSM with the security officer password.

```
lunacm:> hsm login -password <Security Officer password>
```

4. Decide if you wish to operate the HSM in a mode compliant with the Federal Information Processing Standard (FIPS) 140-2. By default the HSM is **not** in FIPS compliant mode.

If you don't need this mode, skip to the next step.

If you do need this mode, change the HSM policy 12 to off. Refer to the latest *Luna T-Series HSM documentation* for more details. To access this documentation, login to the [TCT Customer Support Portal](#), and navigate to **Knowledge Base > Luna T-series**.

```
lunacm:> hsm changehsmpolicy -policy 12 -value 0
```

### Warning

This policy is destructive, meaning that HSM partitions and root keys are deleted when the policy is changed. It is strongly recommended that you decide on this configuration now, and do not change it later. If you wish to change the policy after creating user keys, you must backup your user keys in a cluster or risk losing access to user keys. Contact [customer support](#) for guidance for changing FIPS mode after the initial setup.

5. Log back into the HSM as the security officer.

```
lunacm:> hsm login -password <security officer password>
```

6. Create the partition. Set a partition password and domain name. Retain the partition password. This password is needed later for [HSM root of trust setup](#).

```
lunacm:> partition create -password <partition password> -domain <domain name>
```

7. Exit the `lunacm` utility.
8. Proceed to [HSM root of trust configuration](#).

## Configure for PED Authentication

1. As the System Administrator (ksadmin) SSH in to the appliance (or connect via serial port using your password) and execute `"/usr/safenet/lunaclient/bin/lunacm"` utility. The initial response looks like the following for TCT k570s, including the `SafeNet Assured Technologies, LLC` heading, and the `Luna-T7` model:

```
lunacm (64-bit) v7.11.1-5 (7.11.1-5-ga24a9e8). Copyright (c) 2020 SafeNet Assured Technologies, LLC. All rights reserved.
```

```
Available HSMs:
```

```
Slot Id -> 0
HSM Label -> no label
HSM Serial Number -> XXXXXX
HSM Model -> Luna-T7
HSM Firmware Version -> 7.11.0
HSM Configuration -> Luna PCI (PED) Undefined Mode / Uninitialized
HSM Status -> Zeroized
```

```
Current Slot Id: 0
```

If the LunaCM response does not include these elements, you likely have a [Thales k570 device](#)

The utility displays information on the detected HSM card and allows you to execute various HSM management commands.

Refer to the latest *Luna T-Series Documentation* for more details on these HSM commands. To access this documentation, login to the [TCT Customer Support Portal](#), and navigate to **Knowledge Base > Luna T-series**.

2. Initialize the HSM to allow for PED authentication. Provide an HSM label, which is used for root of trust setup.

These values are temporary and needed for the initial setup.

```
lunacm:> hsm init -initwithped -label <HSM label>
```

You are prompted to create a Security Officer (blue) iKey on the PED.

3. Respond to PED prompts to create the Domain (red) iKey.
4. Decide if you wish to operate the HSM in a mode compliant with the Federal Information Processing Standard (FIPS) 140-2. By default the HSM is **not** in FIPS compliant mode.

If you don't need this mode, skip to the next step.



---

If you do need this mode, change the HSM policy 12 to off.

Refer to the latest *Luna T-Series Documentation* for more details. To access this documentation, login to the [TCT Customer Support Portal](#), and navigate to **Knowledge Base > Luna T-series**.

```
lunacm:> hsm changehsmpolicy -policy 12 -value 0
```

### Warning

This policy is destructive, meaning that HSM partitions and root keys are deleted when the policy is changed. It is strongly recommended that you decide on this configuration now, and do not change it later. If you wish to change the policy after creating user keys, you must backup your user keys in a cluster or risk losing access to user keys. Contact [customer support](#) for guidance for changing FIPS mode after the initial setup.

5. Login to the HSM as the SO using the Security Officer (blue) Key.

```
lunacm:> hsm login
```

6. Create a partition.

```
lunacm:>partition create
```

- a. Respond to the PED prompts to insert USER/Partition Owner (black) iKey and set a PIN.
- b. Respond to the PED prompts to insert Domain (red) iKey and set its PIN.

7. Login to the new partition.

```
lunacm:>partition login
```

You are prompted to re-insert the USER (black) iKey.

8. Request the challenge password from the PED.

```
lunacm:> partition createChallenge
```

The Login Secret Value is displayed on the PED. Take note of this value. This is the partition password which is needed for [root of trust setup](#).

9. Activate the partition.

```
lunacm:>partition changepolicy -policy 22 -value 1
```

**Note**

This instructs the HSM to cache PED credentials and allows the k570 appliance to authenticate to the HSM using only the challenge secret (password) without requiring the black PED key to always be connected to the HSM.

10. Changing policies causes the Security Officer to be logged out. To log back in:

```
lunacm:>hsm login
```

Present the Security Officer (blue) iKey when prompted.

11. Change the partition policy to allow auto-activation.

```
lunacm:>partition changepolicy -policy 23 -value 1
```

**Note**

This instructs the HSM to cache PED credentials for up to two hours after loss of power. After two hours, the HSM cached PED credentials expires and the k570 appliance fails to run its services. In this case, reboot the CipherTrust Manager, disconnect and reconnect the PED USB cable, and re-authenticate with the HSM using the black PED key when prompted. With auto-activation enabled, the HSM re-caches the PED credentials. You can also configure [remote PED access](#).

12. Changing policies causes the Security Officer to be logged out. To log back in:

```
lunacm:>hsm login
```

Present the Security Officer (blue) iKey when prompted.

13. Exit the lunaCM utility.
14. Proceed to [HSM root of trust configuration](#).

## HSM Root of Trust Configuration

### Caution

If you intend to restore root of trust keys from a Backup HSM, restore the keys **before** performing root of trust configuration.

Configuring root of trust and then restoring root of trust keys can result in the CipherTrust Manager application becoming unavailable after reboot, requiring customer support to recover.

Configure the k570 appliance to use the Luna PCIe HSM card using these steps:

1. Access the 'partition-label' and the 'partition challenge' created during the HSM initialization procedure. These are needed to use the CipherTrust Manager HSM API or CLI `hsm setup` command.
2. You are now ready to configure the HSM as Root of Trust. Refer to CipherTrust Manager HSM Setup API or the CLI `hsm setup` command to configure the appliance to use your newly initialized Luna PCIe HSM. [Root of Trust Configuration](#) provides more details.

## Remote PED Configuration

After initial configuration, you can set up remote PED access for the k570. In that way, you do not have to visit the data center to perform lunaCM and PED operations in the future. You might have to perform PED operations after setup to recover from power outages, or for rare, advanced troubleshooting scenarios.

1. Determine if you have a Thales k570 appliance or a Trusted Cyber Technologies k570 appliance.

The Trusted Cyber Technologies k570 appliance has the label "Trusted Cyber Technologies CipherTrust k570" on the front bezel and on the order summary. In addition, the initial LunaCM response contains the `SafeNet Assured Technologies, LLC` and indicates an HSM Model of `Luna-T7`.

2. As the System Administrator (ksadmin) SSH in to the appliance (or connect via serial port using your password) and execute `/usr/safenet/lunaclient/bin/lunacm` utility.

The displayed `Model` should be `Luna K7` for Thales k570 device, `Luna T7` for the Trusted Cyber Technologies k570 appliance.

3. For a Thales k570 appliance:

- a. View the available slots with `slot list`.
- b. Select the slot with the label "Admin Token Slot".

```
lunacm:>slot set -slot <slot_id_number>
```

- c. Login with the Partition Security Officer role. On the PED, you are prompted to present the Blue Partition

---

Security Officer key and enter its PIN.

```
lunacm:>role login -name so
```

4. For a Trusted Cyber Technologies k570 appliance, login with the Security Officer role. On the PED, you are prompted to present the Blue Partition Security Officer key and enter its PIN.

```
lunacm:>hsm login
```

5. Initialize the orange remote PED vector key. You are prompted to insert a PED key and follow PED prompts. You may create a new key or re-use an existing orange key.

```
lunacm:>ped vector -init
```

6. Prepare a workstation to act as a remote PED server to access the k570 in the data center.

#### Note

More details for remote PED server set up are available in [Luna HSM Documentation](#).

- a. Install Luna Client with remote PED as an option. Luna Client version 7.0.1 or above is required for the Thales k570 appliance. Thales TCT Luna Client version 7.1.1 or above is required for the Trusted Cyber Technologies 570 appliance.
- b. Connect the remote PED to the workstation via USB, and to the power source via the power adapter.
- c. Open a Command Prompt window on the computer (for Windows 7, this must be an Administrator Command Prompt). Locate and run PedServer.exe. Set PedServer.exe to its "listening" mode.

```
c: > PedServer -m start
Ped Server Version 1.0.6 (10006)
Ped Server launched in startup mode.
Starting background process
Background process started
Ped Server Process created, exiting this process.
c:\PED\ >
```

- d. Verify that the service has started with `pedserver -mode show`.

Look for mention of the default Server Port "1503" (or other, if you specified a different listening port). In addition, "Ped2 Connection Status:" should say "Connected." This indicates that the Luna PED that you connected was found by PED Server.

---

## Example Output

```
PedServer.exe -m show
Ped Server Version 1.0.6 (10006)
Ped Server launched in status mode.

Server Information:
 Hostname: host
 IP: 0.0.0.0
 Firmware Version: 2.9.0-2
 PedII Protocol Version: 1.0.1-0
 Software Version: 1.0.6 (10006)

 Ped2 Connection Status: Connected
 Ped2 RPK Count 0
 Ped2 RPK Serial Numbers (none)

Client Information: Not Available

Operating Information:
 Server Port: 1503
 External Server Interface: Yes
 Admin Port: 1502
 External Admin Interface: No
 PED Write Delay: 0 (microsecs)

 Server Up Time: 223 (secs)
 Server Idle Time: 158 (secs) (70%)
 Idle Timeout Value: 1800 (secs)

 Current Connection Time: 0 (secs)
 Current Connection Idle Time: 0 (secs)
 Current Connection Total Idle Time: 0 (secs) (100%)
 Total Connection Time: 122 (secs)
 Total Connection Idle Time: 62 (secs) (50%)

Show command passed.
```

- e. SSH to the k570 and open LunaCM.

```
ssh -I <key> ksadmin@<IP>
/usr/safenet/lunaclient/bin/lunacm
```

- f. Start the PED client on the k570.

```
ped connect -ip <remote_PED_workstation_IP> -port 1503
```

## Securely Transporting a CipherTrust Manager k570 Appliance

You might need to move your configured CipherTrust Manager appliance to a new data center or place it into storage. This can be a convenient way to pre-configure CipherTrust Manager appliances at a central location before shipping them to regional data centers.

During this process, you can ensure that the on-board PCIe HSM and its cryptographic material have not been modified in transit by using the Secure Transport Mode (STM). STM temporarily locks the HSM, retaining the current configuration and key material, and recording the current state.

The commands and supported authentication method for STM are different for a Thales CipherTrust Manager k570 appliance and a Trusted Cyber Technologies (TCT) k570 appliance. The TCT k570 appliance has the label "Trusted Cyber Technologies CipherTrust k570" on the front bezel and on the order summary. In addition, the initial LunaCM response contains the `SafeNet Assured Technologies, LLC` and indicates an HSM Model of `Luna-T7`.

## Securely Transport a Thales k570 Appliance

Thales k570 appliances can access STM. This feature is described in detail in [Luna PCIe HSM documentation](#).

STM generates a unique 16-character verification string and a 16-character random user string. These unique strings allow you to verify whether or not the HSM has been tampered while in STM. When recovering from STM, you will be asked to provide the random user string.

### To prepare the CipherTrust Manager for transportation

1. Log in to the CipherTrust Manager as `ksadmin` via serial console or SSH.
2. Stop CipherTrust Manager services. This avoids the services trying to communicate with the HSM while STM is applied.

```
ksadmin@cipherttrust:~$ sudo systemctl stop keysecure
```

3. Launch the `lunacm` utility from `/usr/safenet/lunaclient/bin/`

```
ksadmin@keysecure:/usr/safenet/lunaclient/bin$./lunacm
lunacm (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.

Available HSMs:

Slot Id -> 3
Label -> kspart1
Serial Number -> 1442956002008
Model -> Luna K7
Firmware Version -> 7.0.1
```

```

Configuration -> Luna User Partition With SO (PW) Signing With Cloning
Mode
Slot Description -> User Token Slot

Slot Id -> 8
Label -> kspart1
Serial Number -> 619745
Model -> Luna K7
Firmware Version -> 7.0.1
Configuration -> Luna HSM Admin Partition (PW) Signing With Cloning Mod
e
Slot Description -> Admin Token Slot
HSM Configuration -> Luna HSM Admin Partition (PW)
HSM Status -> L3 Device

Current Slot Id: 3

```

4. If you have a PED-authenticated k570, you must deactivate the Crypto Officer role on the user partition. If you have a password authenticated k570, skip ahead to step 5.

### Warning

This command is needed to ensure the verification string is generated correctly on recovery. Leaving the Crypto Officer in an auto-activated state will result in a verification string mismatch.

Activated Crypto Officers display the phrase Activated with the command `lunacm:>role show -n CO` if you need to confirm the Crypto Officer's state.

- a. Switch to the slot number labeled with `Luna User Partition` in the above output. This is slot 3 in our example configuration.

```
lunacm:>slot set -slot <user_partition_slot_number>
```

- b. Log in as the Partition Security Officer. You are prompted to present the Partition SO PED iKey.

```
lunacm:>role login -n po
```

- c. Deactivate the Crypto Officer role.

```
lunacm:>role deactivate -n Crypto Officer
```

5. Backup the HSM contents, as described in [Luna PCI documentation](#).
6. Switch to the slot number labeled with `Luna Admin Partition`. This is slot 8 in our example configuration.

```
lunacm:>slot set -slot <admin_partition_slot_number>
```

7. Log in as the HSM Security Officer. You are prompted to enter the HSM SO password or to present the HSM SO PED key.

```
lunacm:>role login -n so
```

8. Set the transport mode.

```
lunacm:>stm transport

You are about to configure the HSM in STM.
Are you sure you wish to continue?

Type 'proceed' to continue, or 'quit' to quit now ->proceed

Configuring the HSM for transport (may take a few seconds)...

HSM was successfully configured for transport.

Please record the displayed verification & random user strings.
These are required to recover from Secure Transport Mode.

Verification String: AAAA-AAAA-AAAA-AAAA

Random User String: BBBB-BBBB-BBBB-BBBB
```

9. Record the presented Verification String and Random User String. Both are needed to recover the HSM and CipherTrust Manager.
10. Power down the appliance, unrack it, and ship it to its destination.
11. Transmit the verification string and random user string to the receiver of the HSM using a secure method, distinct from the transport of the physical HSM, so that it is not possible for an attacker to have access to both the HSM and the verification codes while the HSM is in STM.



## To recover the CipherTrust Manager after transportation

### Note

You can use [Remote PED](#), if previously configured for the k570, to perform necessary PED operations outside the data center.

1. Ensure you have the two strings that were presented when the HSM was placed into STM.
2. [Place the HSM into a rack](#) and set up [networking](#). You need SSH or console access to recover the HSM.

### Note

You don't need to change the ksadmin password, or re-configure the HSM, as with a first installation.

3. Login to the CipherTrust Manager as ksadmin via SSH or serial console.
4. Launch the lunacm utility from `/usr/safenet/lunaclient/bin/`

```
ksadmin@keysecure:/usr/safenet/lunaclient/bin$./lunacm
lunacm (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.

Available HSMs:

Slot Id -> 3
Label -> kspart1
Serial Number -> 1442956002008
Model -> Luna K7
Firmware Version -> 7.0.1
Configuration -> Luna User Partition With SO (PW) Signing With Cloning
Mode
Slot Description -> User Token Slot

Slot Id -> 8
Label -> kspart1
Serial Number -> 619745
Model -> Luna K7
Firmware Version -> 7.0.1
Configuration -> Luna HSM Admin Partition (PW) Signing With Cloning Mod
e
Slot Description -> Admin Token Slot
```

```
HSM Configuration -> Luna HSM Admin Partition (PW)
HSM Status -> L3 Device

Current Slot Id: 3
```

- Switch to the slot number labeled with `Luna HSM Admin Partition` in the above output. This is slot 8 in our example configuration.

```
lunacm:>slot set -slot <admin_partition_slot_number>
```

- If you are using a Remote PED, connect the slot to the Remote PED workstation. Otherwise, skip to the next step.

```
lunacm:>ped connect -ip <remote_PED_workstation_IP> -port 1503
```

- Login as the HSM Security Officer. You are prompted to enter the HSM SO password or to present the HSM SO PED key. Remote PEDs also prompt for the orange remote PED vector key.

```
lunacm:>role login -n so
```

- Run the `stm recover` command, providing the Random User String.

```
lunacm:>stm recover -r BBBB-BBBB-BBBB-BBBB
```

You are presented with the verification string.

- If the presented verification string matches the original verification string, type `proceed` to continue.

### Caution

If the presented verification string does not match the original verification string, this indicates HSM tampering. Type `quit` to remain in secure transport mode. Contact [Customer Support](#) to investigate.

Calculating the verification string (may take a few seconds)...

Verification String: AAAA-AAAA-AAAA-AAAA

CAUTION: You are attempting to recover the HSM from Secure Transport Mode. If the verification string does not match the one you were provided out-of-band, there may be an issue with the HSM. Type 'quit' at the prompt to remain in Secure Transport Mode.

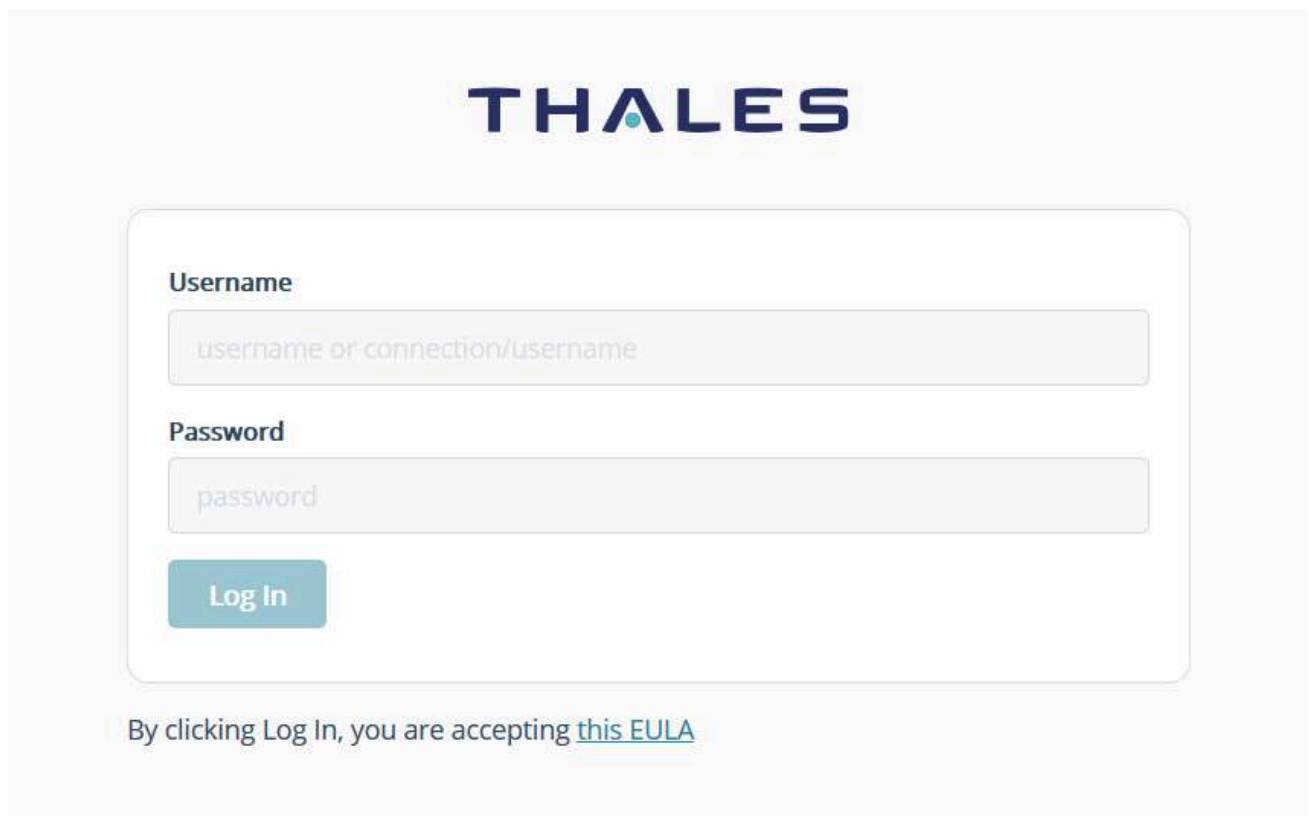
```
If the verification strings match, or if you wish to bypass this check,
type 'proceed' to recover from Secure Transport Mode.
```

Are you sure you wish to continue?

Type 'proceed' to continue, or 'quit' to quit now ->proceed

The CipherTrust Manager services boot up.

For password-authenticated k570s, recovery is complete. You can now login to the CipherTrust Manager GUI at the configured IP address, and access your keys.

The image shows the login interface of the Thales CipherTrust Manager. At the top, the 'THALES' logo is displayed in a dark blue, sans-serif font. Below the logo is a light gray rounded rectangle containing the login fields. The first field is labeled 'Username' in bold, with a placeholder text 'username or connection/username'. The second field is labeled 'Password' in bold, with a placeholder text 'password'. Below these fields is a teal-colored button with the text 'Log In' in white. At the bottom of the login area, there is a line of text: 'By clicking Log In, you are accepting [this EULA](#)'.

PED-authenticated k570s must have the Crypto Officer role logged in before you can access CipherTrust Manager services.

### To login and reactivate the Crypto Officer Role for PED-Authenticated Devices

The Crypto Officer must be logged in before you can access CipherTrust Manager services.

## Note

This login re-activates the Crypto Officer role (unless you have disabled the activation partition policies, 22 and 23, which is not recommended). The HSM caches PED credentials and allows the k570 appliance to authenticate to the HSM using only the challenge secret (password) without requiring the black PED key to always be connected to the HSM. However, in the event of a power outage of more than 2 hours, the HSM cached PED credentials will expire and the k570 appliance will fail to run its services. In this case, instruct the k570 appliance to re-authenticate with the HSM using the black PED key. You can also [configure remote PED access](#).

1. If you are using a local PED, simply present the black PED iKey for the Crypto Officer to the PED, as prompted on the PED screen when CipherTrust Manager services start up.
2. If you are using a Remote PED:
  - a. Switch to the slot number labeled with `Luna User Partition`. This is slot 3 in our example configuration.

```
lunacm:>slot set -slot <user_partition_slot_number>
```

- b. Connect the slot to the Remote PED workstation.

```
lunacm:>ped connect -ip <remote-PED-workstation-IP> -port 1503
```

- c. Log in as the Crypto Officer. Provide the PED iKey, and the auto-activation password as prompted.

```
lunacm:>role login -n co
```

3. You can now login to the CipherTrust Manager GUI at the configured IP address, and access your keys.



**Username**

**Password**

**Log In**

By clicking Log In, you are accepting [this EULA](#)

## Securely Transport a Trusted Cyber Technologies (TCT) k570 Appliance

TCT k570 appliances that use PED authentication can apply Secure Transport Mode.

As part of Secure Transport Mode, you imprint one or more purple PED keys (also called Secure Recovery Keys or SRK) with a new split of the HSM's master key. You are also presented with a unique SRV verification string, which allows you to verify whether or not the HSM has been tampered while in STM. The purple PED key and verification string are required to recover the HSM and should be stored and shipped separately from the CipherTrust Manager appliance.

Advanced options for re-using an existing SRK, creating more copies of the PED key, and the MofN feature are described in the latest *Luna T-Series Documentation*. To access this documentation, login to the [TCT Customer Support Portal](#), and navigate to **Knowledge Base > Luna T-series**.

### To prepare the CipherTrust Manager for transportation

1. Connect the PED to the appliance.
2. Login to the CipherTrust Manager as `ksadmin` via serial console or SSH.
3. Launch the lunacm utility from `/usr/safenet/lunaclient/bin/`
4. Login to the HSM with `hsm login`.
5. Run the command `srk enable`. This command performs a resplit of the MTK before moving one of the (new) splits out to your purple PED Key(s).

6. Follow the PED prompts, introducing the purple key (\*) and pressing buttons on the PED keypad.
7. Record the SRV verification string when it is presented. This string is required to recover both the HSM and CipherTrust Manager.
8. Run `srk transport` to enable secure transport mode.  
  
You are prompted to insert the purple PED key.
9. Power down the appliance, unrack it, and ship it to its destination.
10. Ship the imprinted purple SRK separately from the appliance shipment.
11. Send the SRK verification string via another path.

### To recover the CipherTrust Manager after transportation

1. Ensure you have the purple PED key and the SRK Verification String that were presented when the HSM was placed into STM.
2. Place the HSM into a rack and set up [networking](#). You need SSH or console access to recover the HSM.

#### Note

You don't need to change the ksadmin password, or re-configure the HSM, as with a first installation.

3. Login to the CipherTrust Manager as ksadmin.

```
ssh -i priv_key.pem ksadmin@<IP_address>
```

4. Launch the lunacm utility from `/usr/safenet/lunaclient/bin/`
5. Run the command `srk recover`.

The PED prompts for the SRK (purple PED Key) and shows the verification string.

6. Compare the presented verification string with the original verification string. If they match, insert the purple PED key as prompted. If the strings do not match, the unit may have been tampered. Follow [tamper recovery instructions](#).
7. Run the command `hsm login`.

The PED prompts for the HSM/Security Officer blue key.

8. Run the command `partition login`.

---

The PED prompts for the User (black) PED key.

Recovery is complete. You can now login to the CipherTrust Manager GUI at the configured IP address, and access your keys.

The image shows the login interface of the Thales CipherTrust Manager. At the top, the Thales logo is displayed in a dark blue font. Below the logo is a login form with a light gray border. Inside the form, there are two input fields: the first is labeled 'Username' and contains the placeholder text 'username or connection/username'; the second is labeled 'Password' and contains the placeholder text 'password'. Below these fields is a teal-colored button with the text 'Log In' in white. At the bottom of the form, there is a line of text: 'By clicking Log In, you are accepting [this EULA](#)'.

## Remote PED-Authenticated k570 Initialization

If you wish to centralize PED (PIN entry Device) key access for k570 appliances from a headquarters, you can have your data center technicians initialize k570s with temporary PED keys. You can then establish a remote PED connection, and re-set the PED keys from a remote location.

### Note

This page applies to the Thales k570 CipherTrust Manager appliance. For remote PED-authenticated initialization for the Trusted Cyber Technologies CipherTrust k570 appliance, refer to the latest *Luna T-Series Documentation*. To access this documentation, login to the [TCT Customer Support Portal](#). The *TCT k570 Remote PED – Multi-locations Tech Note* that covers this setup may be found in **Knowledge Base > CipherTrust > TechNotes**.

## Planning Number of PED keys

You will need to ship duplicate HSM SO (blue), HSM domain (red), and remote PED vector (orange) PED keys to the data center to establish the initial remote connection.

You can then reset the HSM SO blue key, and remote PED vector orange keys remotely for more long-term k570 HSM access. You can create new versions of these PED keys for each k570 appliance or reuse the same set of keys for multiple appliances.

For each HSM, you remotely create a partition to act as a root of trust. This partition requires a Partition SO (blue) PED key, a partition cloning domain (red) PED key and a Crypto Officer (black) PED key. You can create new versions of the blue and red partition PED keys for each k570 appliance or reuse the same set of keys for multiple HSMs.

