Cloudera Manager 7.13.2

Ranger KMS

Date published: 2020-11-30
Date modified: 2026-02-27



https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2026. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Ranger KMS overview....4

Configuring Ranger KMS High Availability....4
Configure High Availability for Ranger KMS with DB....4

Rotating Ranger KMS access log files....14

Using the Ranger Key Management Service....15
Accessing the Ranger KMS Web UI....15
List and Create Keys....20
Roll Over an Existing Key....21
Delete a Key....25

Configuration of cryptographic parameters....25
Master key configurations....25
Zone key configurations....26

Securing the Key Management System (KMS)....27
Enabling Kerberos Authentication for the KMS....27
Configuring TLS/SSL for the KMS....28

Migrating Ranger Key Management Server Role Instances to a New Host....28
Migrate the Ranger Admin role instance to a new host....29
Migrate the Ranger KMS DB role instance to a