All PED key roles have options for [MofN quorum controls](#), so you may split the key secret across multiple PED keys if desired.

## Prepare keys and a remote PED server setup at a central location

1. Fully **install** a k570 appliance in a central location, including HSM initialization and PED key imprinting. The PED prompts give options to duplicate blue HSM Security Officer and red Cloning Domain PED keys during initialization.
2. If you haven't already, duplicate the blue HSM Security Officer PED key and red Cloning Domain PED key. Create one blue key and one red key for each data center location. The PED need only be connected to outlet power. It does not need to be connected to the k570 appliance.

### Note

More options for key duplication are described in [Luna HSM documentation](#).

- a. On the Luna PED, insert the desired key and press < to navigate to the main menu. Then, press 4 to enter Admin mode.
  - b. Press 7 on the PED keypad to duplicate
  - c. When prompted, insert a blank target PED key, or a non-blank whose data is no longer needed. Press Enter and respond to the prompts.
  - d. Remove the newly imprinted PED key and press Enter. The PED goes back to PED key mode awaiting further commands.
  - e. Repeat the above steps to duplicate the other PED key role.
  - f. Identify the new PED keys with tags or other markers, and record a PED PIN (if any) in secure fashion, according to your security policies.
3. Create an orange Remote PED key and duplicate it. This is required for a remote connection between the central PED and the remote data center. Create one orange key duplicate for each data center location.
    - a. Connect the PED to the k570 USB port. If the PED screen doesn't respond, connect the A/C adapter to the PED and to a power outlet as well.



- b. As the System Administrator (ksadmin) SSH in to the appliance (or connect via serial port using your password) and execute `"/usr/safenet/lunaclient/bin/lunacm"` utility.
- c. Make sure an HSM admin slot is selected.

To see the available slots, enter:

```
lunacm:> slot list
```

Look for a slot with description "Admin Token Slot".

To select the active slot, enter:

```
lunacm:> slot set -slot <number>
```

- d. Login as SO with the command `role login -name SO`.
- e. Have an orange PED key ready. Create and imprint the RPV (Remote PED Vector):

```
ped vector init
```

- f. Respond to the PED prompts, and choose to make duplicate(s). Provide an orange PED key along with additional blanks to make duplicates.

4. Prepare a workstation to act as a remote PED server.

#### Note

More details for remote PED server set up are available in [Luna HSM Documentation](#).

- a. Install Luna Client version 7.0.1 or above, with remote PED as an option. This software is available on the [Customer Support Portal](#).
- b. Connect the remote PED to the workstation via USB. If the PED screen doesn't respond, connect the A/C adapter to the PED and to a power outlet as well.
- c. Open a Command Prompt window on the computer (for Windows 7, this must be an Administrator Command Prompt). Locate and run PedServer.exe. Set PedServer.exe to its "listening" mode.

```
c: > PedServer -m start
Ped Server Version 1.0.6 (10006)
Ped Server launched in startup mode.
```

```
Starting background process
Background process started
Ped Server Process created, exiting this process.
c:\PED\ >
```

- d. Verify that the service has started with `pedserver -mode show`.

Look for mention of the default port "1503" (or other, if you specified a different listening port). In addition, "Ped2 Connection Status:" should say "Connected." This indicates that the Luna PED that you connected was found by PED Server.

- e. Take note of the workstation's IP address. This is required for the k570 in the data center to initiate a connection to the workstation.

5. Ship the duplicate PED keys to their data center destinations.

## Perform initial setup at the data center

1. Rack mount the appliance.
2. Establish a connection and change default passwords.
3. As the System Administrator (ksadmin) SSH in to the appliance (or connect via serial port using your password) and execute `/usr/safenet/lunaclient/bin/lunacm` utility.

The utility displays information on the detected HSM card and allows you to execute various HSM management commands.

4. Make sure an HSM admin slot is selected.

To see the available slots, enter:

```
lunacm:> slot list
```

Look for a slot with description "Admin Token Slot".

To select the active slot, enter:

```
lunacm:> slot set -slot <number>
```

5. Re-initialize the HSM.

```
lunacm:> hsm factoryReset
lunacm:> hsm init -label <admin token slot label>
```

**Note**

At this point, you can use `slot list` to see that the slot with description "Admin Token Slot" now has a label.

You are asked to present a red HSM domain key to the PED. Following PED prompts, re-use the shipped red key.

6. Initialize the HSM Security Officer (SO) role:

```
lunacm:> role login -n so
```

You are asked to present a blue HSM SO key to the PED. Following PED prompts, re-use the shipped blue key.

7. Initialize the remote PED vector, to allow a remote connection from headquarters.

```
ped vector init
```

You are asked to present an orange remote PED key to the PED. Following PED prompts, present the shipped orange key.

## Initialize the partition and re-set the PED keys

These steps are performed from the central headquarters location, using the Remote PED.

1. SSH to the remote HSM and open lunacm.

```
ssh -I <key> ksadmin@<IP>
/usr/safenet/lunaclient/bin/lunacm
```

2. Establish remote PED connection from the k570 to the remote PED workstation at headquarters.

**Note**

Other approaches for establishing a PED connection are described in [Luna Documentation](#).

```
ped connect -ip -port 1503
```

3. Re-imprint blue and orange PED keys to change the passwords.

- 
- a. Overwrite the HSM remote PED vector key. Present the orange key to the remote PED.

```
role login -name so
ped vector init
role logout
```

- b. Overwrite the HSM SO (blue) key to change the password. Present the blue key to the remote PED.

```
role login -name so
role changepw -name so
role logout
```

- c. Exit and relaunch LunaCM to test the new set of orange and blue keys.
- d. Re-establish the remote PED connection.

```
ped connect -ip <remote_PED_workstation_IP> -port 1503
```

- e. Authenticate with the newly issued blue and orange keys.

```
role login -name so
```

4. Create the partition for the CipherTrust Manager root of trust and logout.

```
partition create
role logout
```

5. Use `slot list` to view the slot number for the new partition and switch to that slot.

```
slot list
slot set -slot <slot number of user token slot created above>
```

6. Connect again.

```
ped connect -ip <remote_PED_workstation_IP> -port 1503
```

7. Initialize partition and the partition SO role.

```
partition init -label <new_partition_label>
role login -name Partition SO
```

---

You are asked to create a new blue Partition SO key, or re-use an old one. Follow the PED prompts.

8. As the Partition SO, activate the partition.

**Note**

This instructs the HSM to cache PED credentials and allows the k570 appliance to authenticate to the HSM using only the challenge secret (password) without requiring the black PED key to always be connected to the HSM. However, in the event of a power outage of more than 2 hours, the HSM cached PED credentials will expire and the k570 appliance will fail to run its services. In this case, instruct the k570 appliance to re-authenticate with the HSM using the black PED key.

```
lunacm:> partition changepolicy -policy 22 -value 1 lunacm:> partition changepolicy -policy 23 -value 1
```

9. As the Partition SO, initialize the Crypto officer role:

- a. Enter the command to initialize.

```
lunacm:> role init -name Crypto Officer
```

- b. Respond to prompts on the terminal and PED to create the initial Crypto Officer credential.

**Caution**

The Crypto Officer PED key is valid for the initial login only. The Crypto Officer must change this initial credential using the command `role changepw` immediately. Failing to change the credential results in a `CKR_PIN_EXPIRED` error when accessing the partition.

- c. If using PED authentication, create an initial Crypto Officer challenge secret. As with the PED key, it is valid for the first Crypto Officer login only and must be changed immediately.

```
lunacm:>role createchallenge -name co
```

10. Reset the Crypto Officer's credentials.

- a. Log in the Crypto Officer. When prompted for the password, provide the initial challenge secret (PED authentication).

```
lunacm:> role login -name Crypto Officer
```

- 
- b. Run the following command, which will reset the Crypto Officer PED key secret. Respond to the PED and terminal prompts.

```
lunacm:> role changePw -name Crypto Officer
```

- c. Change the initial challenge password. The passwords are not masked.

```
lunacm:> role changePw -name Crypto Officer -old <existing challenge secret> -newpw <new challenge secret>
```

- d. Log in again to activate/cache the new Crypto Officer credentials.

```
lunacm:> role login -name Crypto Officer
```

11. Exit the `lunacm` utility.

12. Repeat the initial data center set up and remote partition initialization for every k570 device in the data center.

## Backing Up k570 Root of Trust Keys

The Thales CipherTrust Manager k570 appliance embeds a Luna PCIe HSM, to act as root of trust, storing root of trust keys in a partition.

For redundancy, you can backup the root of trust keys from a Luna PCIe HSM to a Luna Backup HSM. You can then restore the root of trust keys on to a different Thales CipherTrust Manager k570 appliance to meet the required configuration.

### Note

The steps documented on this page are specific to Thales CipherTrust Manager k570 appliances. Contact Trusted Cyber Technologies customer support for help performing this operation with the TCT CipherTrust Manager k570 appliance.

Supported Luna Backup HSMs are:

- Luna Backup HSM 7 v1
- Luna Backup HSM 7 v2
- Luna Backup HSM G5

If you have upgraded the PCIe HSM firmware, make sure the Backup HSM has a [version compatible with HSM firmware 7.7.0 or higher](#)

The steps depend on whether the k570 PCI HSM is [password-authenticated](#) or [PIN Entry Device\(PED\)-authenticated](#). The [prerequisites](#) and required target Thales CipherTrust Manager configuration are the same for

---

---

both authentication types.

## Prerequisites

- You require `ksadmin` level access to the CipherTrust Manager with an SSH key.
- Obtain Luna HSM Client from the Thales customer support portal.

We recommend version 10.4.0 or higher, for full compatibility with all Luna Backup HSM models and the k570.

- Make sure the target CipherTrust Manager k570 to be restored to meets the required configuration.

### Caution

Failing to meet the required target Thales CipherTrust Manager k570 configuration can result in the CipherTrust Manager application becoming unavailable after reboot, requiring customer support to recover.

## Required Target Thales CipherTrust Manager k570 Configuration

Before you can restore keys into a CipherTrust Manager k570 appliance, you must ensure that there are no root of trust keys already present on the Luna PCI HSM.

### New k570 Appliance

For a new k570 appliance which has never been deployed:

1. Proceed with Thales CipherTrust Manager k570 physical appliance deployment, including initializing the Luna PCIe HSM partition.
2. Do not setup the root of trust HSM before restoring root of trust keys.

If you have set up the root of trust HSM, follow the steps for existing k570 appliance.

### Existing k570 Appliance

For a k570 which has a root of trust HSM configured, or has ever contained data such as keys or users:

1. Perform a system backup to retain any keys or users. Do not use the **tied to HSM** option.
2. Perform a system reset of the appliance.
3. As the Crypto Officer, log into the partition and delete any keys remaining on the Luna PCIe HSM partition.
  - a. SSH in as `ksadmin`, and execute `"/usr/safenet/lunaclient/bin/lunacm"` utility.
  - b. Find and login to the User Token Slot.

```
lunacm:> slot list
```

Notice the slot with the slot description "User Token Slot".

```
lunacm:> slot set -slot <slot number of user-token-slot>
```

- c. Login as the Crypto Officer.

```
lunacm:> role login -name co
```

- d. Clear all the keys present in the partition. You are prompted to confirm.

```
lunacm:> partition clear
```

- e. List the partition contents to make sure there are no objects remaining.

```
lunacm:> partition contents
```

- f. Exit the `lunacm` utility.

## Backup and Restore for Password-Authenticated HSM

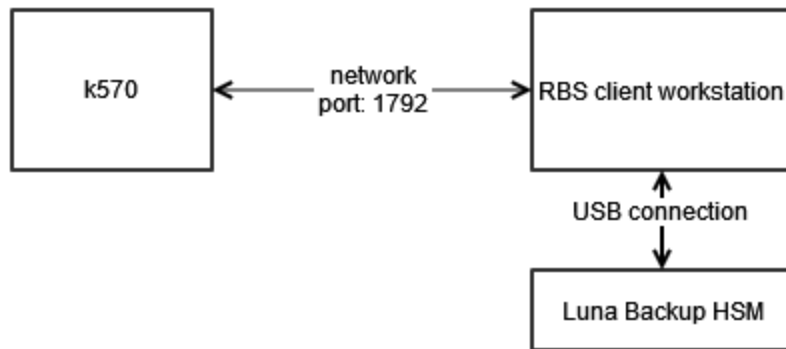
After you have fulfill the prerequisites, the high level workflow is for password-authenticated HSM is:

1. Establish connections between all the devices, client workstation, source k570, and Luna Backup HSM.
2. Backup the root of trust keys.
3. Establish connections between all the devices, client workstation, target k570, and Luna Backup HSM.
4. Restore the root of trust keys to a new k570.

## Required Network Setup

With the Luna PCIe HSM and a LunaCM client embedded in the Thales CipherTrust Manager k570 appliance and hardened USB access, backup and restore requires a specific networking setup for all components.





You must install the Remote Backup Server (RBS) on a workstation. RBS acts as a client to the Luna Backup HSM, and the Luna PCIe HSM acts as a client to RBS. RBS connects to the Luna Backup HSM through a local USB connection. The k570 PCIe HSM uses the vtl utility to connect to RBS over port 1792.

## Establish connections between all devices

1. Use the provided USB cable to connect the Luna Backup HSM to the client workstation.
2. Install Luna HSM Client on your client workstation, including the **Backup** component.
3. Find the rbs program included in the Luna client installation.

The default path on Windows is `C:\Program Files\SafeNet\LunaClient\rbs.exe`, and the default path on Linux is `/usr/safenet/lunaclient/rbs/bin`.

4. Run the command `rbs --config` to select the Luna Backup HSM device.

When you have specified your selection, enter X to exit the configuration tool.

5. If a server key and password don't exist for RBS, generate a new key with command `rbs --genkey` and enter a new RBS password when prompted.

The certificate is generated at the following locations:

- Linux/UNIX: `<LunaClient_install_directory>/rbs/server/server.pem`
- Windows: `<LunaClient_install_directory>\cert\server\server.pem`

6. Start the rbs server on port 1792 with the command `rbs s -port 1792`. Enter the RBS password that you created in the previous step.
7. Securely transfer the `server.pem` certificate on the client workstation to the CipherTrust Manager k570 using `scp`. The certificate is located at `C:\Program Files\SafeNet\LunaClient\certs\server\`.
8. SSH into the CipherTrust Manager k570 as `ksadmin`.
9. Change into the directory containing the vtl utility. The vtl utility is located at `/usr/safenet/lunaclient/`

---

```
bin.
```

10. Use vtl to connect the Luna PCIe HSM to the RBS client workstation.

```
./vtl addserver -n <client_workstation_ip_address> -c <path_to_the_server.pem>
```

11. Verify the Luna Backup HSM is visible with vtl.

```
./vtl listservers
```

## Back up the Root of Trust Keys

1. Establish connections between all the devices, client workstation, source k570, and Luna Backup HSM.
2. Still in an SSH session as ksadmin, execute `"/usr/safenet/lunaclient/bin/lunacm"` utility.
3. Identify the three visible slots, and note the displayed `Slot Id` for each slot. The `Slot Description` field indicates each slot's purpose.
  - `User Token Slot` contains the k570 root of trust keys, and is the authorization point for access to the keys.
  - `Admin Token Slot` is the slot for configuring Luna PCIe HSM overall.
  - `Net Admin Token Slot` indicates the Luna Backup HSM device.
4. Switch to the slot for the `User Token Slot`.

```
slot set -s <user_token_slot_id>
```

5. Login as the Crypto Officer. Use the password or challenge secret configured during deployment.

```
role login -n co -p <crypto_officer_password>
```

6. Backup the partition contents, including root of trust keys, to the Luna Backup HSM. This process creates a new partition on the Luna Backup HSM. You set a security officer password for the Backup HSM, and set a partition name, cloning domain, and partition password for the new partition.

```
partition archive backup -s <net_admin_token_slot> -partition <a_backup_partition_name> -do <cloning_domain_name> -sop <backup_hsm_security_officer_password> -pas <partition_password_on_backup>
```

## Restore the Root of Trust Keys to a Different k570 Appliance

### Caution

Ensure the k570 appliance meets the [required target Thales CipherTrust Manager k570 configuration](#). Failing to meet this configuration can result in the CipherTrust Manager application becoming unavailable after reboot, requiring customer support to recover.

1. Establish connections between all the devices, client workstation, target k570, and Luna Backup HSM.
2. Still in an SSH session as ksadmin, execute `"/usr/safenet/lunaclient/bin/lunacm"` utility.
3. Identify the three visible slots, and note the displayed `Slot Id` for each slot. The `Slot Description` field indicates each slot's purpose.
  - `User Token Slot` is the target slot which will contain the k570 root of trust keys, and act as the authorization point for access to the keys.
  - `Admin Token Slot` is the slot for configuring Luna PCIe HSM overall.
  - `Net Admin Token Slot` indicates the Luna Backup HSM device.
4. Switch to the slot for the `User Token Slot`.

```
slot set -s <user_token_slot_id>
```

5. Login as the Crypto Officer. Use the password or challenge secret configured during deployment.

```
role login -n co -p <crypto_officer_password>
```

6. Restore the root of trust keys from the backup device slot. Provide the backup HSM's partition name, cloning domain, security officer password, and partition password values set during backup.

```
partition archive restore -s <net_admin_token_slot> -partition <the_backup_partition_name> -do <cloning_domain_name> -sop <security_officer_password> -pas <partition_password_on_backup>
```

7. Proceed to [configure the Luna PCIe as the Root of Trust](#).

You require the 'partition-label' and the 'partition challenge' created during the HSM initialization procedure.

8. (Optional) If you have restored multiple root of trust keys, you can [check which key is active](#), and [rotate the active key](#) to a different root of trust key, if desired.

## Backup and Restore for PED-Authenticated HSM

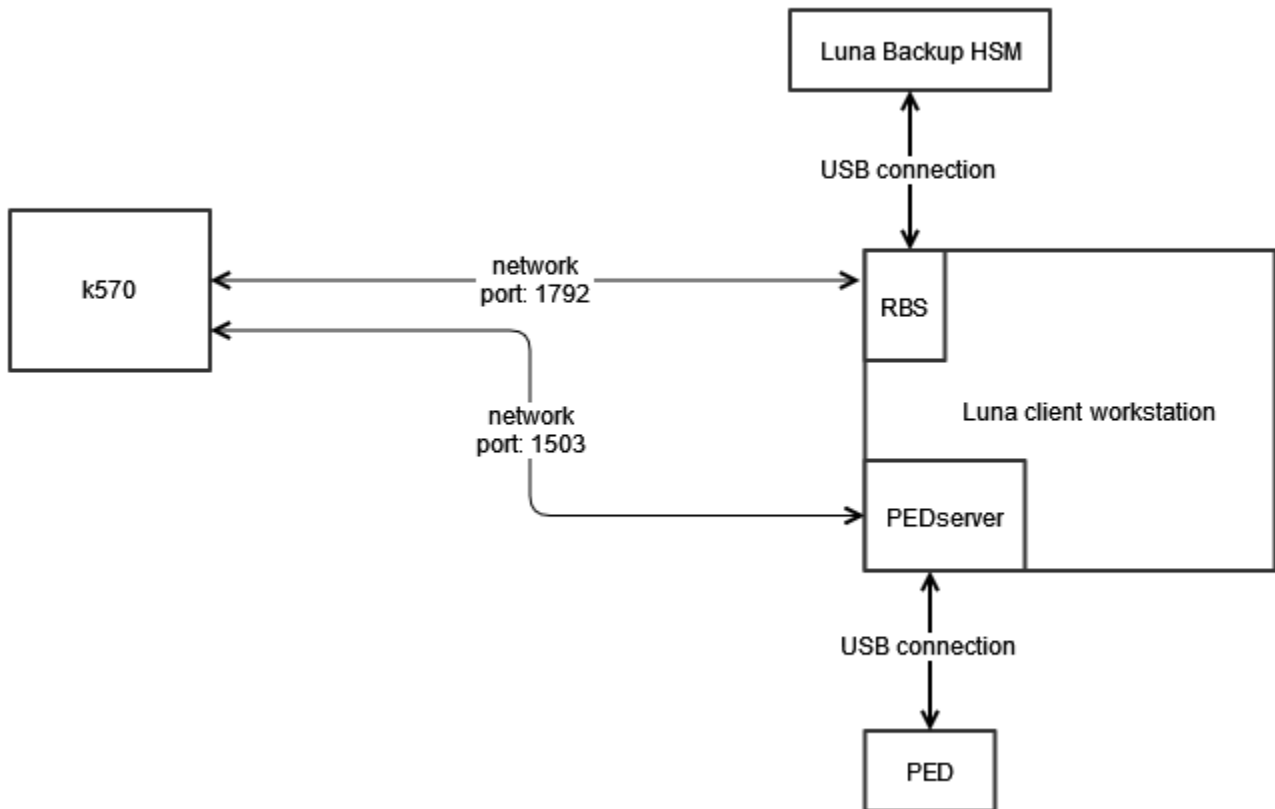
After you have fulfilled the prerequisites, the high level steps are:

1. Set up the client workstation and Luna Backup HSM for remote ped authentication.
2. Establish connections between the source k570 and client workstation.
3. Backup the root of trust keys.
4. Establish connections between the target k570 and client workstation.
5. Restore the root of trust keys to a new k570.

### Note

If you are consulting Luna documentation, PED authentication is now referred to as multifactor quorum authentication.

## Required Network Setup for PED Authentication



You must install Luna Client on a work station, including the Remote Backup Server (RBS) and PEDServer utilities.

RBS acts as a client to the Luna Backup HSM, and the Luna PCIe HSM acts as a client to RBS. RBS connects to the Luna Backup HSM through a local USB connection. The k570 PCIe HSM uses the vtl utility to connect to RBS over port 1792.

PEDServer allows for remote access to the PED over network port 1503. This is required to authenticate the k570 PCI-e HSM and the Luna Backup HSM at the work station.

## Set Up the Client Workstation and Luna Backup HSM for Remote PED Authentication

1. Use the provided USB cable to connect the Luna Backup HSM to the client workstation.
2. Use the provided USB cable to connect the PED to the client workstation.
3. Install Luna HSM Client on your client workstation, including the **Backup** and **Remote PED** component.
4. Navigate to the Luna client install directory and start the PEDserver utility.

```
pedserver -mode start
```

### Note

The default directory path on Windows is C:\Program Files\SafeNet\LunaClient\, and the default directory path on Linux is /usr/safenet/lunaclient/.

5. Initialize the Luna Backup HSM for PED/multifactor quorum authentication, as specified in the Luna documentation for the particular Backup HSM model.
  - [Initializing for remote multifactor quorum authentication \(Applies to both Luna Backup HSM v1 and v2\)](#)
  - [Initializing Luna Backup HSM G5 for PED authentication](#)
6. Login as the HSM SO. When prompted, insert the blue PED key.

```
lunacm:> role login -n so
```

7. In LunaCM, run the command to initialize the remote PED vector and confirm. When prompted, insert a blank key into the PED to be imprinted as an orange PED vector PED key.

```
lunacm:> ped vector -init
```

```

You are about to initialize the Remote PED Vector
Are you sure you wish to continue?

Type 'proceed' to continue, or 'quit' to quit now -> proceed

RPV was successfully initialized.

Command Result : No Error

```

8. Set up a secure channel between the Luna Backup HSM and the PED server, if you have not already done so.

- a. Login as the HSM SO.

```
lunacm:> role login -n so
```

- b. Provide the IP address of the Remote PED server.

```
lunacm:> ped set -ip <client_workstation_ip_address> -port 1503
```

- c. Connect to the Remote PED server.

```
lunacm:> ped connect
```

- d. Login as the HSM SO again. Respond to the PED prompts to insert the blue HSM SO and orange PED Vector key.

```
lunacm:> role login -n so
```

## Establish Connections Between the Client Workstation and the k570

1. Start the Remote Backup Server (RBS) and connect it to the k570
2. Find the rbs program included in the Luna client installation.

The default path on Windows is `C:\Program Files\SafeNet\LunaClient\rbs.exe`, and the default path on Linux is `/usr/safenet/lunaclient/rbs/bin`.

3. Run the command `rbs --config` to select the Luna Backup HSM device.

When you have specified your selection, enter X to exit the configuration tool.

4. If a server key and password don't exist for RBS, generate a new key with command `rbs --genkey` and enter a new RBS password when prompted.

---

The certificate is located at:

- Linux/UNIX: `<LunaClient_install_directory>/rbs/server/server.pem`
- Windows: `<LunaClient_install_directory>\cert\server\server.pem`

5. Start the rbs server on port 1792 with the command `rbs s -port 1792`. Enter the RBS password that you created in the previous step.
6. Securely transfer the `server.pem` certificate on the client workstation to the CipherTrust Manager k570 using `scp`. The certificate is located at `C:\Program Files\SafeNet\LunaClient\certs\server\.`
7. SSH into the CipherTrust Manager k570 as `ksadmin`.
8. Change into the directory containing the `vtl` utility. The `vtl` utility is located at `/usr/safenet/lunaclient/bin`.
9. Use `vtl` to connect the Luna PCIe HSM to the RBS client workstation.

```
./vtl addserver -n <client_workstation_ip_address> -c <path_to_the_server.pem>
```

10. Verify the Luna Backup HSM is visible with `vtl`.

```
./vtl listservers
```

11. Create an orange Remote PED key for the k570.
  - a. Connect the PED to the k570 USB port. If the PED screen doesn't respond, connect the A/C adapter to the PED and to a power outlet as well.
  - b. Change into the `/usr/safenet/lunaclient/bin/lunacm` directory.
  - c. Login as SO with the command `role login -name SO`.
  - d. Have an orange PED key ready. Create and imprint the RPV (Remote PED Vector):

```
hsm ped vector init
```

- e. Respond to the PED prompts.
- f. Re-connect the PED to the client workstation.

## Backup the Root of Trust Keys from PED-Authenticated k570

1. On the k570 CipherTrust Manager, set the Luna client to access the Luna Backup HSM and use the remote PED.

- a. SSH in as ksadmin, and execute `/usr/safenet/lunaclient/bin/lunacm` utility.
- b. Identify the three visible slots, and note the displayed `Slot Id` for each slot. The `Slot Description` field indicates each slot's purpose.
  - `User Token Slot` is the slot which contains the k570 root of trust keys, and act as the authorization point for access to the keys.
  - `Admin Token Slot` is the slot for configuring Luna PCIe HSM overall.
  - `Net Admin Token Slot` indicates the Luna Backup HSM device.
- c. Switch to the slot for the `Net Admin Token Slot`.

```
slot set -s <luna_backup_hsm_slot_id>
```

- d. Provide the IP address of the Remote PED server.

```
lunacm:> ped set -ip <client_workstation_ip_address> -port 1503
```

- e. View the PED connection type for the Luna Backup HSM.

```
lunacm:> ped get
```

The result shows that the Luna Backup device is set to local PED mode.

```
HSM slot 1 listening to local PED (PED id=0).
```

- f. Confirm the Remote PED Server IP address and port are set correctly.

```
lunacm:>ped show

Configured Remote PED Server information

Remote PED Server IP address: <configured_ip_address>
Remote PED Server Port: 1503

Command Result : No Error
```

- g. Connect the Luna Backup HSM to the Remote PED Server.

```
lunacm:>ped connect
```



```
Command Result : No Error
```

## 2. Backup the k570 root of trust keys to the Luna Backup HSM.

- a. List the visible slots and identify the slot number for the `User Token Slot`, which is the slot which contains the k570 root of trust keys.

```
lunacm:>slot list
```

- b. Switch to the `User Token Slot`

```
lunacm:>slot set -s <slot_number>
```

- c. Login as the Crypto Officer. Use the challenge secret configured during deployment.

```
role login -n co -p <crypto_officer_password>
```

- d. Backup the partition contents, including root of trust keys, to the Luna Backup HSM. This process creates a new partition on the Luna Backup HSM. As part of this process, you present the orange Remote PED vector, blue HSM SO, and red Domain PED Keys initialized for the Luna Backup HSM to the HSM connected to the client workstation. You also present the black Crypto Officer PED key initialized for the k570 root of trust key partition. Respond to the PED prompts.

```
lunacm:>partition archive backup -s <net_admin_token_slot> -partition <back
up_HSM_partition_name>
```

## Restore the Root of Trust Keys to a Different PED-Authenticated k570

### Caution

Ensure the k570 appliance meets the [required target Thales CipherTrust Manager k570 configuration](#). Failing to meet this configuration can result in the CipherTrust Manager application becoming unavailable after reboot, requiring customer support to recover.

1. Establish connections between the client workstation and the k570.
2. In an SSH session as ksadmin, execute `"/usr/safenet/lunaclient/bin/lunacm"` utility.
3. Identify the three visible slots, and note the displayed `Slot Id` for each slot. The `Slot Description` field indicates each slot's purpose.

- `User Token Slot` is the target slot which will contain the k570 root of trust keys, and act as the authorization point for access to the keys.
- `Admin Token Slot` is the slot for configuring Luna PCIe HSM overall.
- `Net Admin Token Slot` indicates the Luna Backup HSM device.

4. Switch to the slot for the `User Token Slot`.

```
slot set -s <user_token_slot_id>
```

5. Login as the Crypto Officer. Use the challenge secret configured during deployment.

```
role login -n co -p <crypto_officer_password>
```

6. Restore the root of trust keys from the backup device slot. When prompted, provide the orange Remote PED key and the black Crypto Officer PED key.

```
partition archive restore -s <net_admin_token_slot> -partition <the_backup_partition_name>
```

7. Proceed to [configure the Luna PCIe as the Root of Trust](#).

You require the 'partition-label' and the 'partition challenge' created during the HSM initialization procedure.

8. (Optional) If you have restored multiple root of trust keys, you can [check which key is active](#), and [rotate the active key](#) to a different root of trust key, if desired.

## Hardware Specifications

### Available Models

- k470 - no HSM, 4x1Gbit ports
- k470 - no HSM, 2x1Gbit ports and 2x10Gbit ports
- k570 - HSM, 4x1Gbit ports
- k570 - HSM, 2x1Gbit ports and 2x10Gbit ports
- k160 - HSM

### Specifications for k470 and k570

The base chassis for k470 and k570 are the same, and so the specifications are presented together. However,

the k570 includes a PCIe HSM and the k470 does not.

The CipherTrust Manager k470 and k570 Appliances are 1U high and fit into standard 19-inch equipment racks.



## Hardware Specifications for k470 and k570

| Hardware Specification      | Value                                                                                                                                                                                                          |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dimensions                  | 19.0"(W) x 21"(D) x 1.75"(H)                                                                                                                                                                                   |
| Weight                      | 12.7 kg(28lbs)                                                                                                                                                                                                 |
| Memory                      | 16 GB                                                                                                                                                                                                          |
| Hard Disk and Protections   | 1 X 2TB SATA SE (Spinning Disk)                                                                                                                                                                                |
| Serial Port                 | 1                                                                                                                                                                                                              |
| CPU                         | 1 CPU with 4 cores                                                                                                                                                                                             |
| Ethernet / NICs             | 4 X 1 GB<br>2 X 10 GB<br>2 X 1 GB                                                                                                                                                                              |
| IPMI                        | Not Available                                                                                                                                                                                                  |
| Power Supplies              | 2 Power Supplies; Average Power (Watts) 0.7A @120V 84W<br>Maximum Power (Watts) 100-240V 50-60Hz<br>Voltage: 100W<br>Power Cord PSE Certified<br>Available for multiple country profiles for power consumption |
| Chassis Intrusion Detection | k470 and k570: Tamper seals                                                                                                                                                                                    |
| Operating Temperature       | 0 to 35°C (32 to 95°F)                                                                                                                                                                                         |
| Non-Operating Temperature   | -20 to 60°C (-4 to 140°F)                                                                                                                                                                                      |
| Safety Agency Approvals     | CB Scheme, CSA-UL, FCC Part 15, Subpart B, Class B, EN55032:2010, EN55024:2010, EN61000-3-2:2006 +A1:2009 +A2:2009, EN61000-3-3:2008 ICES-003 Issue 4 February 2004, C-Tick, AS/NZS CISPR                      |

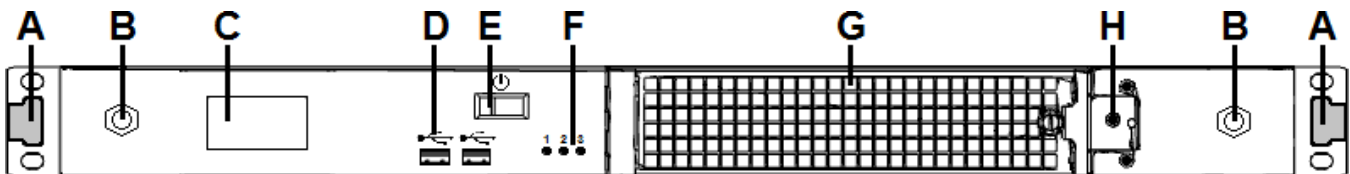
| Hardware Specification      | Value                                                                                                                                                                                                       |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                             | 22:2009, VCCI V-3/2009.04, KN22, KN24, BIS (in progress)                                                                                                                                                    |
| FIPS 140-2 Certifications   | Level 3 with HSM as root of trust <ul style="list-style-type: none"> <li>▪ Certification for k570 included HSM, external Luna Network HSM</li> <li>▪ Certification for Data Protection on Demand</li> </ul> |
| Embedded HSM Administration | K570 (Built in HSM) , Management Console and REST API allow configuration to HSM                                                                                                                            |

## Firmware Specifications for k470 and k570

| Firmware Specification                  | Value                                                                                                                                                  |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Administrative Interfaces               | Management Console                                                                                                                                     |
| Max Keys                                | 1,000,000 keys                                                                                                                                         |
| Max Domains                             | 1000                                                                                                                                                   |
| API Support                             | <ul style="list-style-type: none"> <li>• <b>Native:</b> REST, NAE-XML, KMIP</li> <li>• <b>SDK:</b> PKCS#11, JCE, .NET Core, MS CAPI, MS CNG</li> </ul> |
| Security Authentication                 | Username/Password (Local User), AD/LDAP , Certificate based authentication                                                                             |
| Cluster Support                         | Clustering is supported between physical and /or virtual appliances for High Availability                                                              |
| Backup                                  | Manual and Scheduled; Option for HSM key to encrypt CM backup                                                                                          |
| Network Management                      | SNMP v1, v2c, v3 , NTP, Syslog-TCP                                                                                                                     |
| Syslog Formats                          | RFC-5424, CEF, LEEF                                                                                                                                    |
| Software Certifications and Validations | FIPS 140-2 L3 with k570; K470 and Virtual CipherTrust Manager can use an External HSM as Root of Trust for Master Key protection                       |

## Front Panel for k470 and k570

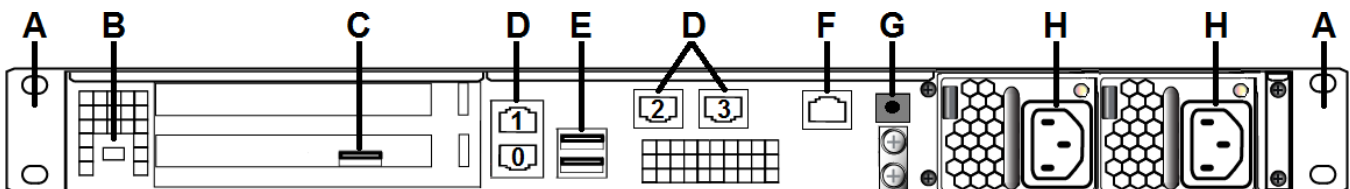
The front panel is illustrated below, with the secure locking bezel removed:



| Item | Name                         | Description                                                                                                                                                                                    |
|------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A    | Front ear brackets           | Connect to the front of the appliance chassis with the provided screws, allowing it to be mounted in a standard 19-inch equipment rack. The extending tabs act as posts for the locking bezel. |
| B    | Mounts for locking bezel     | The secure locking bezel connects to the appliance faceplate here.                                                                                                                             |
| C    | Front-panel display          | Displays basic configuration and status information for the appliance.                                                                                                                         |
| D    | USB 3.0 ports                | These USB ports are disabled during the bootup sequence. We do not recommend using them. We recommend monitoring these ports by <a href="#">forwarding host log messages</a> .                 |
| E    | Stop/start switch            | Powers the appliance on or off.                                                                                                                                                                |
| F    | Fan status LEDs              | The appliance has three (3) cooling fans. If these lights are illuminated, the fans are working correctly.                                                                                     |
| G    | Ventilation fan filter cover | Removable cover allows cleaning of air filter.                                                                                                                                                 |
| H    | Fan bay securing screws      | Torx screw secures the fan bay.<br><br>Opening to swap fan modules triggers a tamper event on the appliance.                                                                                   |

## Rear Panel for k470 and k570

The rear panel is illustrated below:

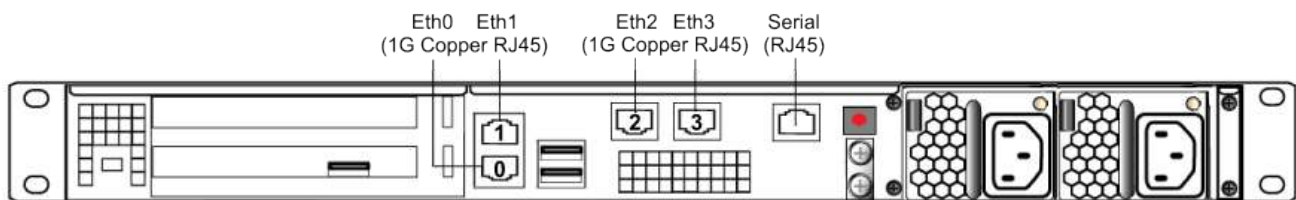


| Item | Name                      | Description                                                                                                                                 |
|------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| A    | Sliding rail brackets     | Connect to the sliding rails mounted on the sides of the appliance chassis, allowing it to be mounted in a standard 19-inch appliance rack. |
| B    | Kensington lock connector | Allows the appliance to be secured to a desk or equipment rack using Kensington lock.                                                       |

| Item | Name                                    | Description                                                                                                                                                                    |
|------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| C    | Models with HSM: HSM card with USB port | When authenticating with a PED, the PED must be connected directly to the USB port of the HSM card. The other USB ports on the appliance will not work for PED connection.     |
| C    | Models without HSM: <i>No HSM card</i>  | No HSM card installed - a blank plate is installed.                                                                                                                            |
| D    | LAN ports                               | The appliance has a total of four (4) 1Gbit LAN ports.                                                                                                                         |
| E    | USB 3.0 ports                           | These USB ports are disabled during the bootup sequence. We do not recommend using them. We recommend monitoring these ports by <a href="#">forwarding host log messages</a> . |
| F    | RJ45 serial port                        | Connect a terminal to this port using the included RJ45 to USB cable.                                                                                                          |
| G    | Decommission button                     | This button should only be pressed as part of decommissioning and zeroizing the appliance.                                                                                     |
| H    | Power supplies                          | 2 Power Supplies connect the appliance to power. For proper redundancy and best reliability, the power cables should connect to two completely independent power sources.      |

## Network Interface for 1Gbps k470 and k570 Models

1Gbps models come with four 1 Gbit RJ45 Ethernet network ports (labeled 0, 1, 2, and 3). The network interfaces are mapped to Eth $n$ , as detailed in the following illustration.



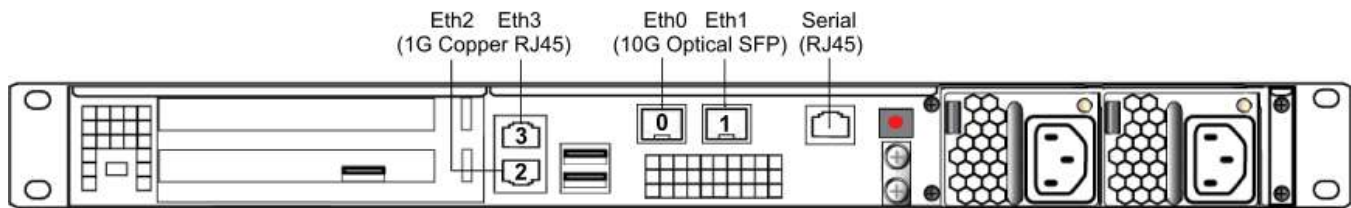
All ports are identical and equally usable and can be bonded in any combination. By default port Eth0 is configured for DHCP but this can be configured.

Refer to [Network Configuration Tutorial](#) for more information.

## Network Interface Mapping for 1Gbps/10Gbps k470 and k570 Models

10 Gbps models provide two 10Gbps SFP optical Ethernet network ports (labeled 0 and 1) and two 1Gbps copper RJ45 network ports (labeled 2 and 3). The network interfaces are mapped to Eth $n$ , as detailed in the following

illustration.



You can optionally bond the network interfaces Eth0 with Eth1 or Eth2 with Eth3 to form a logical interface, providing a redundant active/standby virtual interface. By default port Eth0 is configured for DHCP but this can be configured.

Refer to [Network Configuration Tutorial](#) for more information.

## Front Panel LCD for k470 and k570 Models

The Front Panel displays the product name (CipherTrust), the firmware version, and the IP address of any configured network interface.

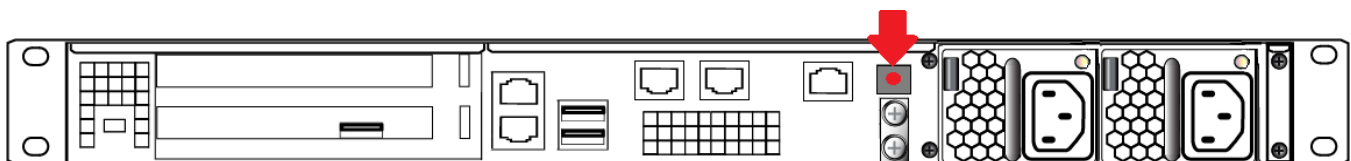


## HSM Emergency Decommission Button for k570 models

The CipherTrust Manager k570 Appliance includes a way to decommission the HSM, or permanently deny access to all objects on it, without need for either a serial console or a remote (SSH) connection.

To directly decommission the HSM inside the appliance, press and release the small red button on the rear panel.

- The appliance does not need to be powered on.
- The appliance does not need to have power cables connected.



You will need a small screw-driver or other tool to reach the Emergency Decommission button. This is intentional, to prevent accidental pressing of that button.

## What the Emergency Decommission Button Does

When you press the Decommission button, all partitions and their contents are deleted, as well as the audit role, and the audit configuration. The HSM policy settings are retained.

**Warning**

For k570 models, pressing the Decommission button makes all data and keys created on the CipherTrust Manager become **permanently** unusable, including keys in backups associated to the HSM.

To bring the HSM back into service, you need to:

1. Reinitialize the HSM.
2. Reinitialize the audit role and reconfigure auditing.
3. Recreate the partitions.
4. Reinitialize the partition roles.
5. Reset the system `kscfg system reset`.
6. Setup the HSM as described in [Hardware Security Module](#).

At this point, you can recreate encryption keys on CipherTrust Manager.

## Power Consumption for k470 and k570 Models

When installed and connected to appropriate electrical power sources, CipherTrust Manager k470 or k570 Appliance draws power as follows:

| Activity                                                      | Draw                                                               |
|---------------------------------------------------------------|--------------------------------------------------------------------|
| Standby (connected to AC electrical mains but not powered on) | 26W (typical)                                                      |
| Power-on Input Surge                                          | 15A (typical)<br>40A at 90-132VAC (max)<br>60A at 180-265VAC (max) |
| Active (under load from clients)                              | 84W to 90W (typical)<br>100W to 105W (max)                         |

The appliance has two power supplies, each rated at 350W, either of which is capable of running the system alone.

## Hardware Specifications for k160 Model

The TCT CipherTrust Manager k160 Appliance is a compact cryptographic key management platform that protects and manages cryptographic keys and associated policies used to encrypt the most sensitive data-at-rest. This cost-effective solution is ideal for small to medium sized deployments commonly found in small offices, remote sites, and tactical environments. The k160 includes a FIPS 140-2 Level 3 token or a high assurance cryptographic token as its hardware root of trust. The token hardware security module (HSM) operates as a secure root of trust by encrypting all sensitive objects (e.g. keys, certificates, etc.) in CipherTrust Manager, with keys that are generated by, and reside in, the token HSM. The removable token HSM provides an easy to use method to support common key management scenarios such as rapid key delivery disablement, key destruction,



cryptographic erase, and time of use restrictions. Simply removing the token allows for keeping mission critical data safe.



## k160 Technical Specifications

### Physical Characteristics

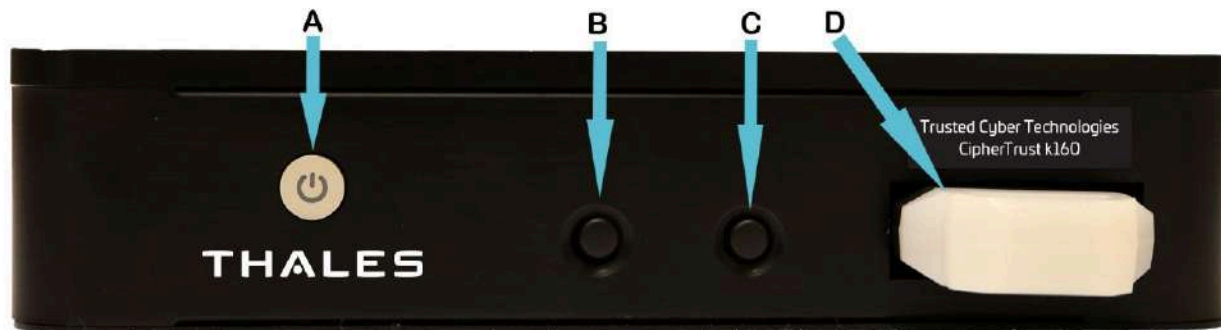
- CipherTrust k160 Dimensions: 6.5" x 4.0" x 1.5"
- Weight: 1.2 lbs.
- Direct mount or 1U 19in. rack mount (optional accessory)
- Thermal Storage: -30°C ~ 80°C
- Thermal Operation: -30 ~ 65°C
- Storage Humidity: 5 ~ 95% @ 40C
- Operating Humidity: 0% ~ 90% relative humidity
- Vibration Testing: Random, 1Gm, 5~500Hz
- Power: included external power supply; locking DC power connector
- Power Range: input 120-240V AC, 1.5A, 50-60Hz; output 12V DC, 40W

### Interfaces

- Web UI Management
- Serial and SSH command line
- KMIP, NAE-XML, and REST API
- 1G Ethernet interface
- Integrated Token HSM connection

## k160 Front Panel

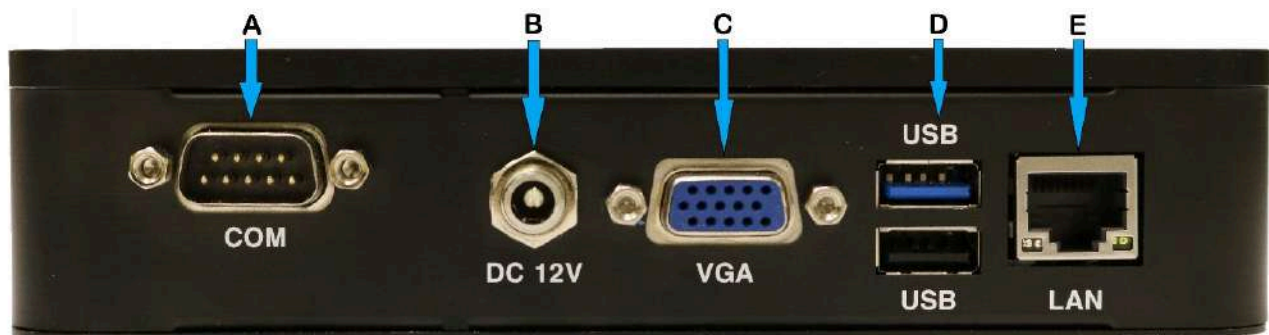
The front panel is illustrated below, showing the High Assurance token inserted:



| Item | Name           | Description                     |
|------|----------------|---------------------------------|
| A    | Power On/Off   | Powers the appliance on or off. |
| B    | Round button 1 | Reserved for future use.        |
| C    | Round button 2 | Reserved for future use.        |
| D    | USB HSM token  | High Assurance Token.           |

## k160 Rear Panel

The rear panel is illustrated below:



| Item | Name         | Description                                                          |
|------|--------------|----------------------------------------------------------------------|
| A    | Console Port | Connect a terminal to this port using the included DB9 to USB cable. |
| B    | DC 12V       | Power Adapter connection.                                            |
| C    | VGA          | Connect a standard VGA monitor.                                      |
| D    | USB ports    | Disabled/Not used.                                                   |
| E    | LAN          | CAT5 Networking Port                                                 |

---

# HSM Tamper Behavior

## Tampering with the Appliance

Hardware tamper events are detectable events that imply intrusion into the appliance interior.

One such event is removal of the lid (top cover). The lid is secured by anti-tamper screws, so any event that lifts that lid is likely to be a serious intrusion.

## Decommission

The red "Decommission" button recessed behind the back panel is not a tamper switch. Its purpose is different. See [HSM Emergency Decommission Button](#) for a description.

## What Happens When You Tamper

First, the HSM disconnects and disallows any communication.

Second, the system turns on the "HSM Offline" alarm to alert system admins. This will happen within 30 seconds.

Third, if the system is unable to communicate with the HSM consecutively for 5 minutes, it restarts itself to ensure that all sensitive cached information is erased from the running system. If the system is still unable to communicate with the HSM after a restart, services are prevented from starting up and the status API reflects this problem. In addition, the GUI displays an error indicating a problem communicating with the HSM.

A tampered system does not recover by itself and requires the system admin to intervene. Recovering requires console access (for example SSH). If you are configured for PED authentication, you will also need access to the HSM PED and the corresponding HSM tokens (iKeys).

### To clear a tamper flag on a Thales k570 local HSM

1. Make sure an HSM admin slot is selected.
  - a. Run the `/usr/safenet/lunaclient/bin/lunacm` tool in the console as the system admin (ksadmin).
  - b. Check the available slots.

```
lunacm:> slot list
```

- c. Look for a slot with description "Admin Token Slot".
  - d. Select the active slot.

```
lunacm:> slot set -slot <number>
```

2. Verify that the system is unable to communicate with the HSM using this command:

```
lunacm:> hsm tampershow
```

Response:

```
Driver command failed: Input/output error
Failed to display tampers error=5.

Command Result : No Error
```

3. An HSM restart is required before any operations can be performed against the HSM using this command:

```
lunacm:> hsm restart
```

#### Note

If the PED is connected to the system, it is required at this time to reconnect it by physically unplugging and reconnecting the USB cable; otherwise it will not be able to communicate with the HSM.

4. Make sure an HSM admin slot is selected. Refer to step 1 for details.
5. After the HSM has restarted, view the tamper event using this command:

```
lunacm:> hsm tampershow
```

6. You can now clear the tamper flag using these commands:

```
lunacm:> role login -name so
lunacm:> hsm tamperclear
```

At this point the HSM should be operational again.

#### Note

During tampering all cached HSM tokens are cleared. Therefore, any CipherTrust Manager attempt to log in to the HSM will trigger PED authenticate. Re-authentication using the black PED key is required to get the appliance running again.

## To clear a tamper flag on a TCT k570 local HSM

1. Power on the unit. If the unit was still running when the tamper occurred, power cycle the unit. Connect to the unit over serial cable and run the following as the `ksadmin`:

```
cd /usr/safenet/lunaclient/bin
./lunacm
```

The HSM indicates it has been tampered.

2. If you are configured for Password Authentication, to clear the tamper flag, log in as the HSM/Security Officer:

```
:::text
lunacm:> hsm login
```

3. If you are configured for PED Authentication, to clear the tamper flag, use the following commands:

- a. `lunacm:> hsm login`

You are prompted for the HSM/Security Officer (blue) iKey

- b. `lunacm:> partition login`

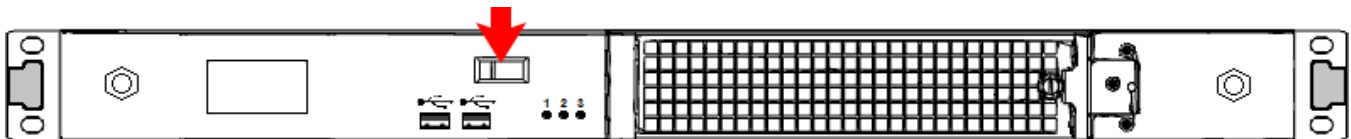
You will be prompted for the User (black) iKey.

## Reboot the Appliance

### Power On

On the back panel, ensure that the power supplies are connected and working - the green LED on each power supply glows steadily when in operation.

If the appliance does not immediately begin to start up, press and release the **START/STOP** switch  on the front panel.



The appliance begins to power up.

If power was removed while the system was on (either a power failure, or the power cable was disconnected), the system restarts without a button press. This behavior allows unattended resumption of activity after power interruption.

The front-panel LCD begins showing activity, then settles into the ongoing system status display once the appliance has completed its boot-up and self-test activity.

### Note

For k570s configured with local HSMs using PED authentication: if the appliance has been powered down for more than two hours, the PED credentials are no longer cached. In this case, after booting the system, disconnect and reconnect the PED USB cable, and re-authenticate with the HSM using the black PED key when prompted. With auto-activation enabled, the HSM re-caches the PED credentials. You can also configure [remote PED access](#).

## Power Off

To power-off the appliance locally, press and release the START/STOP switch. Do not hold it in. The appliance then performs an orderly shutdown (that is, it closes the file system and shuts down services in proper order for the next startup). This takes approximately 30 seconds to complete. In the unlikely event that the system freezes and does not respond to a momentary “STOP” switch-press, then press and hold the START/STOP switch for five seconds. This is an override that forces immediate shutoff.

### Caution

Never disconnect the power by pulling the power plug. Always use the START/STOP switch.

To switch off the appliance from the console command line, use the command **sudo poweroff**.

## Reboot

To perform a system restart, you can switch the power off and then on again using the momentary-contact START/STOP switch on the front panel of the system, or issue the **sudo reboot** command.

To switch off the system, issue the **sudo poweroff** command, or use the START/STOP switch on the appliance front panel:

- If you issue the **poweroff** command, the graceful shutdown takes 15 to 20 seconds
- After you momentarily press and release the START/STOP switch, the system performs a graceful shutdown, which takes 15 to 20 seconds.

If the system does not appear to be properly shutting down, then press and hold the front-panel START/STOP switch, which forces an immediate shutdown. This is not normally required, and should never be done unless it is required, since it bypasses the normal, graceful file-system closing and shutdown procedure.

## Automatic Restart Following a Power Interruption

If the appliance was deliberately powered down, using the START/STOP switch or the **sudo poweroff** command, it remains off until you press the START/STOP switch. However, if power was removed while the system was on (either a power failure, or the power cables were disconnected - not good practice), then the system restarts without a button press.

This behavior allows unattended resumption of activity after power interruption. In most cases, it is assumed that

this would never be needed, as you would install the appliance with its two power supplies connected to two completely separate, independent power sources, at least one of which would be battery-backed (uninterruptible power supply) and/or generator-backed.

## Power Supply and Fan Maintenance

The two power supplies in the CipherTrust Manager k470 or k570 Appliance are hot-swap capable, meaning that one is sufficient to power the appliance while the other is removed and replaced, with no service interruption. The indicator light (LED) on each power supply shows different behavior, depending upon conditions.

| Power Supply Condition                          | Power Supply LED                    |
|-------------------------------------------------|-------------------------------------|
| DC present/only standby output on               | Flashing green (1Hz)                |
| Power supply DC output ON and OK                | Steady green                        |
| Power supply failure                            | Steady RED                          |
| Power supply warning                            | Flashing Blue/Red (1Hz) alternating |
| Input power failure (only in n+1 configuration) | Flashing Red (1Hz)                  |

A power supply controller in the appliance monitors the state of the power supplies. It ensures that a failed power supply still gets sufficient direct current from the remaining power supply to light the indicator LED. The controller also sounds an audible alarm when there is a problem, such as one power supply not being connected to AC main power.

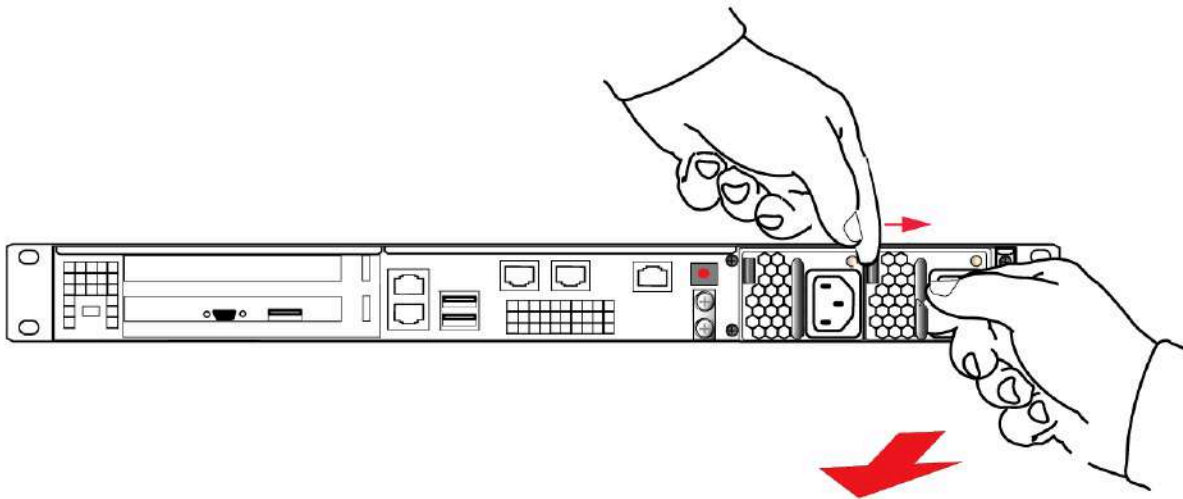
If only one power supply is present, the audible alarm is silent. If you wish to operate your appliance with only one power supply, we recommend that you remove the second supply to silence the audible alarm.

## Replacing a Power Supply

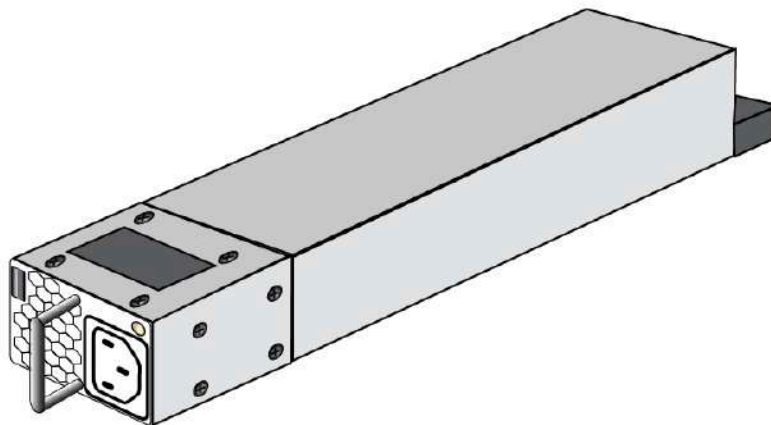
You may need to replace a power supply in the event of a failure.

### To remove a power supply:

1. Face the rear panel of the appliance.
2. Disconnect/unplug the selected power supply.
3. Press the lever sideways to release the power supply retaining catch, and simultaneously pull the handle out toward you.



4. Withdraw the power supply completely, using your other hand to support the body of the power supply as it emerges.



### To Reinstall a Power Supply:

1. To replace a power supply, reverse the steps above. Press firmly to seat the connector. The power supply can be fully inserted only in its proper orientation.
2. Connect an AC power cord.



---

## The Fans

In normal operation, the fans should require no maintenance.

You might need to perform the following tasks:

- Clean the filter (occasionally)
- Replace a defective fan (rarely)

### Cleaning the Filter

The ventilation grille, located to the right, on the appliance front panel, is secured in two parts, by two screws - a knurled, captive thumb-screw, and a Torx T8 screw. The knurled screw can be fastened or released without tools. It secures the lattice screen that in turn retains the mesh air filter.

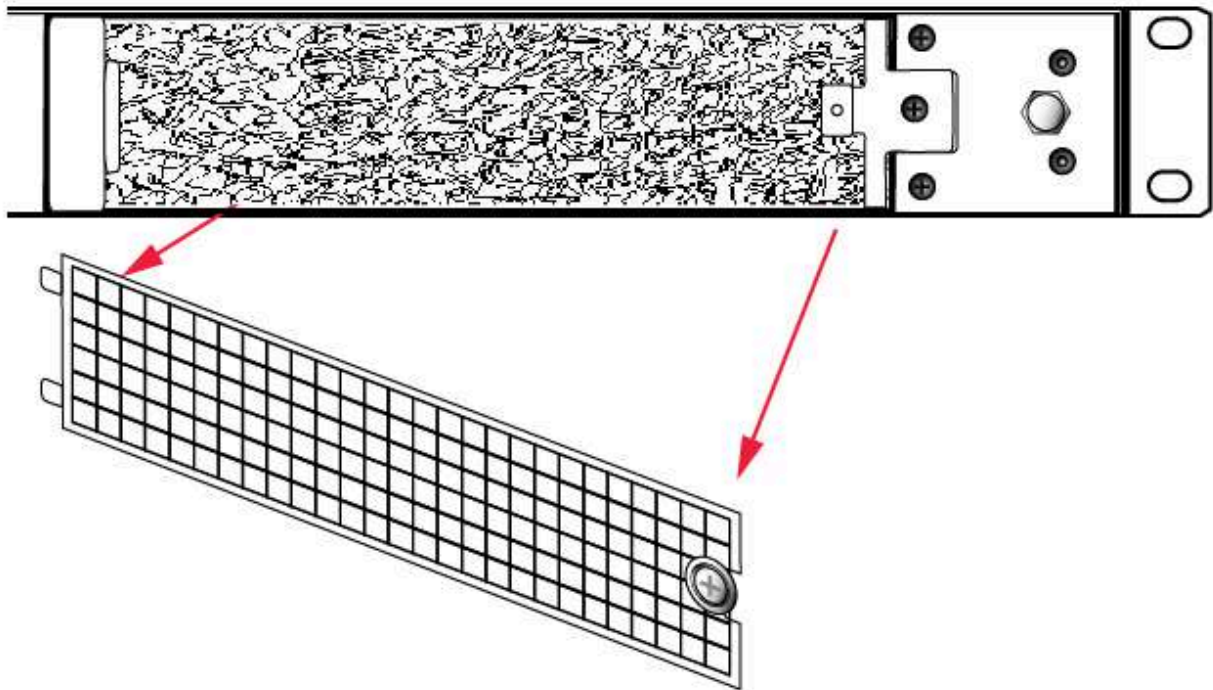
While we recommend controlled-atmosphere environments for greatest longevity and reliability of the equipment, we recognize that some environments might include some dust in the air. The mesh filter traps larger particulate matter before it can be drawn into the interior of the appliance. In less-than-perfect non-clean-room conditions, the mesh might accumulate a buildup of dust, and should be cleaned occasionally for best cooling airflow into the equipment.

#### Note

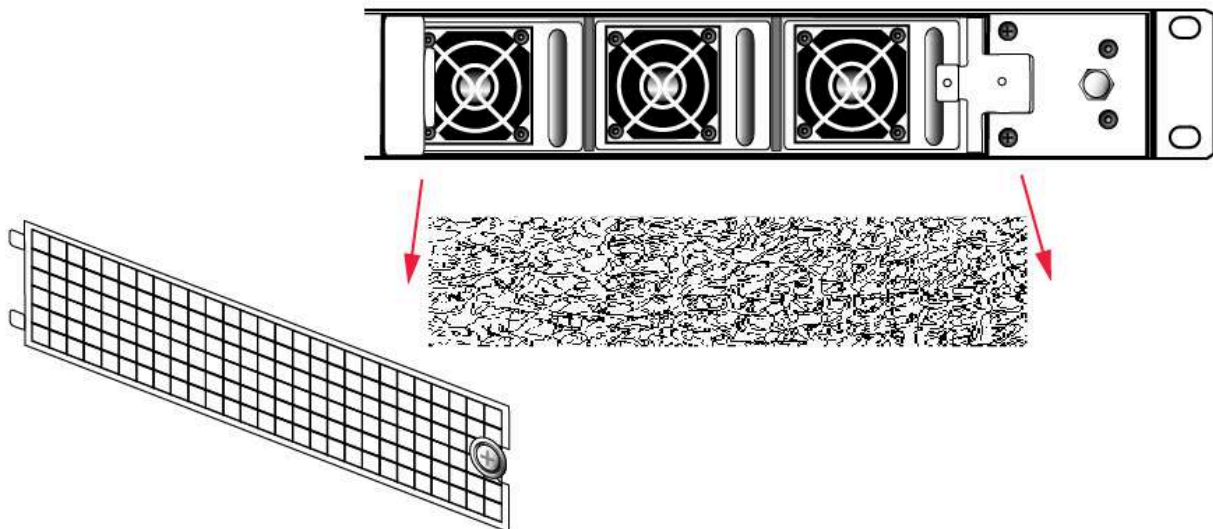
Accessing the air filter mesh in front of the fans (using the thumbscrew to open the retaining grille) does not cause a tamper.

#### To clean the filter:

1. Twist the knurled knob counter-clockwise until it no longer secures the airflow lattice. The lattice is anchored at its left end by two tabs, and can be easily pulled off the appliance, once the retaining screw is loosened. Do so.



2. With the air filter exposed, it is easy to grasp the mesh with fingers and tug it free. The mesh is flexible and is held in its cavity only by friction. If it is dusty, handle carefully so as not to dislodge any dirt that could then be sucked in by the fans.



3. To clean the filter, either blow it out with compressed air (away from the vicinity of the appliance), or rinse with water. If using water, ensure that the mesh is dry before reinstalling.
4. To reinstall the mesh, place it in its cavity in front of the fans, and use fingers or a blunt tool to tuck-in the corners.
5. Then, replace the lattice in front of the mesh by inserting the tabs first, then swinging the lattice closed like a door, and securing with the knurled screw.

## Replacing a Fan

The three fan modules (each containing two in-line fans) provide cooling redundancy.

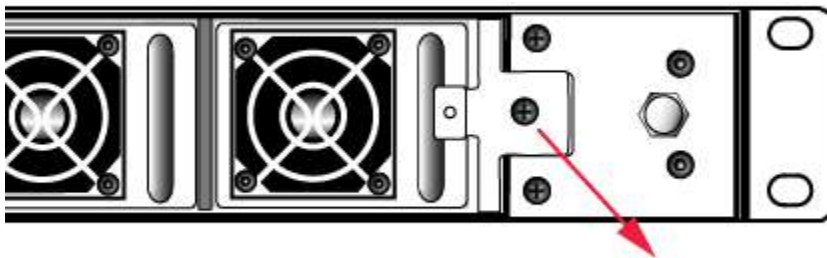
If a fan fails, an `ipmievd` event is logged in `/var/log/syslog` file accessible to the `ksadmin` user with SSH access. As well these messages can be [forwarded to an external syslog server](#) or downloaded to an LTSV file along with other [debug logs](#).

The following is an example event message:

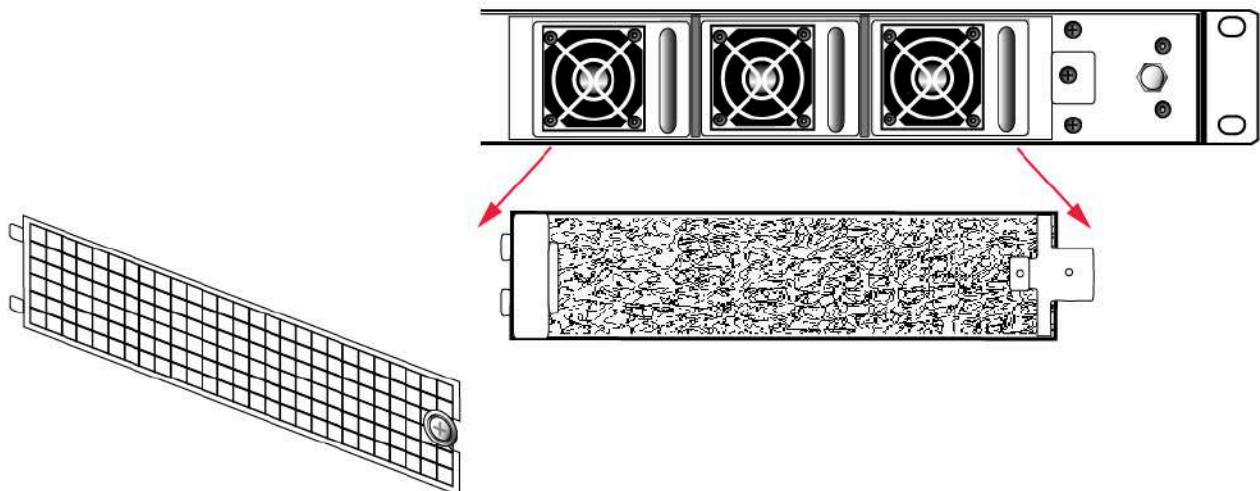
```
Sep 8 18:00:23 ciphertrust ipmievd: Fan sensor PSU1 FAN_value . Lower Critical going low Asserted (Reading 0 < Threshold 1300 RPM)
```

### To replace a fan

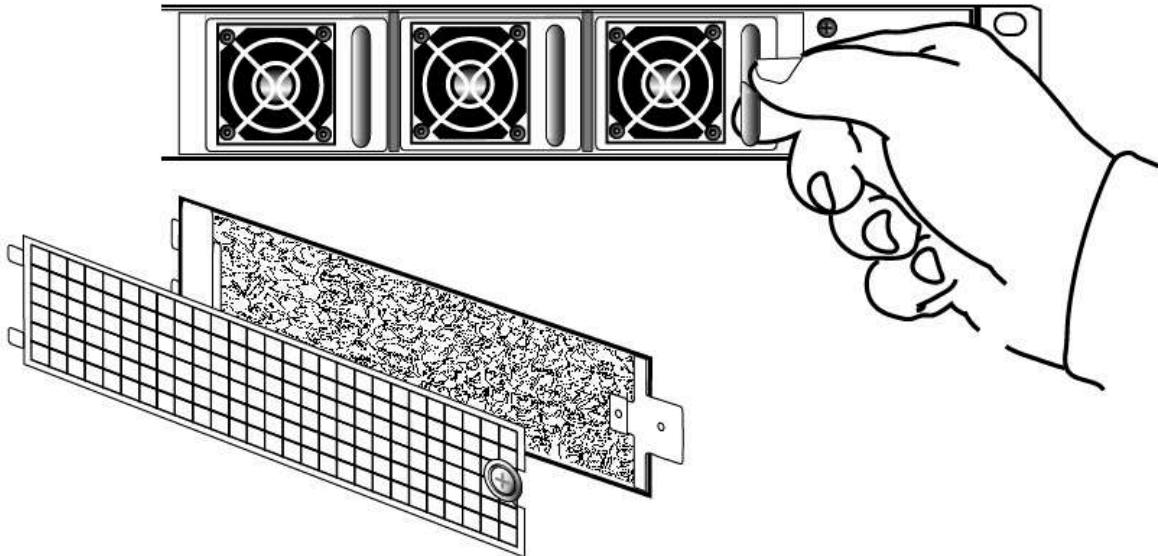
1. Power down the appliance. In an SSH session, login as `ksadmin` and issue the **`sudo reboot`** command.
2. Open the fan bay. Use a Torx number 8 screwdriver to remove the screw that secures the right-side tab of the fan retainer.



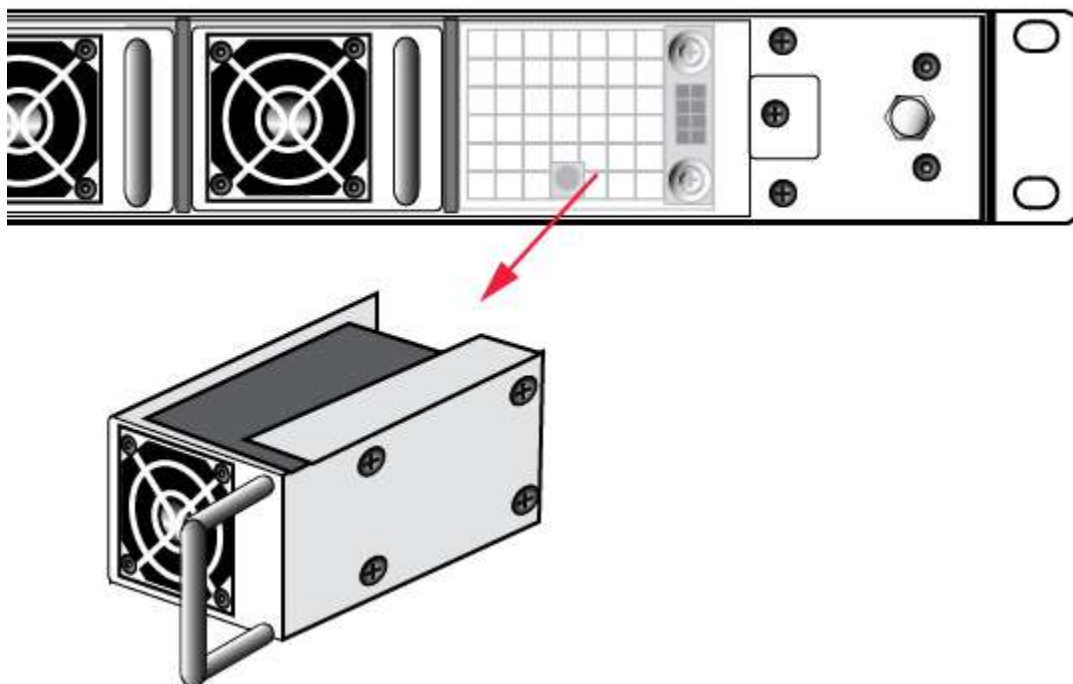
3. The fan retainer is anchored at its left by two tabs - swing the retainer out like a door, and remove it. There is no need to separate the filter mesh and its retainer from the larger fan retainer; the assembly can come out as one piece. The illustration below happens to show them separated.



4. The fan modules are now exposed and are held in place only by the friction of their electrical connectors.
5. Grasp the handle of the selected fan module and pull straight out toward you.



6. After slight initial resistance, the fan module should easily slide free of the appliance.



7. To replace the fan module or install a new one, reverse the above sequence.

The index peg on the back of the module, and the matching index hole at the back of the fan bay, ensure that the module can be inserted only in its proper orientation.

8. Power on the appliance by pressing and releasing the **START/STOP** switch on the front panel.

### Note

For k570s configured with local HSMs, if the appliance has been powered down for more than two hours, `ksadmin` needs to reactivate the HSM partition (`lunacm partition activate -partition <name_of_partition>`), so that they once more become available to your registered clients.

Additionally, if the HSM uses PED authentication, the PED credentials are no longer cached after two hours. In this case, after booting the system, disconnect and reconnect the PED USB cable, and re-authenticate with the HSM using the black PED key when prompted. With auto-activation enabled, the HSM re-caches the PED credentials. You can also configure [remote PED access](#).

## Serial Connections

You can connect a computer directly to the console port of the CipherTrust Manager Appliance using a serial connection. From the console you log in as the System Administrator (aka, 'ksadmin' user), create a secure password, start-up the system and access the IP address of the Graphical User Interface (GUI).

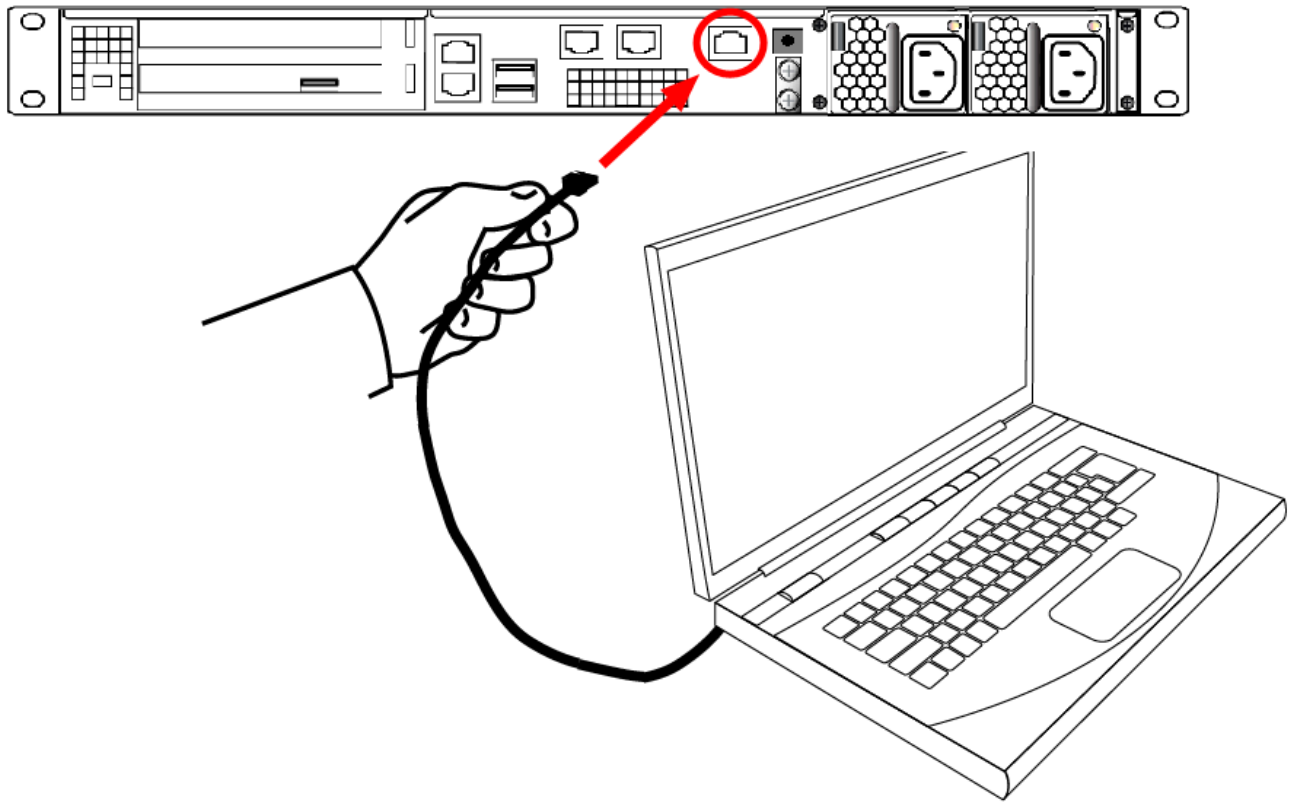
Direct administration connection to the console via serial terminal is required for these reasons:

- During initial configuration, do not yet know the IP address dynamically allocated by your DHCP server.
- After deployment, if you re-configure network settings (i.e., change the IP address) via SSH, you will lose the old IP address connection.

## Open a Serial Connection

1. Connect the serial port on the appliance's rear panel to a terminal server, dumb terminal, PC, or laptop, using the supplied Prolific Technology Inc. USB to RJ45 (with 8P8C connector) adapter.



**Note**

Do not connect the serial cable to one of the Ethernet ports.

2. If the driver for the Prolific Technology Inc. USB to RJ45 (with 8P8C connector) adapter did not download and install automatically, go to [the Prolific Technology Inc website](#) to download and install the PL2303 USB-to-Serial Windows driver.

3. Open **Device Manager (Control Panel > Hardware > Device Manager)** and expand the **Ports (COM and LPT)** folder. If the driver installed successfully, an entry is displayed for the **Prolific USB-to-Serial Comm Port**, followed by the port associated with the adapter. For example:

Prolific USB-to-Serial Comm Port (COM4)

Record the COM port (COM4 in this example) associated with the adapter. You will need this port number when you open a serial connection.

4. Use a terminal emulation package, such as PuTTY, to open a serial connection to the COM port associated with your Prolific USB-to-Serial adapter. Set the serial connection parameters as follows:

**Baud rate:** 19200

**Data bits:** 8



---

**Parity:** None

**Stop bits:** 1

5. When the connection is made, the appliance log in prompt is displayed: **ciphertrust login:**.

#### Note

You may need to press **ENTER** several times to initiate the session.

6. Type Ctrl+C to display the IP address.
7. As the System Administrator, enter "ksadmin" to log in and follow the prompts to create a secure password.

#### Caution

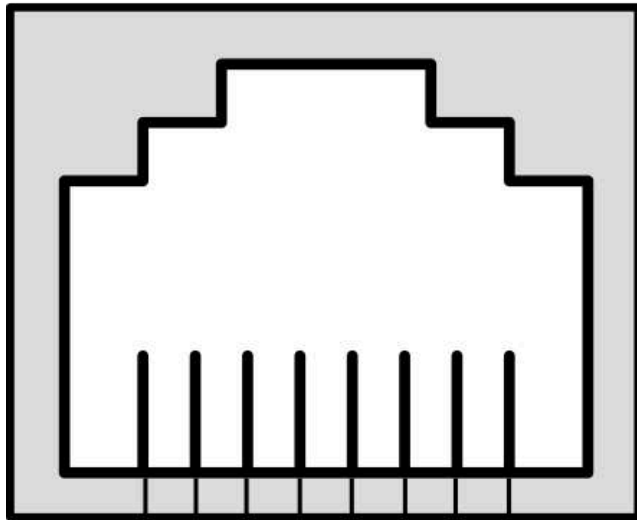
Be sure to retain this password - it will be required to access the system in case of network connectivity problems.

The system starts up, which can take several minutes.

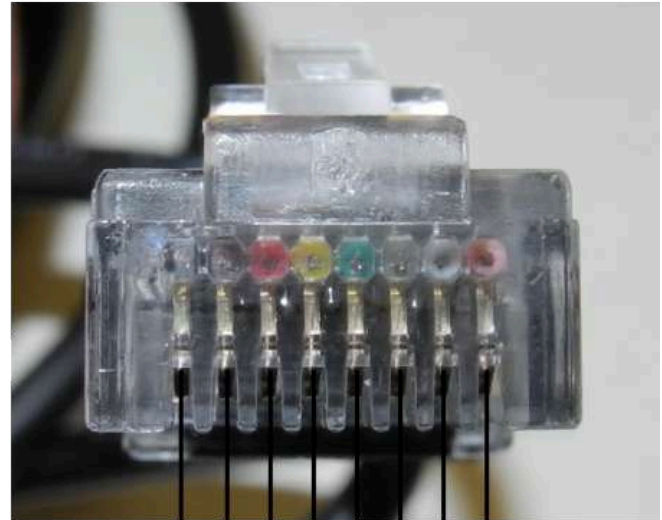
## Serial Port Pinout

The serial port uses a configuration equivalent to the Cisco Terminal Console. The Prolific Technologies Inc. RJ45-to-USB serial adapter cable uses a standard RJ45 pinout configuration:

HSM Serial Port



RJ45-to-USB serial cable



- 1 RTS
- 2 DTR
- 3 TxD
- 4 GND
- 5 GND
- 6 RxD
- 7 DSR
- 8 CTS

#### Note

The CipherTrust Manager appliance does not implement serial handshaking, so RTS/CTS (pins 1 and 8) and DSR/DTR (pins 7 and 2) are not used for a successful connection.

The important signals are TxD and RxD (pins 3 and 6).

## Troubleshooting

This section contains recommendations for troubleshooting serial connections.

### Windows 10 occasionally crashes when trying to detect a serial port

This is a known issue with the Windows 10 PL2303 drivers. If you experience trouble opening a serial connection using Windows 10, use another supported operating system.



## Front Locking Bezel

The locking bezel (pictured below) fits over the front of the CipherTrust Manager k470 or k570 Appliance for maximum physical access security. The purpose of the bezel is to:

- cover the appliance's ports, and the power button,
- lock the appliance to the rack to prevent removal.



The locking bezel comes with three (3) keys for each lock. The locks are keyed differently so that keys can be issued to different security personnel and kept in secure, separate locations.

### To lock the bezel:

1. The locks fit over the posts highlighted below. Fit the bezel over the posts with both keys in the horizontal position.



2. Turn the keys to the vertical position to lock the bezel.



3. Remove the keys and store them in a secure location.

## Replacement Keys

To obtain replacement keys, contact Technical Support. Please have the lock serial numbers ready. You can find these numbers on the sides of the bezel by each lock.

## Network Configuration Tutorial

You can configure multiple network interfaces after installing and initializing a [CipherTrust Manager physical appliance](#) or [private cloud Virtual CipherTrust Manager](#).

### Caution

Network interface configuration on public clouds should be done using the cloud provider's tools.

This configuration is performed using [GNOME NetworkManager](#) and its `nmcli` tool.

This tutorial provides an example of typical network configuration steps. These steps show how to view network devices, view network connections, set DHCP with IPv4 for a connection, and then modify the connection to use a static IP instead of DHCP.

[Planning](#) is required for network interface bonding, configuring static routes, or configuring VLAN.

## Basic Network Configuration Tasks with nmcli

1. Connect as the `ksadmin` user to the CipherTrust Manager appliance through its serial connection, or the Virtual CipherTrust Manager through its console.

### Caution

Modifying a remote network interface over SSH is risky. The remote connection will stop responding if the IP address settings are incorrectly configured, resulting in the remote machine being unreachable.

2. Use `nmcli` to list the available network devices (also called network interfaces or NICs). The devices listed in this example are truncated for brevity.

```
nmcli device

DEVICE TYPE STATE CONNECTION
kylo0 bridge connected kylo0
ens3 ethernet connected Wired connection 1
ens4 ethernet disconnected --
veth05d02c4 ethernet unmanaged --
...
```

The output of this command might show over 30 devices. However, only a few require explanation:

- `kylo0` is used for internal communication by CipherTrust Manager services. It should never be altered in any way.
- `ens3` is an active device with a NetworkManager connection profile named `Wired Connection 1`.
- `ens4` is an inactive device with no NetworkManager connection profile defined.
- `veth05d02c4` is an unmanaged device and should be ignored because NetworkManager cannot be used to control or configure the device.

As a general rule, device names starting with `eth` or `en` are devices an administrator can configure.

3. Use `nmcli` to see an active device's live values. In this example the only active device so far is `ens3`.

```
nmcli device show ens3

GENERAL.DEVICE: ens3
GENERAL.TYPE: ethernet
GENERAL.HWADDR: 00:50:56:99:3F:54
GENERAL.MTU: 1500
GENERAL.STATE: 100 (connected)
GENERAL.CONNECTION: Wired connection 1
GENERAL.CON-PATH: /org/freedesktop/NetworkManager/ActiveConnection/8
```

```
WIRED-PROPERTIES.CARRIER: on
IP4.ADDRESS[1]: 10.121.105.97/22
IP4.GATEWAY: 10.121.104.1
IP4.DNS[1]: 10.121.8.7
IP4.DNS[2]: 172.16.2.13
IP4.DNS[3]: 172.16.2.12
IP6.ADDRESS[1]: fe80::bd7e:b93f:7e66:4b92/64
IP6.GATEWAY:
```

The output shows that IPv4 has an address and accompanying values and that IPv6 only has a link-local address.

4. Use `nmcli conn` to see the configured network devices that NetworkManager calls *connections*. Notice that the connections listed in this example do not include device `ens4` because it is not yet configured.

```
nmcli conn

NAME UUID TYPE DEV
ICE
Wired connection 1 9d86421b-7032-48eb-ac5a-3c84d285d01e 802-3-ethernet ens
3
```

5. Use `nmcli` to create a connection configuration for the device `ens4` using DHCP for IPv4 and no IPv6 configuration. When `ipv4.method` and/or `ipv6.method` are not explicitly provided, then NetworkManager uses `auto` by default.

### Note

There is a known issue in CipherTrust Manager instances upgraded from 2.4 and earlier, where the network interface name may be associated with a different MAC address after a reboot. Because of this issue, it is highly recommended to bind the connection to the device's MAC address instead of the network interface name when creating the connection profile.

- a. View details for `ens4`. Retain the MAC address, shown as the `GENERAL.HWADDR` value, for creating a connection later.

```
nmcli device show ens4

GENERAL.DEVICE: ens4
GENERAL.TYPE: ethernet
GENERAL.HWADDR: 00:50:56:99:3F:55
```

```
...
```

- b. Create the connection, binding it to the MAC address. Provide a meaningful connection name based on the context of use. For example, if `ens3` is responsible for web traffic and `ens4` is responsible for database traffic, then naming the connections `web` and `db` provides better context for how they are used.

```
nmcli conn add type ethernet con-name <connection_name> ifname ' ' -- ethernet.mac-address 00:50:56:99:3F:55 ipv4.method auto ipv6.method ignore
```

6. Use `nmcli` to confirm the connection is created with the correct configuration for device `ens4`. The configuration and active values in this example are truncated for brevity.

```
nmcli conn show <connection_name>

connection.id: <connection_name>
connection.uuid: d797d28c-fe8a-49ab-8181-271870d6cfc6
connection.interface-name: ens4
connection.type: 802-3-ethernet
...
ipv4.method: auto
...
ipv6.method: ignore
...
IP4.ADDRESS[1]: 10.121.105.113/22
IP4.GATEWAY: 10.121.104.1
IP4.DNS[1]: 10.121.8.7
IP4.DNS[2]: 172.16.2.13
IP4.DNS[3]: 172.16.2.12
...
```

The output format uses lowercase key names to indicate configuration values (e.g. `ipv4.method`) and uppercase key names to indicate live values (e.g. `IP4.ADDRESS[1]`). From this output we can see the newly configured device's IP address obtained via DHCP is 10.121.105.113.

7. Try connecting to the connection's IP address from a browser and confirm that CipherTrust Manager UI loads. If the UI does not load, try pinging the address to confirm the IP address can be reached or use other network tools such as `netcat` to verify connectivity.
8. You can use the `modify` sub-command to change the connection configuration. Use `nmcli` to modify device `ens4`'s connection to use a static IP address instead of DHCP. You must provide a gateway and DNS server(s).

```
nmcli conn modify <connection_name> ipv4.method manual ipv4.addresses 10.121.1
```

```
05.18/22 ipv4.gateway 10.121.104.1 ipv4.dns 8.8.8.8,8.8.4.4

nmcli conn show <connection_name> | grep IP4.ADDRESS

IP4.ADDRESS[1]: 10.121.105.113/22
```

9. To ensure that DHCP-provided DNS servers are ignored, run the command:

```
nmcli conn modify <connection_name> ipv4.ignore-auto-dns yes
```

10. Notice that the IP4.ADDRESS[1] field listed in the second command still has the original IP address obtained via DHCP. In order to activate the modification, the connection must be restarted using the `up` sub-command:

```
nmcli conn up <connection_name>

Connection successfully activated (D-Bus active path: /org/freedesktop/Network
Manager/ActiveConnection/15)

nmcli conn show <connection_name> | grep IP4.ADDRESS

IP4.ADDRESS[1]: 10.121.105.18/22
```

## Planning Network Configuration

You can configure multiple network interfaces after installing and initializing a [CipherTrust Manager physical appliance](#) or [private cloud Virtual CipherTrust Manager](#).

This configuration uses [GNOME NetworkManager](#) and its `nmcli` tool

`nmcli` is available to the `ksadmin` user through the CipherTrust Manager appliance's serial connection, or the Virtual CipherTrust Manager's console.

### Caution

Modifying a remote network interface over SSH is risky. The remote connection will stop responding if the IP address settings are incorrectly configured, resulting in the remote machine being unreachable.

This section details available settings and techniques to plan your network configuration. There is also a [network configuration tutorial](#) available for a simple set up without network interface bonding, VLAN, or static routes.

You might wish to:

- [set IPv4 or IPv6 for a Network Device](#).

---

We suggest explicitly disabling IPv6 unless it is required for your network. That way, there are fewer IP addresses exposed for CipherTrust Manager network devices, and therefore fewer ways for potential attackers to reach CipherTrust Manager.

- [configure a Static IP Address or DHCP](#).

By default, CipherTrust Manager uses DHCP. You can configure a static IP address or DHCP so that CipherTrust Manager conforms to the way your network more broadly assigns IP addresses.

- [set a static MAC address for a connection](#)

There is a known issue in CipherTrust Manager instances upgraded from 2.4 and earlier, where network device names sometimes swap MAC addresses after a reboot. This has been observed for network interfaces beginning with `eth` and bonded connections created from network interfaces beginning with `eth`.

We recommended creating a connection for each hardware network device so that the NetworkManager connection binds to the hardware device's MAC address instead of the network interface name, even if the network device is not connected. It is important to apply this to all network devices because even a network device that is not currently used may be used in the future.

- [configure a Static Route](#).

A static route can restrict externally-bound services, such as a connection to an external root-of-trust HSM, to a specific network interface. You can additionally disable the default route so that only traffic for a specific external service goes through the network interface.

- [bond network interfaces](#).

A bonded network interface provides redundancy and performance improvements by aggregating two or more network interfaces to form a single logical interface.

- [configure Virtual LAN \(VLAN\)](#).

VLAN can be useful to virtually isolate CipherTrust Manager network interfaces from other systems on the physical network to enhance security without the need for additional network equipment.

## Setting IPv4 or IPv6 Protocol for a Network Device

By default, NetworkManager uses `auto` for `ipv4.method` and `ipv6.method`, creating both IPv4 and IPv6 addresses for a CipherTrust Manager network device. You can explicitly provide `ipv4.method` and/or `ipv6.method` to disable creation of either type of address.

You can specify IPv4 or IPv6 when first adding a network connection, or when modifying a network connection.

Disabling IPv6 for a network interface does not disable the link local address. As traffic to and from link local addresses cannot be propagated through a router, an IPv6 link local address generally does not present security concerns with external attackers. If you still need to disable the link local address, contact customer support.

## To Add a New Connection with IPv4 Only

1. Use `nmcli` to see an active device's live values to create a connection. Retain the MAC address, shown as the `GENERAL.HWADDR` value, for creating a connection later.

```
$ nmcli device show ens3

GENERAL.DEVICE: ens3
GENERAL.TYPE: ethernet
GENERAL.HWADDR: 00:50:56:99:3F:54
```

2. In the `nmcli connection add` command, specify `ipv4.method auto` `ipv6.method ignore`.

```
nmcli conn add type ethernet con-name <connection_name> ifname ' ' -- ethernet.mac-address <desired_MAC_address> ipv4.method auto ipv6.method ignore
```

3. View the connection details to ensure IPv6 is disabled. `ipv6.method` is set to `ignore` for correct configuration. For example:

```
nmcli con show <connection_name>
connection.id: eth0
[...]
ipv6.method: ignore
[...]
IP6.ADDRESS[1]: fe80::c9c:4cff:fe6a:1267/64
```

## To Modify an Existing Connection to Disable IPv6

1. Modify the IPv6 configuration for the connection.

```
nmcli con modify <connection_name> ipv6.method ignore
```

2. Apply the configuration changes to the connection.

```
nmcli con up <connection_name>
```

3. View the connection details to ensure IPv6 is disabled. `ipv6.method` is set to `ignore` for correct configuration.

```
nmcli con show <connection_name>
connection.id: eth0
```



```
[...]
ipv6.method: ignore
[...]
IP6.ADDRESS[1]: fe80::c9c:4cff:fe6a:1267/64
```

## Configuring Static IP Address or DHCP

If you are setting up a new connection, use `nmcli conn add` to set static IP address or DHCP. If you are modifying an existing connection, use `nmcli conn modify` to set static IP address or DHCP. By default, CipherTrust Manager attempts to use DHCP for its connections.

Both static IP sections include setting manual gateways and DNS servers.

### To set a static IP for a new connection

1. Use `nmcli` to see an active device's live values to create a connection. Retain the MAC address, shown as the `GENERAL.HWADDR` value, for creating a connection later.

```
$ nmcli device show <device-name>

GENERAL.DEVICE: ens3
GENERAL.TYPE: ethernet
GENERAL.HWADDR: 00:50:56:99:3F:54
```

2. Run the following command, providing a gateway and DNS address, with the `auto` method to set a static IPv4 address and no IPv6 address:

```
nmcli conn add type ethernet con-name <connection_name> ifname ' ' -- etherne
t.mac-address <desired_MAC_address> ipv4.method manual ipv6.method ignore ipv
4.addresses <static_IP_address> ipv4.gateway <gateway_address> ipv4.dns <dns_a
ddresses>
```

3. To ensure that DHCP-provided DNS servers are ignored, run the command:

```
$ nmcli conn modify <connection_name> ipv4.ignore-auto-dns yes
```

4. To activate the DNS setting, restart the connection using the `up` sub-command:

```
$ nmcli conn up <connection_name>
```

### To change to a static IP from DHCP

1. Modify the connection, providing a manual static IP address, a gateway, and DNS servers.

```
$ nmcli conn modify <connection_name> ipv4.method manual ipv4.addresses <static_ip_address> ipv4.gateway <gateway_ip_address> ipv4.dns <dns_server>
```

2. To ensure that DHCP-provided DNS servers are ignored, run the command:

```
$ nmcli conn modify <connection_name> ipv4.ignore-auto-dns yes
```

3. To activate the modification, restart the connection using the `up` sub-command:

```
$ nmcli conn up <connection_name>
```

```
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/15)
```

## To set DHCP for a new connection

1. Use `nmcli` to see an active device's live values to create a connection. Retain the MAC address, shown as the `GENERAL.HWADDR` value, for creating a connection later.

```
$ nmcli device show <device-name>
```

```
GENERAL.DEVICE: ens3
GENERAL.TYPE: ethernet
GENERAL.HWADDR: 00:50:56:99:3F:54
```

2. Run the following command with the `auto` method to set DHCP with IPv4 and to ignore IPv6:

```
nmcli conn add type ethernet con-name <connection_name> ifname ' ' -- ethernet.mac-address <MAC_address> ipv4.method auto ipv6.method ignore
```

## To change to DHCP from a static IP

1. Run the following command with the `auto` method to set DHCP with IPv4 and to ignore IPv6:

```
nmcli conn modify con-name <connection_name> ifname ' ' ethernet.mac-address <desired_MAC_address> ipv4.method auto ipv6.method ignore
```

2. To activate the new setting, restart the connection using the `up` sub-command:

```
$ nmcli conn up <connection_name>
```

---

## Configuring Static MAC Address

For CipherTrust Manager appliances, some configurations can result in alternating MAC address after reboot. We have observed alternating MAC addresses with network interface names beginning with `eth` and bonded connections created from network interfaces beginning with `eth` in CipherTrust Manager devices upgraded from 2.4.

We recommended that each connection binds to the hardware device's MAC address instead of the network interface name, even if the network device is not connected. It is important to apply this to all network devices because even a network device that is not currently used may be used in the future.

### To set a connection to a static MAC address

This configuration applies to a non-bonded interface. To set a static MAC address for a bonded interface, consult [Bonding Network Interfaces](#).

1. View MAC address currently in use for a network interface.

```
nmcli device show <network-device-name>
```

#### Example Result

```
GENERAL.DEVICE: ens3
GENERAL.TYPE: ethernet
GENERAL.HWADDR: 00:50:56:99:3F:54
...
```

The `GENERAL.HWADDR` value is the MAC address to associate to a connection name.

2. Modify the desired connection to use the desired MAC address.

```
$ nmcli connection modify <connection_name> ifname ' ' ethernet.mac-address <desired_MAC_address>
```

3. Apply the configuration change.

```
$ nmcli conn up <connection_name>
```

---

## Configuring Static Routes

You can use static routes to ensure that outbound traffic from a specific service only goes through a specific network interface.

For example, you might want to ensure traffic to an external root-of-trust HSM only takes place through the `ens3` network interface. If you set a static route for `ens3` that includes the HSM's IP address range, CipherTrust Manager directs all traffic bound for the HSM's IP address through `ens3`.

---

If you want to dedicate a network interface to only route traffic from a specific service, and not to route traffic from other services, you must also [disable the default route](#).

To configure a static route:

1. Modify the NetworkManager connection to use a static route.

Run the command:

```
$ nmcli conn modify <connection name> +ipv4.routes "<static route> <gateway>"
```

**Example:**

```
$ nmcli conn modify ens3 +ipv4.routes "192.168.122.0/24 172.30.1.1"
```

2. Apply the configuration change.

Run the command:

```
$ nmcli conn up <connection name>
```

**Example:**

```
$ nmcli conn up <connection name>
```

```
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/10)
```

3. Verify that the route is active.

```
$ ip route

default via 172.30.1.1 dev ens3 proto static

172.30.0.128/26 dev kylo0 proto kernel scope link src 172.30.0.129

172.30.1.0/24 dev ens3 proto kernel scope link src 172.30.1.71

192.168.122.0/24 via 172.30.1.1 dev ens3 proto static
```

The output shows that the route is now active.

## Disable the Default Route

---

If an interface should *only* communicate with IP addresses matching a static route for the interface, remove the

---

default route entry. Removing the default route entry for an interface disallows traffic to all other destination IPs.

For example, if you set a static route to restrict HSM traffic to the eth1 interface, leaving the default entry in place means that CipherTrust Manager might also direct traffic for LDAP through eth1.

By default, all active network interfaces have a default route entry in the routing table. The metric value (lower means higher priority) determines which interface is used to route traffic that does not have an explicit static route. Unless otherwise configured, the metric value might change for a given interface (for example, after reboot or after modifying an interface's configuration) which might result in traffic for an external service being routed via a different interface.

1. Check if the interface has a default route. In this example we are checking for `eth1`.

```
$ ip route
```

#### Example response

```
default via 172.30.1.1 dev eth0 proto dhcp metric 101
default via 172.30.1.1 dev eth1 proto dhcp metric 102
172.30.1.0/24 dev eth0 proto kernel scope link src 172.30.1.71 metric 101
172.30.1.0/24 dev eth0 proto kernel scope link src 172.30.1.72 metric 102
172.30.0.128/25 dev kylo0 proto kernel scope link src 172.30.0.129
192.168.122.0/24 via 172.30.1.1 dev eth0 proto static
```

A default route for `eth1` is present on the second line.

2. Remove the route for `eth1`.

```
$ nmcli c mod eth1 ipv4.never-default true ipv6.never-default true
$ nmcli c up eth1
```

3. Verify that the default route is removed:

```
$ ip route
```

#### Example response

```
default via 172.30.1.1 dev eth0 proto dhcp metric 101
172.30.1.0/24 dev eth0 proto kernel scope link src 172.30.1.71 metric 101
172.30.1.0/24 dev eth0 proto kernel scope link src 172.30.1.72 metric 102
172.30.0.128/25 dev kylo0 proto kernel scope link src 172.30.0.129
192.168.122.0/24 via 172.30.1.1 dev eth0 proto static
```

The default route for `eth1` is removed.

# Bonding Network Interfaces

The official [Linux kernel HOWTO](#) documents this feature in detail.

The [nmcli's reference documentation](#) covers details specific to NetworkManager configuration.

## Note

Bonding network interfaces is intended for physical appliances, for example, CipherTrust Manager k570. It is generally not relevant to virtual CipherTrust Manager instances, as bonding is not useful in a virtual environment.

1. Start by using `nmcli` to create a bonded connection for fail-over only (the most basic type of bond).

```
$ nmcli conn add type bond con-name bond0 ifname bond0 -- bond.options mode=active-backup
Connection 'bond0' (c6661f6b-5d68-4ffa-9ae8-63f0e4224cb4) successfully added.
$ nmcli conn show bond0
bond.options: mode=active-backup
```

Alternatively, for certain networking environments such as in a virtual environment, the default setting of `fail_over_mac=0` might not be optimal and setting `fail_over_mac=1` can be an option. Please refer to the Linux Kernel HOWTO for more information. The command to create the bonded connection would then be:

```
$ nmcli conn add type bond con-name bond0 ifname bond0 -- bond.options mode=active-backup,fail_over_mac=1
Connection 'bond0' (c6661f6b-5d68-4ffa-9ae8-63f0e4224cb4) successfully added.
$ nmcli conn show bond0
...
bond.options: mode=active-backup,fail_over_mac=1
```

2. Next, use `nmcli` to list devices and add some of them to the bond.

```
$ nmcli device
DEVICE TYPE STATE CONNECTION
...
ens3 ethernet connected ens3
ens4 ethernet disconnected --
...
```

3. Create the bond connections for `ens3` and `ens4` by specifying `bond0` as their master and bring them up.

Notice that `ens3` has an existing connection profile, so delete it first to ensure only the bond connection profile is used.

```
$ nmcli conn delete ens3
Connection 'ens3' (f4e38090-6d3c-4612-b5b5-2d4c6a0c361d) successfully deleted.
$ nmcli conn add type ethernet con-name ens3-bond0 ifname ens3 master bond0
Connection 'ens3-bond0' (3d87c66e-a128-4e4f-897a-b31cf1b5fdbf) successfully added.
$ nmcli conn add type ethernet con-name ens4-bond0 ifname ens4 master bond0
Connection 'ens4-bond0' (b488d1f4-4a73-43ee-8a7f-9e992a850903) successfully added.
$ nmcli conn up ens3-bond0
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)
$ nmcli conn up ens4-bond0
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/8)
```

- Next, use `nmcli conn up bond0` to bring up the bonded interface and confirm that it received an IP address.

```
$ nmcli conn show bond0
...
connection.type: bond
...
IP4.ADDRESS[1]: 10.121.105.113/22
```

- Verify the bonded interface continues to receive and transmit data when either of the enslaved network interfaces are down (e.g. by removing the cable, disabling the port in the switch, etc.).
- It is recommended to restart the OS to verify that the bond persists across reboots. If it does not then check if a connection profile exists for any of the interfaces that compose this bond and have an equal or higher connection priority. Either delete the duplicate connection profile or increase the priority of the bonded connections so it has precedence.
- If preferred, set a static MAC address, which persists across reboots, for the bonded interface. Otherwise, CipherTrust Manager appliances can alternate the MAC address after reboot.

```
$ nmcli connection modify bond0 802-3-ethernet.cloned-mac-address "<desired_MAC_address>"
```

**Note**

This feature should be used together with `fail_over_mac=0` on the bonded network interface, the default setting.

## Virtual LAN (VLAN)

Virtual Local Area Network (VLAN) is a mechanism to virtually isolate systems located on the same physical network. If you want to configure your CipherTrust Manager network adapters to use VLAN, we recommend having supporting network equipment and also configuring VLAN on the network layer.

VLAN can be configured with any VLAN ID and it is also possible to mix non-VLAN and multiple VLANs on the same network adapter. When mixing non-VLAN and multiple VLANs be careful to keep subnets and routing correct for the machine to be able to route traffic using the correct VLAN ID.

All examples are illustrated using a static IP address but DHCP can be used assuming there is a VLAN aware DHCP server on the network.

### Switch a network adapter to VLAN only

This example disables any non-VLAN configuration on the selected network adapter and enables VLAN ID 10 with a static IP address. With this configuration, this machine can only talk to other machines using the same VLAN ID.

Delete any existing connections to make sure no stale configuration is left on the connections. If the connections are not already configured, the commands fail as expected with 'Error: unknown connection'.

```
$ nmcli conn delete ens3.10
$ nmcli conn delete ens3
```

Target setup:

- `ens3` with no network configuration
- `ens3.10` with VLAN ID 10 and a static IP address `192.168.1.2`

Setup commands:

```
$ nmcli conn add type vlan con-name ens3.10 dev ens3 id 10 ip4 192.168.1.2/24
```

Any traffic to the subnet `192.168.1.0/24` uses VLAN 10 unless there are other subnets and routes configured on this machine using the same subnet.

### Mixing non-VLAN and VLAN on same network adapter

This example enables both non-VLAN and VLAN on the same network adapter with different subnets. With this configuration, this machine can talk to other machines without VLAN on one subnet and with VLAN on another subnet.



Delete any existing connections to make sure no stale configuration is left on the connection. If the connections are not already configured, the commands fail as expected with 'Error: unknown connection'.

```
$ nmcli conn delete ens3.10
$ nmcli conn delete ens3
```

Target setup:

- ens3 with no VLAN and with static IP address 192.168.1.2
- ens3.10 with VLAN ID 10 and a static IP address 192.168.2.2

Setup commands:

```
$ nmcli conn add type ethernet con-name ens3 ifname ens3 ip4 192.168.1.2/24
$ nmcli conn add type vlan con-name ens3.10 dev ens3 id 10 ip4 192.168.2.2/24
```

Any traffic to the subnet 192.168.1.0/24 does not use VLAN and any traffic to the subnet 192.168.2.0/24 uses VLAN 10 unless there are other subnets and routes configured on this machine using the same subnet.

## Multiple VLANs on same network adapter

It is also possible to have multiple VLANs on the same network adapter. This example disables any non-VLAN configuration on the selected network adapter and enables VLAN ID 10 and VLAN ID 20 with static IP addresses with different subnets. With this configuration this machine can talk to two different VLANs using two different subnets.

Delete any existing connection to make sure no stale configuration is left on the connection. If the connections are not already configured, the commands fail as expected with 'Error: unknown connection'.

```
$ nmcli conn delete ens3.10
$ nmcli conn delete ens3.20
$ nmcli conn delete ens3
```

Target setup:

- ens3 with no network configuration
- ens3.10 with VLAN ID 10 and a static IP address 192.168.1.2
- ens3.20 with VLAN ID 20 and a static IP address 192.168.2.2

Setup commands:

```
$ nmcli conn add type vlan con-name ens3.10 dev ens3 id 10 ip4 192.168.1.2/24
$ nmcli conn add type vlan con-name ens3.20 dev ens3 id 20 ip4 192.168.2.2/24
```

Any traffic to the subnet 192.168.1.0/24 uses VLAN 10 and any traffic to the subnet 192.168.2.0/24 uses VLAN 20 unless there are other subnets and routes configured on this machine using the same subnet.

## VLAN with bonding

VLAN can also be used together with bonding. The principle is very similar to the normal non-bonding case with the difference that the bonded network adapter is used as the device. Consult [Bonding Network Interfaces](#) for

more information on setting up bonding.

## Switch a network adapter to VLAN only

This example disables any non-VLAN configuration on the selected bonded network adapter and enables VLAN ID 10 with a static IP address. With this configuration this machine can only talk to other machines using the same VLAN ID.

Delete any existing connection to make sure no stale configuration is left on the connection. If the connections are not already configured, the commands fail as expected with 'Error: unknown connection'.

```
$ nmcli conn delete bond0.10
$ nmcli conn delete ens3-bond0
$ nmcli conn delete ens4-bond0
$ nmcli conn delete bond0
```

Target setup:

- `bond0` with no network configuration
- `bond0.10` with VLAN ID 10 and a static IP address `192.168.1.2`

Setup commands:

```
$ nmcli conn add type bond con-name bond0 ifname bond0 -- bond.options mode=active-backup ipv4.method disabled ipv6.method ignore
$ nmcli conn add type ethernet con-name ens3-bond0 ifname ens3 master bond0
$ nmcli conn add type ethernet con-name ens4-bond0 ifname ens4 master bond0
$ nmcli conn add type vlan con-name bond0.10 dev bond0 id 10 ip4 192.168.1.2/24
```

Any traffic to the subnet `192.168.1.0/24` uses VLAN 10 unless there are other subnets and routes configured on this machine using the same subnet.

## Mixing non-VLAN and VLAN on same network adapter

This example enables both non-VLAN and VLAN on the same network adapter with different subnets. With this configuration this machine can talk to other machines without VLAN on one subnet and with VLAN on another subnet.

Delete any existing connection to make sure no stale configuration is left on the connection. If the connections are not already configured, the commands fail as expected with 'Error: unknown connection'.

```
$ nmcli conn delete bond0.10
$ nmcli conn delete ens3-bond0
$ nmcli conn delete ens4-bond0
$ nmcli conn delete bond0
```

Target setup:

- `bond0` with no VLAN and with static IP address `192.168.1.2`
- `bond0.10` with VLAN ID 10 and a static IP address `192.168.2.2`

Setup commands:

```
$ nmcli conn add type bond con-name bond0 ifname bond0 ip4 192.168.1.2/24 -- bond.options mode=active-backup
$ nmcli conn add type ethernet con-name ens3-bond0 ifname ens3 master bond0
$ nmcli conn add type ethernet con-name ens4-bond0 ifname ens4 master bond0
$ nmcli conn add type vlan con-name bond0.10 dev bond0 id 10 ip4 192.168.2.2/24
```

Any traffic to the subnet 192.168.1.0/24 does not use VLAN and any traffic to the subnet 192.168.2.0/24 uses VLAN 10 unless there are other subnets and routes configured on this machine using the same subnet.

## Install the CLI Toolkit

CipherTrust Manager includes a CLI toolkit, named **ksctl**, that can be downloaded and run locally to control a remote CipherTrust Manager.

### Note

**ksctl** is designed to be run from a remote system, not on the CipherTrust Manager itself.

**ksctl** exclusively uses the REST API to communicate with CipherTrust Manager, so anything you can do with the CLI 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 adding users and groups, and end-user functions, such as creating keys.

To get started with **ksctl** you must do the following:

1. [Download and unzip the ksctl\\_images.zip file.](#)
2. [Setup the ksctl-os file](#) for your system. This includes setting some basic configuration variables for ksctl authentication.
3. (Optional) Further set [ksctl configuration variables](#), if desired.
4. [Run ksctl.](#)

## Download and unzip the ksctl\_images.zip file

1. Enter the IP address of your CipherTrust Manager system in your browser.
2. If you are logged into CipherTrust Manager, select the **API** link at the top right. If you are logged out, select **API & CLI Documentation** at the bottom right.
3. Select the **\*\* CLI Guide page and click on Download CLI\*\*** button:



4. Unzip the **ksctl\_images.zip** file.

Example set of available images in **ksctl\_images.zip** file:

```
ksctl-darwin-amd64
ksctl-linux-amd64
ksctl-win-amd64.exe
```

## Setup the *ksctl*-os for your system

The **ksctl** utility can be set up on [Windows](#), [Linux](#), and [macOS](#).

### Windows

1. After unzipping, rename **ksctl-win-amd64.exe** to **ksctl.exe**.

#### Note

**ksctl.exe** is a single executable with no dependencies, so it can be run from anywhere.

2. Move the **ksctl.exe** file to a folder easily accessed by the local command prompt.

#### Note

You can also add the location of the **ksctl.exe** file to your PATH variable or copy **ksctl.exe** to a location already in your PATH.

3. Create a **ksctl** configuration file using the local command prompt **cmd.exe**:

- a. Create a **.ksctl** directory in your **%HOMEPATH%** directory:

```
mkdir %HOMEPATH%\ksctl
cd %HOMEPATH%\ksctl
```

- b. In the **%HOMEPATH%.ksctl** directory, create a **config.yaml** file with the following items:

```
KSCTL_USERNAME: admin
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<ciphertrust manager ipaddress> or https://<ciphertrust
```

```
t_hostname>
KSCTL_NOSSLVERIFY: true
```

### Tip

Enter `set homedir` to find or confirm your `%HOMEPATH%` directory.

## Linux

1. After unzipping, rename the file of choice to **ksctl** and move it to a directory within your PATH.

### Note

**ksctl** can be run from any directory.

2. Create a **ksctl** configuration file:
  - a. Create a **.ksctl** directory in your **\$HOME** directory:

```
mkdir $HOME/.ksctl
cd $HOME/.ksctl
```

- b. In the **\$HOME/.ksctl** directory create a **config.yaml** file with the following items:

```
KSCTL_USERNAME: admin
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<ciphertrust manager ipaddress> or https://<ciphertrust_hostname>
KSCTL_NOSSLVERIFY: true
```

3. Bash completions can be generated by typing `ksctl bashcomp` at the bash prompt and following the instructions.

## macOS

1. After unzipping, rename **ksctl-darwin-amd64** to **ksctl** and move it to a directory within your PATH.
2. In Finder, right-click or control-click the **ksctl** executable and select **Open**. You cannot open the executable by double-clicking.

**Note**

For some versions of macOS, a prompt appears indicating that macOS cannot verify the developer. You need to dismiss the prompt and navigate into Security & Privacy settings to manually allow **ksctl**.

3. On the prompt, click **Open** to proceed.
4. Create a **ksctl** configuration file:
  - a. Create a **.ksctl** directory in your **\$HOME** directory:

```
mkdir $HOME/.ksctl
cd $HOME/.ksctl
```

- b. In the **\$HOME/.ksctl** directory create a **config.yaml** file with the following items:

```
KSCTL_USERNAME: admin
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<ciphertrust manager ipaddress> or https://<ciphertrust
t_hostname>
KSCTL_NOSSLVERIFY: true
```

**Tip**

Enter `env | grep HOME` to find your **\$HOME** directory.

## ksctl Configuration Variables

Summary of **ksctl** configuration variables for the `config.yaml` configuration file:

```
KSCTL_USERNAME: <your username>
KSCTL_PASSWORD: <your password>
KSCTL_URL: https://<ciphertrust manager ipaddress> or https://<ciphertrust_localh
ost>
KSCTL_NOSSLVERIFY: <true or false. The default value is false.>
KSCTL_CONNECTION: <LDAP connection name>
KSCTL_TRUSTED_CA_FILE: <Filename of the Trusted CA. Required when KSCTL_NOSSLVERI
FY is set to false.>
KSCTL_SERVER_NAME: <Used to verify the hostname on the server certificate. Requir
ed when KSCTL_NOSSLVERIFY is set to false, the hostname is not present on the cli
ent workstation `etc/hosts` file, or the hostname is not mapped on the local DNS
```

```

server>
KSCTL_VERBOSITY: <true or false for verbosity>
KSCTL_JWT: <JWT>
KSCTL_RESPFMT: <response output format, the default "json" is the only supported
value at present>
KSCTL_TIMEOUT: <Time limit in seconds to wait for a response. Set to `0` for no t
imeout. The default value is 30, if not set.>
KSCTL_AUTH_DOMAIN: <The ${cm} domain where the user is authenticated>

```

**ksctl** is easiest to use with the configuration file (config.yaml). However, those parameters can be overridden by using these flags on the command line:

```

--user (replaces KSCTL_USERNAME)
--password (replaces KSCTL_PASSWORD)
--url (replaces KSCTL_URL)
--connection (replaces KSCTL_CONNECTION)
--nosslverify (replaces KSCTL_NOSSLVERIFY=false)
--nosslverify=false (replaces KSCTL_NOSSLVERIFY=true)
--trusted-ca-file (replaces KSCTL_TRUSTED_CA_FILE)
--server-name (replaces KSCTL_SERVER_NAME)
--verbose (replaces KSCTL_VERBOSITY=false)
--verbose=false (replaces KSCTL_VERBOSITY=true)
--jwt (replaces KSCTL_JWT)
--respfmt (replaces KSCTL_RESPFMT)
--timeout int (replaces KSCTL_TIMEOUT)

```

### Note

- If `KSCTL_VERBOSITY` and `KSCTL_NOSSLVERIFY` are set to true in the config.yaml file, you must use `--verbose=false` or `--nosslverify=false` flags to override them. This is because these two settings are boolean values.
- We recommend setting `KSCTL_TRUSTED_CA_FILE` and `KSCTL_SERVER_NAME` to verify the CipherTrust Manager [web interface](#) server certificate. In most cases, `ksctl` communicates with the web interface. When the [disk is encrypted and in the preboot stage](#), `ksctl` communicates with the preboot interface. In this state, we recommend using the `--trusted-ca-file` and `--server-name` flags to provide values matching the preboot interface certificate. It is also possible but not recommended to use `--nosslverify` to skip certificate verification.

## Authenticating to ksctl

Setting the `config.yaml` as indicated in [Setup the ksctl OS for your system](#) means that most `ksctl` commands authenticate you as the admin user.

If you wish to authenticate as another user you can either:

- Run `ksctl login --user <username> --password <password>` command. All subsequent commands in the session keep the authentication context, and are run as the logged in user.

- Use `--user <username>` and `--password <password>` flags on each individual command.

As well all commands are run in the root domain by default. If you want to authenticate to a **child domain**, use the `--domain <domain_name>` flag. You can either:

- Run `ksctl login --domain <domain_name>` command. All subsequent commands in the session keep the authentication context, and are run in the specified domain.
- Use `--domain <domain_name>` flags on each individual command.

## Run ksctl

### Note

This is an example only; not all supported CLI commands and flags are shown.

1. At the command line prompt, enter **ksctl**. You will get the following example output:

```
ksctl
```

```
This command line utility can be used to interface with a CipherTrust Manager device.
```

```
Configuration parameters can be used to pass in the username, URL, etc. These parameters can be passed to the application in several ways.
```

```
When the configuration parameters are available from multiple sources, the order of precedence (highest to lowest) is:
```

1. Command line arguments
2. Configuration file passed via the command line
3. Environmental variables
4. Configuration file in the default location

```
The parameters passed in as command line arguments override everything else.
```

```
The structure of the configuration file must be based on 'config_example.yaml'. The name of the default configuration file is 'config.yaml'. It should be located in the '.ksctl' subdirectory of the user's home directory.
```

```
On Linux, this file should be located in: '~/.ksctl/config.yaml'.
```



On Windows, this file should be located in: 'C:/ksctl/config.yaml'.

If defined, proxy environment variables (i.e. 'http\_proxy') will be honored for all functions except 'init getpw'. 'init getpw' internally uses ssh which means that ssh must be configured to be proxy aware for the users local operating system.

#### Usage:

```
ksctl [command]
```

#### Available Commands:

|                |                                                                  |
|----------------|------------------------------------------------------------------|
| akeyless       | Akeyless Gateway                                                 |
| alarms         | Manage Alarms. List, state and clear commands.                   |
| backup         | Perform system or domain database backups.                       |
| backupkeys     | Manage Backup keys. Used for securing backups.                   |
| banners        | Manage Login Banners.                                            |
| bashcomp       | Generate bash completion script file for all commands and flags. |
| ca             | Manage Certificate Authorities and Certificates.                 |
| cckm           | Manage CipherTrust Cloud Key Manager.                            |
| changepw       | Change a user password.                                          |
| client-records | Get client record(s).                                            |
| clientmgmt     | Create Client Tokens and Register Clients.                       |
| cluster        | Manage Clusters.                                                 |
| .              |                                                                  |
| .              |                                                                  |
| .              |                                                                  |

#### Flags:

|                                   |                                                                                             |
|-----------------------------------|---------------------------------------------------------------------------------------------|
| <code>--auth-domain string</code> | The CipherTrust Manager Domain Where the user is created.                                   |
|                                   | By default the auth domain is root or provide the domain where user created.                |
| <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 except 'tokens create' command.      |
|                                   | By default the command will operate in the root domain or the domain the user is logged-in. |

```

-h, --help help for ksctl
--jwt string The JSON Web Token (JWT) - access token can be passed instead
 of user/password (optional). 'ksctl tokens create' creates a JWT.
--nossllverify Do not verify the certificate for SSL/HTTPS authentication (not recommended)
--password string CipherTrust Manager Server User Password. Do not use this flag to enter the password (masked) from terminal.
--respfmt string Response Output format (json is the only supported value at present, optional) (default "json")
--server-name string Name of the Server which is used to verify the hostname on the returned certificates unless nossllverify is given.
--timeout int Timeout in seconds for TCP connection attempts
--token string The refresh token returned from the login command to be passed instead of user/password (optional).
 'ksctl login' creates a token and writes it to the config file.
--trusted-ca-file string Filename of the Trusted CA to be read from.
--url string CipherTrust Manager Server URL
--user string CipherTrust Manager Server User Name
-v, --verbose Provide verbose output while executing command (optional)

Use "ksctl [command] --help" for more information about a command.

```

2. Try creating a key by entering the command `ksctl keys create --autoname`:

```
ksctl keys create --autoname
```

This will create output similar to the following:

```

{
 "id": "b979d9c7-1374-4b97-a274-dfaa20f8139b",
 "uri": "kylo:kylo:vault:keys:keys-f0f3e1lhqh5r7xavk3l0-rx-ktkmwgzf3-rudppxr4-v0",
 "account": "kylo:kylo:admin:accounts:kylo",
 "application": "ncryptify:gemalto:admin:apps:kylo",
 "devAccount": "ncryptify:gemalto:admin:accounts:gemalto",
 "createdAt": "2016-12-15T15:35:51.322996318Z",
 "name": "Keys-f0F3eLlhQh5R7XAVK3l0_rx-KtkmWGzF3-rUDPpXRW4",

```

```

"updatedAt": "2016-12-15T15:35:51.322996318Z",
"material": "",
"usage": "blob",
"meta": null,
"version": 0,
"algorithm": "AES",
"size": 256,
"format": "raw",
"unexportable": false,
"undeletable": false,
"publickey": ""
}

```

## Changing the Initial Password

The username of the initial CipherTrust Manager user is "admin". This user becomes the initial Application Administrator. By default, the initial password is hardcoded to "admin" in lowercase. You can also auto-generate a password using cloud init. Regardless of whether the initial password is hardcoded or auto-generated, it must be changed before using the CipherTrust Manager.

### Note

To better understand the different user groups within the CipherTrust Manager, refer to [Groups](#).

### To change the initial password

#### If using the GUI:

On initial log in, the Application Administrator is forced to change the password.

#### If using the CLI:

Enter the new password using the following command:

```

ksctl changepw --url=https://<ip-or-host> --user admin --password <current_password> --new-pw <new_password> --confirm-pw <new_password>

```

The default password policies are:

```

Min length: 8
Max length: 30
Min number of upper cases: 1
Min number of lower cases: 1
Min number of digits: 1
Min number of other characters: 1

```

## Retrieving an Auto-generated Initial password

If the initial user's password is auto-generated through cloud-init, it must be retrieved by the System Administrator "ksadmin" via SSH or via the CLI.

The initial password for the initial Application Administrator can be auto-generated using a cloud-init configuration file containing the following setting:

```
#cloud-config
keysecure:
 initial-user:
 generate-password: true
```

Full details of cloud-init deployment are available in [Cloud Init Deployment](#).

### To retrieve the initial password for the Application Administrator using SSH.

The initial password is written out to the file `/var/keysecure/password/out`. Retrieve it using SSH:

```
ssh ksadmin@{addr} cat /var/keysecure/password/out
```

If the ssh key used to launch the instance is not loaded in your ssh-agent, add the key file argument:

```
ssh -i {key file} ksadmin@{addr} cat /var/keysecure/password/out
```

#### Caution

After retrieving the initial password for the Application Administrator, the Application Administrator must log in and change it immediately. Failure to do so could allow an attacker to compromise the system.

### To retrieve the initial password for the Application Administrator using the CLI:

On Linux-like systems that have the `ssh` command line tool available, `ksctl` can be used to execute the `ssh` command to retrieve the initial password.

If you installed the CLI, you can use it to acquire the initial password as well:

```
ksctl init getpw --url https://<ipaddr> --identityFile <private key> --user admin
```

#### Caution

After retrieving the initial password for the Application Administrator, the Application Administrator must log in and change it immediately. Failure to do so could allow an attacker to compromise the system.

# Network Time Protocol Server Configuration

Proper time synchronization on the CipherTrust Manager appliance is crucial for correct functioning of the following features:

- Configuring a Hardware Security Module (HSM) as Root of Trust
- Clustering multiple CipherTrust Managers
- Adding external connections to Connection Manager
- External **user management** protocols such as OpenID Connect (OIDC) or Lightweight Directory Access Protocol (LDAP).
- Client connector applications such as Data Discovery and Classification.
- Integrations with external clouds such as Google.

We strongly recommend configuring at least one Network Time Protocol (NTP) Server for the CipherTrust Manager immediately after deployment to ensure accurate time synchronization. In general, NTP Server configuration ensures that communications between a CipherTrust Manager and any external entity will work.

## Add an NTP Server

For authenticated NTP servers, you must provide an encryption key. The supported key type digest algorithms are MD5, SHA-1, SHA-256, SHA-384, or SHA-512. SHA-256 is the default value applied if you provide an encryption key but do not specify a key type digest algorithm.

If the NTP server is unauthenticated, the IP address or hostname is sufficient.

## Add an NTP Server with ksctl CLI

Use the following command to add an NTP server:

```
ksctl ntp servers add --host <NTP_server_host_or_IP_address> [--key <encryption_key_material> --key-type <key_type _digest_algorithm>]
```

For authenticated NTP servers, provide `--key`. You must additionally specify the `key-type` if the encryption key type digest algorithm is not SHA-256.

## Add an NTP Server with CipherTrust Web UI

1. Navigate to **Admin Settings > NTP**.
2. Click **+ Add NTP Server**.
3. Enter in an **NTP Server hostname**.
4. For an authenticated NTP Server, enter the following values:

- In the **NTP Key** field, provide the symmetric key material.
- Select the **Key Type Digest Algorithm** value from the drop-down list. The available algorithms are MD5, SHA-1, SHA-256, SHA-384, or SHA-512. SHA-256 is the default value applied if you do not make a selection.

5. Click the **Add NTP Server** button.

## Delete an NTP Server

In the CLI:

```
ksctl ntp servers add --host <NTP_server_host_or_IP_address>
```

In the UI:

1. Navigate to **Admin Settings > NTP**.
2. View the **Configured NTP Servers**, and find the server you wish to delete.
3. Click the trash can icon to the right of the server hostname or IP address.

## Starting Services After Deployment

Physical appliances and private cloud instances include an initial SSH key for the System Admin "ksadmin" to use during launch. After launching, this key must be replaced in order for the CipherTrust Manager to start all of its services and become fully functional.

Replacing the SSH key using `ksctl` is a one-time operation during deployment. You cannot replace the key a second time. However, you can always add additional public SSH keys to SSH into CipherTrust Manager. See [Manage public SSH keys for ksadmin](#) for more details.

### Note

If you launched a Virtual CipherTrust Manager from a public cloud such as AWS, Google Cloud, Microsoft Azure, or Oracle Cloud, the SSH key you provided at launch does not need to be replaced.

### To replace the SSH key using the GUI

1. Create an SSH key pair outside of CipherTrust Manager. Your public key must be an RSA key in the OpenSSH format. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security. The corresponding private key can be OpenSSH, PKCS1, or PKCS8 format.
2. Browse to the CipherTrust Manager IP address.
3. If prompted to enter the a new **SSH Public Key**, paste in your SSH public key in the box provided and then

---

select **Add**.

The **Log In** screen appears, verifying that the SSH key has been successfully replaced.

### To replace the SSH key using the CLI

None of the CLI commands shown here require authentication.

1. Create an SSH key pair outside of CipherTrust Manager. Your public key must be an RSA key in the OpenSSH format. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security. The corresponding private key can be OpenSSH, PKCS1, or PKCS8 format.
2. Check if the existing SSH key needs replacement.

```
ksctl services status
```

#### Sample Response:

```
{"status": "bootstrap", "services":[] "messages":
 ["The SSH public key for the ksadmin user must be replaced before the ${cm}
 will be functional"]
}
```

3. Upload the public key to CipherTrust Manager:

```
ksctl ssh keys add -k "A..."
```

CipherTrust Manager Upon successfully adding an authorized SSH key, the CipherTrust Manager services will start working momentarily.

4. Check that the services have started.

```
ksctl services status
```

#### Sample Response:

```
{"status": "started", "services":[]}
```

### To replace the SSH key using the API

None of the API calls shown here require authentication.

1. Create an SSH key pair outside of CipherTrust Manager. Your public key must be an RSA key in the OpenSSH format. We recommend RSA 4096, with RSA 2048 as a minimum size for adequate security. The

corresponding private key can be OpenSSH, PKCS1, or PKCS8 format.

2. Check if the default SSH key needs to be replaced.

```
GET https://{address}/system/services/status
```

#### Sample Response:

```
{"status": "bootstrap", "services":[] "messages": ["The SSH public key for the
ksadmin user must be replaced before the ${cm} will be functional"]}
```

3. Upload the new SSH public key.

```
POST https://{addr}/system/ssh/keys {"key": "A..."}
```

4. Check that the services have started.

```
GET https://{addr}/system/services/status
```

#### Sample Response:

```
{"status": "started", "services":[]}
```

## Accessing the System before Services are Started

If you want to access the system before the services have been has started, you can use the hard-coded SSH key. The default key is show below:

```
-----BEGIN RSA PRIVATE KEY-----
MIIJKgIBAAKCAgEAnwGtotjvpBAukGHp/Y2xCFcUyPrm9KiMFyjreIJUkYe3yWKw
cFdBMh8cn8slAo7B/tIN/tYUnO8u+okx3apQDne41miCYeARXlvq3F8V2rVVtEPk
7d2vQ4TMI9CXfYNeJyil1yFlyMg4CDVnYM2bcJefFkoWZRDAnGG++ZHxKSc8Y/ge
nJH2RyLmCJYaYf6K6Y84ELtpTY1ty6Fd1t1mJanTIs4rAtz0cNbHI+1rK05ztiJO
/EOgywffogNXUs7lKHggk28K0ggl9OgJztQ+tTerdFlVcKwAzXrDOsHv6KHWzr5r
QYtl9nmpGpHjGwNuXdx3fdgQLr/ZQzj5QU1VqZlCg6aMiMFLMVkP3dnTK5om7RDv
DgPmTkj5De1dagBJrrhYsDkU6UgQwosq7gQdGZqLuY8WtkKqBYTvP/GTPG7SBnJp
+muQu3E+7WHBvt+dPG1wiygFua5UtZcyldSdithUSDjYrPFeDW1aSQb6YsTHxklx
+AvuEoX6EBMA43yzFS9fEQZpmdgAm0WXEZcQ0ZKcW322PJ2HEuAeclQuNzcUsim
H+nLfBCKRQIaRpgRttKoqnzTkGtuDWPbxUDoq5N78XQ1CQDZNeNsBL9b8vpAGvxT
sSAC/dBOPbRUU1ss3f9pf5vP3IHlpm/Vnbe+mLnwu8YNcW1U4DQf+m6SfTMCAwEA
AQKCAgBtwvLRv17tqh36nRyhYwYp87LtagW0LgVG14C014M5Uq6T3gReCF2A8+kW
hWzu7w0LkoUGvVFNY+7SE5LQqZjJ4+s8VrhqR/q1YE4oAZtcdoDLB3MooDc0LJ+e
uYdZpS2BaqCngk54432KbRMHx+bd0k0fXdVHvuC3EuJ59SrJxcPyXn0DJ98SDA0v
MC7WJphN0gepxTVTy6VXQ/U7NVgzR5/wDtGyEETM//ZH3sR0CJvnxcvvnJ+Kh8zg
Ud4d06Rs4z/ZhnVQyf8pqGpc2rb08bbxghlwxctoDK5oCShLNizKWrg/brLcCe+p
37seEqhFMsNpo5TIMLcRqGqnoXLXn04B4cA9IM8wWmVvnyxndKGYz5aYn+t37Un+
G7q55wJYgZrpR6JS1hGdyQqiavLvMJP4cVmx9YpxuB87zir5m2xfq5L0TovukqJw
```



```

i1TRNng8Ct25RtmmFsOrikDDccGo8LXiDrnwPQr6eDifp++tpsGvIwGFNohQNeKg
U/F3FAiCtzbagsZtGHfK5EdGij0EWfpyHXDginLlCpvIws010KJDyeDjSgVjrba
Qb03ogcP9ynRAevjdWSNi70pCCMs9kL3+q8tLEPO3dd/VrZIIHTROy7iXLj/a7cC
cfrR6mqoG7yEdO26yA3sASsFvrYkG8k/GNpRWJQZsherKO9wQKCAQEAY2xcEW8l
C52GtksLSN2quxHKUlm+JbBWtE3OFewCd0MxQ6MgW2dohTGj7C8PgfgaNLdXhTiW
58MgZis4/vymPX73ybobG9ExKV59/fW9RA+TS7cxF594AP0kVmMeRXhleJCJuHiC
uOB/c5Kjduy4VyvPuyI4phXAfhZRu6yWp5RziP9uoy9quub4LCZWVXO9dXWEprU
M9+TkNiBjOmCRR2g4Pb4cfG4+GemVjcdQHR7D10vuYZJOST2Orwif2zHE/kJb6Uo
KnLisUs/loFkpH98GM5StdKuT4EKPz6DCyz8NHzfddq2IBhYm/NaNbo17IgNM9k2+
GMTr3fm0HYZmTwKCAQEAYBp0qZwmoozW5pM2AUuoyxzYlgZk+Nn64ZI9xaW0TYRX
rbrMA9HXinyueRc83S/ip4XlHej37fVLgXK+8DIcnqa/sZq0vvPdgAySdO4wmL3
Yq0PbwqxfanDmtPTcTz84jt9SXX23jRbNdXRG+UpvD8U58K7O5O+djRKM5c5gPhV+
ehdIm78rylK5D+XlF5f+9oxwLhnc3MC8+1uVJbrmisGLjTibjvZ+M1zUw+3NMZ/w
N8N9DO2MLhhh6/6EQmNfcUB13GmfuGDIzA3dBt6jB8UjWQpW+FybKtflIqK/9+wK
em0KnW2b+JQiJa0SiUEkxdUjPIeUUoujV8Zc1SF13QKCAQEAuA0ywBNv0TKV0BQL
+AGXl+pgxGGuSStjR3+eFveSw5Vn3WqavBiz2M8xIWDzBwdWvpXZ8E9jOg6yIb65
svuk7b8TmTHP9Tsw390+4aiVth7hhoAZSuZeDYb8eaz0lqZ4KL2d3eorXFrC9i6W
9+L2Zuic6xPrbVoA8BQBQVPGe5klb56s27jrOHAB2vBU7Lvrj3rknYfyU9glI1P
TS7dTuufw+uwnme2lAoz1Y2xIZlCPoRxycpWNgO81Sejjuc8uDV69CjaTo3jzcXn
oqUwCiXOGiMrcpdMX2LBgrdsmY/6IQNarGYVzGv+mn87LiuUXRHfcHqmpIKHZi+n
Z49SwQKCAQEASRKILhcaMTnEaaHXZmoXYjWSi5/49iWe1sHcm+w4k2gLVpc4NiEZ
RVlZ43/mwiY/u1LHHyKrUw1/1Xs+77rNtNQ+GDppCW7/dQJTffU3R/zfwZg3UiH6
UqNJjtboAwrfFStjgQX9AkD+fbNKKg8uUYDY9QXjPaNOJbe7+DsR2DkJBO1ZuUkA
bTeqWCDUNB6y5mX8Vy9XvJlqIlw2rX6oyBzhawSu02HY54a8WO00aTeJES+S10AW
jivRa058p2rcaJGMOpHVXw12ZuvVrWCSkCupTf/SWlF1li8aev3eNnvQAepAPHNk
VCJx3Tk8b+0Pp2kml0e6vajnkKLePlQgmQKCAQEAhwzJQZAL8Rzcpn6BNKzkXJ/c
OOCj/YR38Ka4wkU4BSflzZjfrWUDsPlM8AfcuOeJEqRK4KE0pZSiTHIe25y+WTMt
Ot+PCjJhTTFAnbH7M73sEYD0Urz3B+5/3pQpMIxIDlCSKGZWroMut6rH8HYyn4e6
K1OuVwKRWQR4LnmJvgU/xE6kyjiXdkfHFnEedjCYor2ROhHUP1uEzC223dYilqD3
+a2oVBdjOuEIUGap0DJ9zMCCArKjRCF5Rpf8/8/CVYVLEdL+vgEiiAP7Gv0Puw30
qXIklPUJAlSmm2rXjGim3DRNrnY1CwLH9pOfaRgtZ0inyOfowAouRhtisxz6fQ==
-----END RSA PRIVATE KEY-----

```

### Note

This default key can be replaced via cloud-init.

## System Configuration Utility

The CipherTrust Manager has a 'CLI based' System Configuration Utility (**kscfg**). The "ksadmin" user can remotely access the CipherTrust Manager **kscfg** utility in a private cloud deployment by accessing the Console, or in a physical appliance deployment, by directly connecting to the appliance's console port and using your ksadmin password.

## Logging in as ksadmin user

To log in as **ksadmin**, you must first connect to the CipherTrust Manager console.

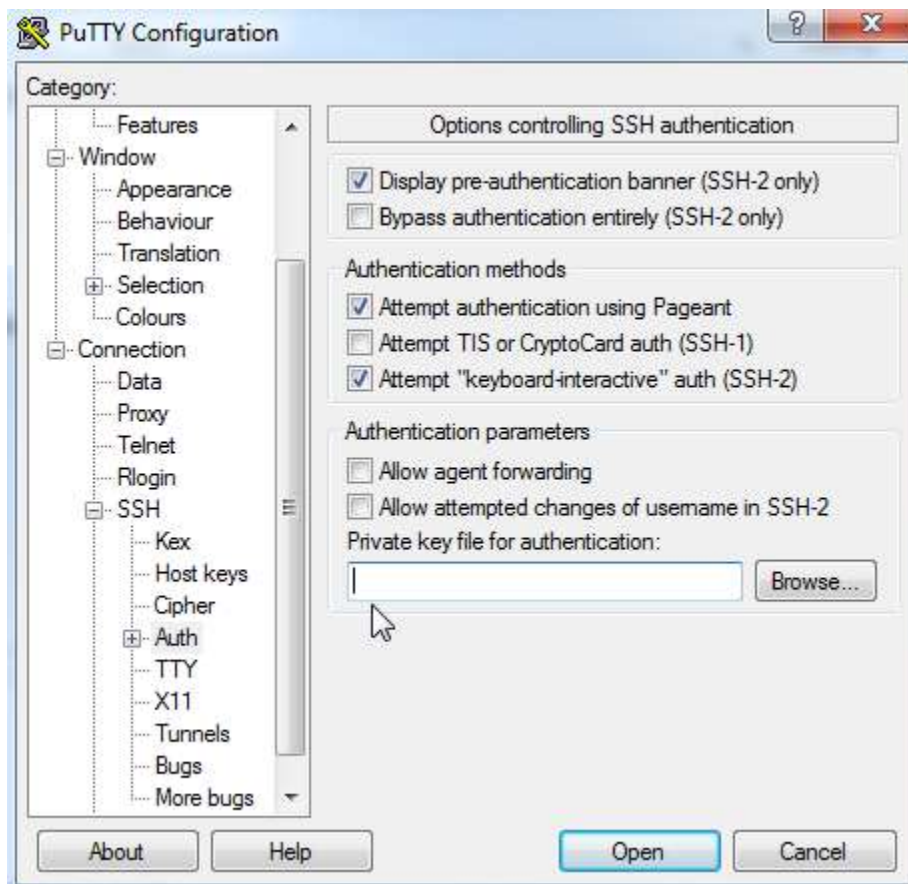
## To connect and log in to the console for public cloud deployments

Consult public cloud documentation on SSH access to images. An SSH key for ksadmin login was provided during Virtual CipherTrust Manager launch on a supported public cloud.

## To connect and log in to the console for private cloud deployments

Using SSH, you can remotely connect to the console port of a CipherTrust Manager instance deployed in a private cloud (e.g. VMware vSphere and HyperV).

1. Using an SSH utility (e.g. PuTTY) select an SSH session and enter the IP address assigned to the CipherTrust Manager instance during deployment. This is the same IP address used to browse to the GUI.
2. If using PuTTY, make sure your SSH keys are in ppk format. If they are in PEM format, you can convert them to ppk (e.g., using PuTTYgen utility).
3. Using the SSH utility, select the path to your SSH Private Key you will use to authenticate the session.



4. Select **Open** to start the SSH session.

## To connect and log in to the console (for physical appliance deployments)

Using a serial cable, you can directly connect your console device (e.g., laptop) to the console port of a physical appliance (k470 and k570).

1. Connect the serial cable from your console device to the physical appliance console port.
2. Log in to the physical appliance as `ksadmin` user using the password you created during Appliance Initialization; refer to [Appliance Initialization](#).

## Using the `kscfg` utility

The CipherTrust Manager **kscfg** utility can be used to retrieve network interfaces (NICs) configurations and values, and to perform a hard system reset or a factory reset.

All configurable network interfaces are always listed and are the same as those available from standard Linux network interfaces utilities such as `ip`, `ifconfig`, and `nmcli`. The network interface names from **kscfg** match the network interface names from the operating system.

A network interface contains two configurable families: `inet` (IPv4) and `inet6` (IPv6). Their methods are:

- `"none"`

For a disabled network family.

- `"dhcp"`

Use DHCP to automatically acquire a network address. (`"auto"` might be preferred for IPv6.)

`"static"`

Statically set a network address.

- `"auto"`

Automatically setup IPv6 from the network environment. (IPv6 only)

## Commands

### To view the available `kscfg` commands

**Example:**

```
kscfg --help
```

**Response:**

```
Command-line interface application for configuring the CipherTrust Manager.

Usage:
 kscfg [command]

Available Commands:
 cluster Cluster configuration
 cluster-report Cluster report generation, used for analyzing/diagnosing cl
uster issues
 help Help about any command
 hsm Manage HSM configurations
```

```

net CipherTrust Manager network configuration
syslog Manage syslog forward configurations
system CipherTrust Manager system commands

Flags:
-h, --help Help for kscfg
-v, --verbose Provide verbose output while executing command (optional)

```

## SSH Port Configuration

### To get the current SSH port

#### Example:

```
kscfg system ssh get
```

#### Response:

```
{
 "Port": "22"
}
```

### To set the SSH port

#### Usage

```
kscfg system ssh set [flags]
```

#### Flags:

```

-h, --help Help for set
--port string Specify the SSH port

```

#### Example:

```
kscfg system ssh set --port 9001
```

#### Response:

```
{
 "Port": "9001"
}
```

## Generate a Cluster Report

Generate a cluster report for detailed information about the current cluster node's health and activity to help troubleshoot issues. A tar.bz2 file is returned, which you can then move to a different system to unpack. You cannot unpack the resulting tar.bz2 file locally.

For routine monitoring, we recommend [cluster status displays](#) available in CipherTrust Manager's GUI, CLI, and REST API.

### Usage:

```
kscfg cluster-report [flags]
```

### Flags:

```
-h, --help Help for cluster-report
```

### Example:

```
kscfg cluster-report
```

### Response:

```
Cluster data collection completed:
/home/ksadmin/cm-report-20250708-192751-3392683317177.tar.bz2
```

## Cluster Allowlist Configuration

To configure the list of IPs that are allowed to the cluster port 5432, refer to [Cluster Allowlist](#).

## Network Configuration

### To list the available network interfaces

#### Example:

```
kscfg net interfaces list
```

#### Response:

```
{
 "skip": 0,
 "limit": 0,
 "total": 3,
 "resources": [
 {
 "name": "eth0",
 "inet": {
 "method": "static",
 "ip": "10.121.105.137",
 "netmask": "255.255.252.0",
 "gateway": "10.121.104.1",
 "dns": [
 "172.16.2.12"
]
 },
 "inet6": {
 "method": "none"
 }
 }
]
}
```

```

 },
 {
 "name": "eth1",
 "inet": {
 "method": "dhcp",
 "ip": "10.121.105.81",
 "netmask": "255.255.252.0",
 "gateway": "10.121.104.1",
 "dns": [
 "172.16.2.12",
 "10.121.8.7",
 "172.16.2.13"
]
 },
 "inet6": {
 "method": "none"
 }
 },
 {
 "name": "eth2",
 "inet": {
 "method": "none"
 },
 "inet6": {
 "method": "none"
 }
 }
]
}

```

## To view information on a specific network interface

### Usage:

```
kscfg net interfaces get [flags]
```

### Flags:

```

-h, --help Help for get
-n, --name string Name for the network interface, for example, 'enp0s25'

```

### Example:

```
kscfg net interfaces get -n eth0
```

### Response:

```

{
 "name": "eth0",
 "inet": {
 "method": "static",
 "ip": "10.121.105.137",
 "netmask": "255.255.252.0",

```

```

 "gateway": "10.121.104.1",
 "dns": [
 "172.16.2.12"
],
 "inet6": {
 "method": "none"
 }
 }
}

```

## To modify the configuration of a specific network interface

### Caution

This operation has been deprecated. Please use NetworkManager's `nmcli` tool to modify a network interface's configuration; refer to: [Network Configuration Tutorial](#).

### Usage:

```
kscfg net interfaces modify [flags]
```

### Flags:

```

-d, --dhcp Use DHCP for the network interface. Deprecated - use "method" instead.
-r, --dns string IP addresses of the DNS servers (comma separated), or "" to unset and use entries in /etc/resolv.conf.
--force-gateway string Force system default gateway update, i.e. overwrite system default gateway when this device is brought up. By default a network interface will only set the system default gateway if is not already set. This feature can be used to force a specific network interface to be used for outgoing traffic initiated from the machine itself. Set to "yes" to enable and "no" to disable.
-g, --gateway string Default gateway, or "" to unset.
-h, --help Help for modify
-4, --inet Use IPv4 for the network interface. (default true)
-6, --inet6 Use IPv6 for the network interface.
-i, --ipaddress string Static IP Address.
-e, --method string Method for obtaining an IP. Accepted inet values are dhcp, none, or static; inet6 values are auto, dhcp, none, or static. (default "static")
-n, --name string A network interface name such as 'enp0s25'.
-m, --netmask string Subnet mask. IPv4 must be an IP (e.g. 255.255.255.0). IPv6 must be the number of bits (e.g. 64).

```

### Example:

```
kscfg net interfaces modify --name eth0 --ipaddress 10.121.105.27 --netmask 255.255.252.0 --gateway 10.121.104.1 --dns 172.16.2.12
```

### Response:

```
{
 "name": "eth0",
 "inet": {
 "method": "static",
 "ip": "10.121.105.27",
 "netmask": "255.255.252.0",
 "gateway": "10.121.104.1",
 "dns": [
 "172.16.2.12"
]
 },
 "inet6": {
 "method": "none"
 }
}
```

## Configure Reserved Internal Network

To configure the reserved internal network:

```
kscfg net internal modify -s <ip_address>
```

### Example

```
kscfg net internal modify -s 172.30.1.0
```

The above command reserves the subnet of the internal network. The "ip\_address" parameter must end with a *zero octet* and should be a private IPv4 address as defined by RFC-1918, which includes the following ranges:

10.0.0.0 - 10.255.255.0

172.16.0.0 - 172.31.255.0

192.168.0.0 - 192.168.255.0

The configuration is applied to the node where the command is executed. If the setting is required for the entire cluster, this command must be executed on each node within the cluster.

### Caution

If the command fails along with the operation, restart the services or reboot the system. If the operational failure continues, please reach out to support for assistance.

## View Reserved Internal Network

To view reserved internal network:

```
kscfg net internal show
```



---

## Response

```
{
 "reserved": "172.30.1.0/24"
}
```

This command shows the reserved subnet for the internal network.

## Forward Host Logs to an External Syslog Server

### Note

This is a different connection than the [syslog server configuration](#) used to forward audit records.

The host logs available to forward include:

- Messages in the `auth.log` and `authpriv.log` syslog facilities.
- Messages with the `CLOUDINIT` tag.
- Messages from the `host-daemon`, `sshd`, `smartd`, `ipmiev`, `ks_support`, `sudo`, and `upgrade` programs
- A subset of messages from `kernel`. Messages from `iptables`, `apparmor`, `usb`, `usbcore`, `usbhid`, `usb-storage`, and `IPMI` are sent.

The displayed log filenames and destination directory on the syslog server depend on the server's configuration.

The host logs are also included as part of the [downloadable debug logs tar.gz.zip](#).

You can use `kscfg syslog forwarder` commands to forward the host logs to a remote syslog.

## To add a remote syslog server

The only required flag is `server`.

We recommend setting TLS for the new syslog forwarder configuration transport protocol. A trusted certificate authority (CA) certificate is required for TLS transport. This is a CA which directly or indirectly signs the syslog server certificate. If you desire mutual authentication, where the syslog server verifies CipherTrust Manager, you can provide a client certificate and client key.

When the syslog connection uses TLS, CipherTrust Manager automatically checks the Certificate Revocation List (CRL) associated with the trusted CA every time a connection is established. This check ensures that the server certificate presented by the external syslog server has not been revoked.

**Note**

Ensure that the general network configuration allows CipherTrust Manager to connect to the CRL server location. If the CRL check fails, the connection remains in a soft fail state, and is allowed to continue. Errors are added to the host logs.

If you do not specify a transport protocol, the value `udp` is applied. With `udp`, log messages are limited to a size of 1024 bytes. After this size, the log message is truncated.

**Usage:**

```
kscfg syslog forwarder add [flags]
```

**Flags:**

```
--client-cert string Syslog client certificate.
--client-cert-file string Syslog client certificate file Alternative way to p
rovide client certificate.
--client-key string Syslog client key.
--client-key-file string Syslog client key file. Alternative way to provide
client key.
-h, --help Help for add
-p, --port int Syslog remote server port. Default value for 'u
dp', 'tcp' and 'tls' are 514, 601 and 6514 respectively.
-q, --queue int Syslog size of queue for messages for forwardin
g to remote server, default 10000.
-r, --retry int Syslog send retry count for forwarding to remot
e server, default 120 times.
-s, --server string Syslog remote server hostname or IP address. Re
quired.
-t, --transport string Syslog remote server transport protocol. Suppor
ted transport protocols are 'udp', 'tcp' and 'tls'.
--trusted-ca string Syslog remote server trusted CA certificate. Requir
ed for tls transport.
--trusted-ca-file string Syslog remote server trusted CA certificate file. A
lternative way to provide CA certificate.
```

**Example:**

```
kscfg syslog forwarder add --server 1.1.1.1 --client-cert-file client-cert.pem
--client-key-file client-key --trusted-ca-file ca-file.pem --transport tls
```

**Response:**

```
{
 "id": "c7619208-860e-4f78-ab34-859e496090ac",
 "server-host": "1.1.1.1",
 "port": 6514,
 "transport": "tls",
 "retry-count": 120,
 "queue-size": 10000,
 "trusted-ca": "-----BEGIN CERTIFICATE-----\nMIIIEwTCCAymgAwIBAgIIYmLF5C
G+O/4wDQYJKoZIhvcNAQELBQAwbDEgMB4GA1UE\nAxAxMXcnN5c2xvZy50aGFsZXNncm91cC5jb20xDDAKB
```

```

gNVBAsTA0NQTDEPMA0GA1UE\nChMGVGVhbnGVzMQ8wDQYDVQQHEwZBdXN0aW4xCzAJBgNVBAGTA1RYMQsw
CQYDVQQG\nEwJVUzAeFw0yMjA0MjIxNTEyMzdaFw0zMjA0MjAxNTEyNDVaMGwxIDAeBgNVBAMT\nF3Jze
XNsb2cudGhhbGVzZ3JvdXAuY29tMQwwCgYDVQQLEwNDUEwxDzANBgNVBAoT\nB1RoYWxlczEPMA0GA1UE
BxMGQXVzdGluMQswCQYDVQQIEwJUWDELMAkGA1UEBhMC\nVVMwggGiMA0GCSqGSIb3DQEBAQUAA4IBjwA
wggGKAoIBgQDEGGS3IpfFcGk7E75\nnGD8GbWh3y8QrRWDSkGbjtYA/UDFESgd2cYK8tAimXYxYaCqYQ8
Pk3n4YCzeYdtWr\nnmMuDvlrOZzNChkMtnuBfDq2xxaLV1sw7ideSbhRs7b7wST0s2ZaaTZ1BUm98kLa4\
nfomhA21XfecKwIdun3mVYz8tv2wvGnz5LUA/VwQelMINJimnRFqjSlIdDnss4vZb\nmXJJIUBcjGeCwHB
KMx6iO+W8t4tVP2LHNEHal6+PlbYYP47SA4AaZKcCrajpDMJDh\nnDAes0rFIhl4mr2s/F+OFOUWEPKWTz
E9hgZJoAYu+fjINR6nMimlrppn056kLG3o\nLjWWR1CCwStHVL00RaQSShlGFwKEDym86sp4lb4Wq7Yg
AE2BM5F0QhqpSAsnp8sn\nnKRYmgOuuak/YZXSSN6DBLdDAv1U7nnh9l4QVvmV+arzlRKarpSDGusThQfH
xmAge\nnYLT2e6ImAkaT3qidweVFvVroAzlXVTcd0YWoqkh0389MsxECaWEAAAnMGUwDwYD\nnVR0TAQH/
BAUwAwEB/zAiBgNVHREEGzAZghdyc3lzbG9nLnRoYXxlcz2dyb3VwLmNv\nbTAPBgNVHQ8BAf8EBQMDBwY
AMB0GA1UdDgQWBbTMv+bTbkr0nBUo07o0fazL6X4h\nnjzANBgkqhkiG9w0BAQsFAAOCAQEAIjRlWUC9eT
KkCR9k7mIwE6sjLkQqQo9CEEqX\nnJhT5FChvxK/TlLb427a3B1glpo0DXXcOt/1A7lzz69vg3FO1DNhP8
ggIuc16qV+c\nnG5eDIzV1XLPOIeyXmImg++/jojtofMJECiYyU0IGQ+3+2rKL3+4F8+yZQBROP0b\nao
9BD/IKH913NSKodkrivn5LSZqdOYi+yiFYUTPe1XA1OnI89E2xH4ZaX0g5Sxxw\nnqWfU3LXrJUD9HQp6I
7hvSHPtByPEWnj1WEVpdlXNuTRZl9Qewp+F2/4xiA0idftU\nnySQLgX9teCgOwn4/TTxgy6v/kWhWJncM
us0T71hUatgATQhqvzk/RH8ucNoKIDKl\nnYezuXywr6To5/9SyH4/z4etCMp+PJH8DQmiYQJB6xRvrNg
S6dCbcYL4pWXuXCKv\nnivzyv/K/G+7PHhgNLLxUZcKzpdxlaSTmtkxhgqWZ7nl/xX+ocrZLjohKHBfBFW
LR\nnSN09lRQcs7RbEpX/HDlIoOzok4KO\nn-----END CERTIFICATE-----\n",
 "client-cert": "-----BEGIN CERTIFICATE-----\nnMIEbTCCAtWgAwIBAgIIYmL8wAp4
ZjgwDQYJKoZIhvcNAQELBQAwbDEgMB4GA1UE\nnAxMXcnN5c2xvZy50aGFsZXNncm9lcC5jb20xDDAKBgN
VBAsTA0NQTDEPMA0GA1UE\nnChMGVGVhbnGVzMQ8wDQYDVQQHEwZBdXN0aW4xCzAJBgNVBAGTA1RYMQswCQ
YDVQQG\nEwJVUzAeFw0yMjA0MjIxOTA2NDFAeFw0yNTAxMTYxOTA2NDVaMGsxHzAdBgNVBAMT\nnFmNsaWV
udC50aGFsZXNncm9lcC5jb20xDDAKBgNVBAsTA0NQTDEPMA0GA1UEChMG\nnVGVhbnGVzMQ8wDQYDVQQHEw
ZBdXN0aW4xCzAJBgNVBAGTA1RYMQswCQYDVQQGEwJV\nnUzCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCA
QoCggEBAKfb1Oz+tul2Grxz/J2x\nn1MBIOjiHJj+d74Bi0VfTZ/xHMCw5pf6xI3AXVw+hukh/qHM0qAZ1
yyOm8PZ155oP\nnTuN62c1ZwkV2c0XvHrjpbK9iIF4zP0y+X23B3eQt7XE3zIiaPmF5+CAMWgPXM1ZR\nnH
yE5qhT7wpGGtNsRVWYezp0XTbPwRdFZe8IeMXIrfiLlONJCLX9ueqR1Ec73QK/e\nn4SVDJHvNwuzXo1BF
RnVh+mLjQFmA+GVsYw6lcbn2tT5T+1lrpPd7ZhZ7AI2XX6T9\nnua3oZXwHNJ9m8Zh+H+qOj4IR/zKZjKs
zHn8wVBjLSZZ8ieXoVpzp+VHbMG1wVqJo\nnBLECAWEAAaOBkzCBkDAMBgNVHRMBAf8EAjAAMB0GA1UdJQ
QWMBQGCCsGAQUFBwMC\nnBggrBgEFBQcDATAhBgNVHREEGjAYghZjbG1lbnQudGhhbGVzZ3JvdXAuY29tM
B0G\nnA1UdDgQWBBS/vJGs7udewv1Lz2/G8T4hbJQn5TAfBgNVHSMEGDAWgBTmv+bTbkr0\nnnBUo07o0fa
zL6X4hjzANBgkqhkiG9w0BAQsFAAOCAQEAnPvw1qon5fusz+HJpXf8\nnD0f4XzEcQf3MWG11KyoTv43
w+4cFH+32pi67fTb0Yu3F8AwbzM0G7WilIuIbfaDU\nnh+Sy/XuczAH2mJ2bhffRqaUD9/53WuMcuqLWj
1TIQaUuf39n06e9AGHTPLvT6ce\nnJ0IAi4MR3Y00a+gJ6MeN/GFhbr3eHPveDSMvRkKaPemxO4cpz05Se
TDtRobioPSq\nnyBUtAYwr4g76mtHZGVvYP2+XgMC6vIuxMOZGfWuS78YT2tJ6Ubdsch5m86ZhkvTD\nnGV
YHKAHrMnIstzEI0r4+BodKm+zMnRsW0pHQMES9brB457GNXU8t136K+v11n98F\nnxX8jn8g//xX7ughI8
mhtrmuHh1R9LIDUqW6fjeSym0dU24c1\nn3/gfgrNgVAXCY+\nn1Hh7oIYU/fvTzH7GFFOzBdbTmefKLJQ4
4M0BBZL9w9TSVzgrPPNDuihw+6L5X0cZ\nnj9UYX/4b1RLVvk9yC+NfgBbiid6NU49vF6kNnjrvrgDE5L\
n-----END CERTIFICATE-----\n"
}

```

## To delete a remote syslog server

### Usage:

```
kscfg syslog forwarder delete [flags]
```

### Flags:

```

-h, --help Help for delete
-i, --id string Syslog forwarder ID. Required.

```

### Example:

---

```
kscfg syslog forwarder delete --id 3ac704de-0701-401b-a742-d6e262673505
```

**Response:**

There is no response for successful execution of this command.

## View a syslog forward configuration

### To view a remote syslog server

**Usage:**

```
kscfg syslog forwarder get [flags]
```

**Flags:**

```
-h, --help Help for get
-i, --id string Syslog forwarder ID. Required.
```

**Example:**

```
kscfg syslog forwarder get --id 3ac704de-0701-401b-a742-d6e262673505
```

**Response:**

```
{
 "id": "3ac704de-0701-401b-a742-d6e262673505",
 "server-host": "1.1.1.1",
 "port": 514,
 "transport": "tcp",
 "retry-count": 120,
 "queue-size": 10000
}
```

### To view all syslog forward configurations

**Usage:**

```
kscfg syslog forwarder list [flags]
```

**Flags:**

```
-h, --help Help for list
```

**Example:**

```
kscfg syslog forwarder list
```

**Response:**

```
{
```

```

 "skip": 0,
 "limit": 10,
 "total": 1,
 "resources": [
 {
 "id": "3ac704de-0701-401b-a742-d6e262673505",
 "server-host": "1.1.1.1",
 "port": 514,
 "transport": "tcp",
 "retry-count": 120,
 "queue-size": 10000
 }
]
 }
}

```

## To update a remote syslog server

### Usage:

```
kscfg syslog forwarder modify [flags]
```

### Flags:

```

--client-cert string Syslog client certificate.
--client-cert-file string Syslog client certificate file Alternative way to provide client certificate.
--client-key string Syslog client key.
--client-key-file string Syslog client key file. Alternative way to provide client key.
-h, --help Help for add
-i, --id string Syslog forwarder ID. Required.
-p, --port int Syslog remote server port. Default value for 'udp', 'tcp' and 'tls' are 514, 601 and 6514 respectively.
-q, --queue int Syslog size of queue for messages for forwarding to remote server, default 10000.
-r, --retry int Syslog send retry count for forwarding to remote server, default 120 times.
-s, --server string Syslog remote server hostname or IP address. Required.
-t, --transport string Syslog remote server transport protocol. Supported transport protocols are 'udp', 'tcp' and 'tls'.
--trusted-ca string Syslog remote server trusted CA certificate. Required for tls transport.
--trusted-ca-file string Syslog remote server trusted CA certificate file. Alternative way to provide CA certificate.

```

### Example:

```
kscfg syslog forwarder modify --id 3ac704de-0701-401b-a742-d6e262673505 --transport tcp
```

### Response:

```

{
 "id": "3ac704de-0701-401b-a742-d6e262673505",

```

```

"server-host": "1.1.1.1",
"port": 514,
"transport": "tcp",
"retry-count": 120,
"queue-size": 10000
}

```

## Entropy Source

The `kscfg system entropy-source` command can be used to configure entropy source.

This command sets entropy source to be used by CipherTrust Manager for random number generation. Entropy source can be one of AUTO, CPU\_JITTER, RDSEED, RDRAND, DEV\_URANDOM, or RNGD\_DEV\_RANDOM. Default is AUTO, where CipherTrust Manager tries to use the best entropy source available on the system. If CipherTrust Manager is configured to use HSM as root of trust, then AUTO defaults to use HSM as the entropy source. If CipherTrust Manager is not configured to use HSM as root of trust, AUTO uses one of CPU\_JITTER, RDSEED, RDRAND, RNGD\_DEV\_RANDOM, or DEV\_URANDOM, in that order.

The CPU\_JITTER option specifies that entropy be collected using a FIPS-certified CPU Jitter entropy library.

RDSEED and RDRAND are CPU instructions and may not be available on all host CPUs. RNGD\_DEV\_RANDOM as well relies on RDRAND instruction being available on the host CPU. When entropy source is set to RDRAND (or RDSEED), CipherTrust Manager directly reads from RDRAND (or RDSEED) to seed the DRBG. When RNGD\_DEV\_RANDOM is set as entropy source, 'rngd' daemon reads from RDRAND and mixes it into the entropy pool in /dev/random to seed the DRBG.

### Caution

If the configured entropy source is unavailable on the system, all the CipherTrust Manager services are unavailable. So, if the entropy source is not set to AUTO, make sure that the entropy source is available on the host. For example, if entropy source is set to RDSEED and RDSEED instruction is not available in the host CPU, you cannot access any CipherTrust Manager services.

The change won't take effect until the CipherTrust Manager appliance is rebooted or CipherTrust Manager services are restarted. Run `"sudo systemctl restart keysecure"` to restart CipherTrust Manager services.

### Usage:

```
kscfg system entropy-source [flags]
```

### Flags:

```

-h, --help Help for entropy-source
-s, --source Entropy source. Can be one of AUTO(default), RDSEED, RDRAND, RNGD_D
EV_RANDOM, or DEV_URANDOM.

```

### Example:

```
kscfg system entropy-source -s RDSEED
```

### Response:

There is no response for successful execution of this command.

## Local Hostname Configuration

You can list or set the local hostname which local applications use to connect to CipherTrust Manager. The default hostname is `ciphertrust`.

### Note

The local hostname affects [clustering](#). To successfully join a cluster, a CipherTrust Manager's hostname can consist only of lowercase letters, numbers, and hyphens.

### To list the local hostname

#### Usage:

```
kscfg system hostname get
```

#### Flags:

```
-h, --help Help for get
```

#### Example Response:

```
ciphertrust
```

### To set the local hostname

### Note

The change won't take effect until the CipherTrust Manager appliance is rebooted or CipherTrust Manager services are restarted. Run "sudo systemctl restart keysecure" to restart CipherTrust Manager services.

#### Usage:

```
kscfg system hostname set [flags]
```

#### Flags:

```
-h, --help Help for set
-n, --host-name this command can set the hostname, updating /etc/hostname file.
```

#### Example:

```
kscfg system hostname set -n newhostname
```

**Response:**

```
Note: please run "sudo systemctl restart keysecure" to have new hostname effective in CipherTrust Manager
```

## System Reset

Consult [Effects of Reset Methods](#) for details on what data is cleared for the different reset methods.

The `kscfg system reset` command can be used to perform a hard reset of the CipherTrust Manager.

**Warning**

This destructive operation wipes all data on the CipherTrust Manager and should be used with care.

**Note**

This command requires the `host-daemon` system service to be up and running.

Normally, the REST API or the CLI should be used for performing the reset. This method of performing the reset should be used as a last resort. This operation deletes all backup keys and the HSM configuration. It is good practice to do the following before running this command:

1. Create and download a backup of the database.
2. Download all the backup keys. Any backups downloaded from this device will not be useful without the backup keys.

`kscfg reset` commands do not reset the HSM and the root of trust keys. This allows you to restore a previous CipherTrust Manager backup taken on the appliance. However, if you performed the factory reset to return the appliance to a fresh security state, and you don't intend to restore a backup, we strongly recommend resetting and re-initializing the HSM to create new root of trust keys. On a k570 device with an embedded PCIe HSM, you reset the HSM using the `lunaCM` command “`hsm factoryReset`” and then re-initialize following the [same HSM configuration process](#) as used during first deployment. For external HSMs used as the root of trust, consult their product documentation to delete the root of trust keys and perform any applicable reset operations.

## Usage

```
kscfg system reset [flags]
```

Flags:

```
-f, --force When this flag is set, any errors encountered during reset are ignored, and the reset procedure
```



```

 continues to the end. This flag must be used with care as it could
place the system in an unuseable state. It
 should be used when all else fails.

-h, --help help for reset

-y, --yes When this flag is set, all user prompts during the reset process ar
e skipped. A default value
 of 'yes' is used as the automatic response to all prompts.

```

## Examples

```
kscfg system reset [-f] [-y]
```

### Response:

```

This will perform a full reset of the CipherTrust Manager services.
WARNING - This is a destructive operation and will wipe all data in the Cipe
rTrust Manager.
It will delete all backupkeys and the HSM configuration.
Normally, the REST API or the CLI should be used for performing the reset.
THIS METHOD OF PERFORMING THE RESET SHOULD BE USED AS A LAST RESORT.
It is good practice to perform the following steps prior to running this comm
and:
 1. Create and download a backup of the database.
 2. Download all the backupkeys; any backups downloaded from this device wi
ll not be useful without the backupkeys.
Do you want to continue? [y/N] y
This will take some time, please wait
Device reset has started. It will take a few minutes to complete.

```

## System Factory Reset

Consult [Effects of Reset Methods](#) for details on what data is cleared for the different reset methods.

The `kscfg system factory-reset` can be used on k470 and k570 appliance models only, to revert the system to its factory defaults.

### Warning

This destructive operation wipes all data on the CipherTrust Manager, including keys, backups, backup keys, system configuration, and logs. It automatically reboots the appliance twice, before booting to the factory firmware version. The appliance's factory version may be below the currently running version. Several [system upgrades](#) may be required to return to the currently running version. Do not manually power-off or reset the appliance while the factory-reset is in progress. This command must be used with care.

**Note**

This command expects the host-daemon system service to be up and running. However, if the host-daemon is not running or not in a good state, the factory-reset can be invoked from command line as ksadmin user by executing "sudo /opt/keysecure/ks\_reset\_to\_factory.sh".

kscfg reset commands do not reset the HSM and the root of trust keys. This allows you to restore a previous CipherTrust Manager backup taken on the appliance. However, if you performed the factory reset to return the appliance to a fresh security state, and you don't intend to restore a backup, we strongly recommend resetting and re-initializing the HSM to create new root of trust keys. On a k570 device with an embedded PCIe HSM, you reset the HSM using the lunaCM command "hsm factoryReset" and then re-initialize following the [same HSM configuration process](#) as used during first deployment. For external HSMs used as the root of trust, consult their product documentation to delete the root of trust keys and perform any applicable reset operations.

**Usage**

```
kscfg system factory-reset [flags]
```

Flags:

```
-h, --help help for factory-reset

-y, --yes When this flag is set, all user prompts during the reset process are skipped. A default value of 'yes' is used as the automatic response to all prompts.
```

**Examples**

```
kscfg system factory-reset [-y]
```

**Response:**

```
WARNING: This operation will revert the system to its factory defaults !!!

(1) This is a destructive operation that erases all CipherTrust Manager data including but not limited to keys, backups, backup keys, and system logs.
(2) Ensure that you have access to serial console to configure the network interface.
(3) Ensure that you have a valid CipherTrust Manager backup of all the data and backup key.
(4) If embedded HSM is available, it will not be reset as part of this operation.
 Re-initialization of embedded HSM is highly recommended after this operation to configure it as the root of trust.
(5) If remote PED was used, it must be re-connected after completion.
(6) This operation may take up to 15 minutes. Make sure you have power backup in place.
(7) Access to the system will be unavailable. DO NOT restart the system during this time.
(8) This operation includes multiple system reboot.
(9) This operation CANNOT be undone.
```

```
Do you want to continue?
```

```
[y/N]
```

## Adding Connector Licenses After System Reset

System reset changes the **Connector Lock Code** for the CipherTrust Manager. After system reset, any license files based on that earlier **Connector Lock Code** cannot be added. You can restore the earlier **Connector Lock Code** from a backup, or by adding the reset CipherTrust Manager node into a cluster with the earlier **Connector Lock Code**. Then, these license files can be added. As well, backup restore and cluster replication include previously installed licenses.

## Resetting a CipherTrust Manager

In some scenarios, you might wish to reset a CipherTrust Manager, deleting its data and restoring to a fresh state. There are a few ways to reset, depending on your access permissions to the CipherTrust Manager.

- If you have completed full initialization of the appliance and can log in to the CLI or the REST API as the root `Admin`, those interfaces are preferred.
- If you can access the CipherTrust Manager through SSH or serial console, and login as the `ksadmin` user, you can use the `kscfg` system configuration utility to perform a `system reset` or `system factory reset`.
- If you have serial access to a physical CipherTrust Manager but cannot login as `ksadmin`, you can perform a `zero knowledge factory reset`.
- For CipherTrust Manager k160 physical appliances only, there are two zeroization buttons on the front of the unit which can be used to reset the appliance.

For k160 models, we recommend use of the zeroization buttons to rapidly restrict all access to sensitive edge-deployed environments through zeroization of keys stored by CipherTrust Manager.

We recommend `REST API/CLI reset` or `kscfg system reset` when the keysecure service cannot start application services due to corrupted data, or when the keysecure service cannot bootstrap the application.

We recommend `kscfg system factory reset` or `Zero knowledge factory reset` when CipherTrust Manager's host operating system or kernel is in a bad state.

## Effects of reset methods

All reset methods reset CipherTrust Manager application configurations which are performed through the GUI, CLI, or REST API related to routine administration and key management. In general, the `REST API/CLI reset` and `kscfg system reset` methods preserve configurations at the operating system level. In general, `kscfg system factory reset`, `zero knowledge factory reset`, and k160 zeroization buttons delete these OS-level configurations.

The following table summarizes differences between the five methods:

| Method                       | Operating System                                                                                                                                                                                                                                                                                                                             | Network                               | Software Version                                                                 | Required Permissions                                                                                                                                    | Supported models                    |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| REST API/CLI reset           | Preserves OS level configurations such as SSH keys and disk encryption.                                                                                                                                                                                                                                                                      | Preserves network configuration.      | Preserves CipherTrust Manager software version.                                  | An Application Administrator in the admins group, such as the default admin account.                                                                    | All.                                |
| kscfg system reset           | Preserves OS level configurations such as SSH keys and disk encryption.                                                                                                                                                                                                                                                                      | Preserves network configuration.      | Preserves CipherTrust Manager software version.                                  | The ksadmin user through SSH access.                                                                                                                    | All.                                |
| kscfg system factory reset   | Deletes OS level configurations such as SSH keys and disk encryption.                                                                                                                                                                                                                                                                        | Deletes network configuration.        | Downgrades CipherTrust Manager to the software version installed at the factory. | The ksadmin user through SSH access.                                                                                                                    | Physical k570 and k470 models only. |
| Zero knowledge factory reset | Deletes OS level configurations such as SSH keys and disk encryption.                                                                                                                                                                                                                                                                        | Deletes network configuration.        | Downgrades CipherTrust Manager to the software version installed at the factory. | Requires serial console access. Thales customer support verifies identity to ensure only authorized persons can perform a zero knowledge factory reset. | Physical k570 and k470 models only. |
| k160 Zeroization Buttons     | Wipes the token of all objects (if token HSM is present); Clears the SSH key; Removes the OS-level logs; Removes/resets the "ksadmin" password (must be reset at next login); Removes upgrade files and upgrade logs; performs a system reset (similar to kscfg system reset command); Powers off the device after approximately one minute. | Deletes custom network configuration. | Preserves CipherTrust Manager software version.                                  | The ksadmin user through SSH access.                                                                                                                    | Physical k160 models only.          |

| Method | Operating System                                  | Network | Software Version | Required Permissions | Supported models |
|--------|---------------------------------------------------|---------|------------------|----------------------|------------------|
|        | (LED Light turns off after success of operation). |         |                  |                      |                  |

## HSM state and backup restore

Most of these methods delete HSM root of trust connection parameters, but do not reset the HSM or delete the root of trust keys. The k160 zeroization buttons wipe all objects from any attached token HSM. This has implications for security and for the ability to restore backups.

### k160 zeroization buttons

You can restore most backups after this zeroization. However, backups which are tied to HSM cannot be restored.

### All other methods

With the HSM root of trust keys preserved, you can restore a previous CipherTrust Manager backup taken on the appliance. However, if you performed the reset to return the appliance to a fresh security state, and you don't intend to restore a backup, we strongly recommend resetting and re-initializing the HSM to create new root of trust keys. On a k570 device with an embedded PCIe HSM, you reset the HSM using the lunaCM command "hsm factoryReset" and then re-initialize following the [same HSM configuration process](#) as used during first deployment. For external HSMs used as the root of trust, consult their product documentation to delete the root of trust keys and perform any applicable reset operations.

## Reset through REST API or CLI

Consult [Effects of Reset Methods](#) for details on what data is cleared for the different reset methods.

### Warning

This destructive operation wipes all data on the CipherTrust Manager and should be used with care.

In the REST API, the root `admin` can use POST with `/v1/system/services/reset` to wipe application data in CipherTrust Manager. You can optionally include `"delay": integer` in the request body to set a delay in seconds. The default delay is 5 seconds.

In the ksctl CLI, the root `admin` can run `ksctl services reset`. You can optionally include the `--delay` flag to set a delay in seconds. The default delay is 5 seconds.

## System Reset

Consult [Effects of Reset Methods](#) for details on what data is cleared for the different reset methods.

The `kscfg system reset` command can be used to perform a hard reset of the CipherTrust Manager.

**Warning**

This destructive operation wipes all data on the CipherTrust Manager and should be used with care.

**Note**

This command requires the host-daemon system service to be up and running.

Normally, the REST API or the CLI should be used for performing the reset. This method of performing the reset should be used as a last resort. This operation deletes all backup keys and the HSM configuration. It is good practice to do the following before running this command:

1. Create and download a backup of the database.
2. Download all the backup keys. Any backups downloaded from this device will not be useful without the backup keys.

kscfg reset commands do not reset the HSM and the root of trust keys. This allows you to restore a previous CipherTrust Manager backup taken on the appliance. However, if you performed the factory reset to return the appliance to a fresh security state, and you don't intend to restore a backup, we strongly recommend resetting and re-initializing the HSM to create new root of trust keys. On a k570 device with an embedded PCIe HSM, you reset the HSM using the lunaCM command “hsm factoryReset” and then re-initialize following the [same HSM configuration process](#) as used during first deployment. For external HSMs used as the root of trust, consult their product documentation to delete the root of trust keys and perform any applicable reset operations.

## Usage

```
kscfg system reset [flags]
```

Flags:

```
-f, --force When this flag is set, any errors encountered during reset are ignored, and the reset procedure continues to the end. This flag must be used with care as it could place the system in an unuseable state. It should be used when all else fails.

-h, --help help for reset

-y, --yes When this flag is set, all user prompts during the reset process are skipped. A default value of 'yes' is used as the automatic response to all prompts.
```

## Examples

```
kscfg system reset [-f] [-y]
```

**Response:**

```

This will perform a full reset of the CipherTrust Manager services.
WARNING - This is a destructive operation and will wipe all data in the CipherTrust Manager.
It will delete all backupkeys and the HSM configuration.
Normally, the REST API or the CLI should be used for performing the reset.
THIS METHOD OF PERFORMING THE RESET SHOULD BE USED AS A LAST RESORT.
It is good practice to perform the following steps prior to running this command:
 1. Create and download a backup of the database.
 2. Download all the backupkeys; any backups downloaded from this device will not be useful without the backupkeys.
Do you want to continue? [y/N] y
This will take some time, please wait
Device reset has started. It will take a few minutes to complete.

```

## System Factory Reset

Consult [Effects of Reset Methods](#) for details on what data is cleared for the different reset methods.

The `kscfg system factory-reset` can be used on k470 and k570 appliance models only, to revert the system to its factory defaults.

### Warning

This destructive operation wipes all data on the CipherTrust Manager, including keys, backups, backup keys, system configuration, and logs. It automatically reboots the appliance twice, before booting to the factory firmware version. The appliance's factory version may be below the currently running version. Several [system upgrades](#) may be required to return to the currently running version. Do not manually power-off or reset the appliance while the factory-reset is in progress. This command must be used with care.

### Note

This command expects the host-daemon system service to be up and running. However, if the host-daemon is not running or not in a good state, the factory-reset can be invoked from command line as ksadmin user by executing "`sudo /opt/keysecure/ks_reset_to_factory.sh`".

`kscfg reset` commands do not reset the HSM and the root of trust keys. This allows you to restore a previous CipherTrust Manager backup taken on the appliance. However, if you performed the factory reset to return the appliance to a fresh security state, and you don't intend to restore a backup, we strongly recommend resetting and re-initializing the HSM to create new root of trust keys. On a k570 device with an embedded PCIe HSM, you reset the HSM using the lunaCM command "`hsm factoryReset`" and then re-initialize following the [same HSM configuration process](#) as used during first deployment. For external HSMs used as the root of trust, consult their product documentation to delete the root of trust keys and perform any applicable reset operations.

## Usage

```
kscfg system factory-reset [flags]
```

Flags:

```
-h, --help help for factory-reset

-y, --yes When this flag is set, all user prompts during the reset process are skipped. A default value of 'yes' is used as the automatic response to all prompts.
```

## Examples

```
kscfg system factory-reset [-y]
```

Response:

```
WARNING: This operation will revert the system to its factory defaults !!!

(1) This is a destructive operation that erases all CipherTrust Manager data including but not limited to keys, backups, backup keys, and system logs.
(2) Ensure that you have access to serial console to configure the network interface.
(3) Ensure that you have a valid CipherTrust Manager backup of all the data and backup key.
(4) If embedded HSM is available, it will not be reset as part of this operation.
 Re-initialization of embedded HSM is highly recommended after this operation to configure it as the root of trust.
(5) If remote PED was used, it must be re-connected after completion.
(6) This operation may take up to 15 minutes. Make sure you have power backup in place.
(7) Access to the system will be unavailable. DO NOT restart the system during this time.
(8) This operation includes multiple system reboot.
(9) This operation CANNOT be undone.

Do you want to continue?

[y/N]
```

## Adding Connector Licenses After System Reset

System reset changes the **Connector Lock Code** for the CipherTrust Manager. After system reset, any license files based on that earlier **Connector Lock Code** cannot be added. You can restore the earlier **Connector Lock Code** from a backup, or by adding the reset CipherTrust Manager node into a cluster with the earlier **Connector Lock Code**. Then, these license files can be added. As well, backup restore and cluster replication include previously installed licenses.

## Zero knowledge factory reset

Consult [Effects of Reset Methods](#) for details on what data is cleared for the different reset methods.



This way of resetting an appliance requires no authentication. Zero knowledge factory reset is available for k570 and k470 physical appliances only. To protect from misuse, this feature requires serial access to the appliance, and contact with Thales customer support.

### Warning

This destructive operation wipes all data on the CipherTrust Manager, including keys, backups, backup keys, system configuration, and logs. It automatically reboots the appliance twice, before booting to the factory firmware version. The appliance's factory version may be below the currently running version. Several [system upgrades](#) may be required to return to the currently running version. Do not manually power-off or reset the appliance while the factory-reset is in progress. This command must be used with care.

1. Open a [serial connection](#) to the appliance.
2. At the `ciphertrust login:` prompt enter `factoryreset`

You are presented with the following options:

```
Options:
1. Initiate factory reset by generating a challenge from this system
2. Input response and perform factory reset
```

3. Enter `1` to generate a challenge request.

```
Your choice: 1
Copy the following request to CipherTrust Manager support:
<challenge_request_string>
```

4. Copy the challenge string and send it to [Thales customer support](#).
5. Press `ENTER` in the serial session to return to the previous options.
6. Once you have received a response text from customer support, enter `2` to input the response.

```
Your choice: 2
Paste response text from support (end with an empty line):
<support_response_string>
```

The following warning is displayed:

```
WARNING: This operation will revert the system to its factory defaults !!!
(1) This is a destructive operation that erases all CipherTrust Manager da
ta including but not limited to keys, backups, backup keys, and system logs.
```

(2) Ensure that you have a valid CipherTrust Manager backup of all the data and backup key.

(3) If embedded HSM is available, it will not be reset as part of this operation.

Re-initialization of embedded HSM is highly recommended after this operation to configure it as the root of trust.

(4) If remote PED was used, it must be re-connected after completion.

(5) This operation may take up to 15 minutes. Make sure you have power backup in place.

(6) Access to the system will be unavailable. DO NOT restart the system during this time.

(7) This operation includes multiple system reboot.

(8) This operation CANNOT be undone.

7. Type `proceed` to continue with the reset.

# CipherTrust Manager



## Overview

CipherTrust Manager enables organizations to centrally manage encryption keys for Thales CipherTrust Data Security Platform and third party products. 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, and robust auditing and reporting of all key management and encryption operations.

CipherTrust Manager is the central management point for the [CipherTrust Data Security Platform](#). It provides a unified management console that makes it easy to discover and classify data, and to protect sensitive data wherever it resides using a comprehensive set of CipherTrust Data Protection Connectors from Thales.

CipherTrust Manager is available in both virtual and physical form-factors that integrate with FIPS 140-2 validated Thales Luna Network or Cloud HSM, and third-party Hardware Security Modules (HSMs) for securely storing master keys with highest root of trust. These appliances can be deployed on-premises as well as in private or public cloud infrastructures. This allows customers to address compliance requirements, regulatory mandates and industry best practices for data security.

## Benefits

- Centralized key and policy management for on-premises data stores and cloud infrastructures
- Reduced business risk with unified data discovery, classification and sensitive data protection
- Simplified management with self-service licensing portal and visibility into licenses in use
- Cloud friendly deployment options with support for AWS, Azure, Google Cloud, VMware, Oracle Cloud Infrastructure and more
- Expanded Hardware Security Module (HSM) support for superior key control
- Unparalleled partner ecosystem of integrations with leading enterprise storage, server, database, application and SaaS vendors



Key  
Management



Access  
Policies



Auditing  
Reporting



Flexible  
APIs



CipherTrust Manager

## Key Capabilities

- **Full Key Lifecycle Management and Automated Operations:** CipherTrust Manager simplifies management of encryption keys across their entire lifecycle, including secure key generation, backup/restore, clustering, deactivation, and deletion. It makes automated, policy-driven operations easy to perform, and generates alarms for events of interest.
- **Unified Management Console:** Provides a unified console for discovering and classifying sensitive data integrated with a comprehensive set of CipherTrust Data Protection Connectors to encrypt or tokenize data to reduce business risk and satisfy compliance regulations.
- **Centralized Administration and Access Controls:** Unifies key management operations with role-based access controls. Authenticates and authorizes administrators and key users using existing AD and LDAP credentials. Prevents unauthorized password change and alerts on simultaneous logins by same user.
- **Self-service Licensing:** Streamlines provisioning of connector licenses through a new customer facing licensing portal. The new management console offers better visibility and control of licenses in use.
- **Secrets Management:** Provides the ability to create and manage secrets and opaque objects for usage on the platform.
- **Multi-tenancy Support:** Supports separation of duties with delegated user management within multiple domains.
- **Developer Friendly REST APIs:** Offers new REST interfaces, in addition to Key Management Interoperability Protocol (KMIP) and NAE-XML APIs, allows customers to remotely generate and manage keys.
- **Flexible HA Clustering and Intelligent Key Sharing:** Provides the option of clustering physical and / or virtual appliances together to assure high availability as well as increased encryption transaction throughput.
- **Robust Auditing and Reporting:** Includes tracking of all key state changes, administrator access, and policy changes in multiple log formats (RFC-5424, CEF, LEEF) for easy integration with SIEM tools.
- **Broad Partner Ecosystem:** CipherTrust Manager provides centralized key management for wide variety of storage partners via KMIP and database partners via Transparent Database Encryption (TDE).

## CipherTrust Manager Features

| Features                                | Virtual Appliances                                                                                                                                                |                                                                                                                                                                   | Physical Appliances                                                                                                                                               |                                 |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
|                                         | k170v                                                                                                                                                             | k470v                                                                                                                                                             | k470                                                                                                                                                              | k570                            |
| <b>Administrative Interfaces</b>        | Management Console, REST API, kscfg (system configuration), (kscl (Command Line Interface)                                                                        |                                                                                                                                                                   |                                                                                                                                                                   |                                 |
| <b>Network Management</b>               | SNMP v1, v2c, v3, NTP, Syslog-TCP                                                                                                                                 |                                                                                                                                                                   |                                                                                                                                                                   |                                 |
| <b>API Support</b>                      | REST, NAE-XML, KMIP, PKCS#11, JCE, .NET, MCCAPI, MS CNG                                                                                                           |                                                                                                                                                                   |                                                                                                                                                                   |                                 |
| <b>Secure Authentication</b>            | Local User , AD/LDAP, LDAPS, Certificate based authentication, Supports Open ID Connect (OIDC)                                                                    |                                                                                                                                                                   |                                                                                                                                                                   |                                 |
| <b>System Formats</b>                   | RFC-5424, CEF, LEEF                                                                                                                                               |                                                                                                                                                                   |                                                                                                                                                                   |                                 |
| <b>Supported HSMs for Root of Trust</b> | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM | Luna Network HSM, Luna T-Series Network HSM, Luna Cloud HSM, AWS Cloud HSM, Azure Dedicated HSM, IBM Cloud HSM, IBM Cloud Hyper Protect Crypto Services Cloud HSM | N/A (has built in HSM)          |
| <b>Automated Deployment Support</b>     | Yes (via Cloud-Init)                                                                                                                                              | Yes (via Cloud-Init)                                                                                                                                              | No                                                                                                                                                                | Yes (via Secure Transport Mode) |
| <b>Maximum Number of Keys</b>           | Tested up to 1M Keys (more possible with appropriately sized virtual environments)                                                                                | Tested up to 1M Keys (more possible with appropriately sized virtual environments)                                                                                | 1 Million Keys                                                                                                                                                    | 1 Million Keys                  |
| <b>Maximum Domains (multi-tenancy)</b>  | 100                                                                                                                                                               | 1000                                                                                                                                                              | 1000                                                                                                                                                              | 1000                            |

## Appliance Specifications

| Physical Appliances       | k470                                                                                     | k570                                                                                                |
|---------------------------|------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| Dimensions                | 19" x 21" x 1.725" (482.6mm x 533.4mm x 43.815mm)                                        |                                                                                                     |
| Hard Drive                | 1x 2TB SATA SE (Spinning Disk)                                                           |                                                                                                     |
| CPU                       | Xeon E3- 1275v6 Processor                                                                |                                                                                                     |
| RAM                       | 16GB                                                                                     |                                                                                                     |
| NIC Support               | 4x1GB or 2x10Gb/2x1Gb (NIC Bonding capable)                                              |                                                                                                     |
| Rack Mount                | Standard 1U rack mountable<br>Sliding rails can be optionally purchased                  |                                                                                                     |
| Reliability               | Dual hot swappable power supplies                                                        |                                                                                                     |
| Safety and Compliance     | CSA C-US, FCC, CE, VCCI, C-TICK, KC Mark, BIS                                            |                                                                                                     |
| Mean Time Between Failure | 165,279 hours                                                                            | 153,583 hours                                                                                       |
| FIPS Support              | Integrates with an external FIPS Certified Physical or Cloud HSM as Secure Root of Trust | Embedded PCI-HSM FIPS 140-2 Level 3 certified – password and multi-factor (PED) (Certificate #3205) |

| Virtual Appliances           | k170v                                                                                                                                                                                                                                                                                                                                                               | K470v                                                                                                                                                         |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System Requirements          | <ul style="list-style-type: none"> <li>RAM (GB): 16</li> <li>Hard Disk (GB): 100</li> <li>NICS: 1 or more</li> <li>CPUs: up to 4 CPU max</li> </ul>                                                                                                                                                                                                                 | <ul style="list-style-type: none"> <li>RAM (GB): 16 or more</li> <li>Hard Disk (GB): 200 or more</li> <li>NICS: 2 or more</li> <li>CPUs: 5 or more</li> </ul> |
| Clouds/Hypervisors Supported | <ul style="list-style-type: none"> <li>Public Clouds: AWS Cloud, Microsoft Azure, Google Cloud Enterprise (GCE), Oracle Cloud Infrastructure (OCI), Alibaba Cloud</li> <li>Private Clouds/Hypervisors: VMware vSphere (6.5, 6.7 and 7.0), Microsoft Hyper-V, Nutanix AHV, OpenStack (QCOW2)</li> </ul> <p>* AWS GovCloud, Azure Government Cloud also supported</p> |                                                                                                                                                               |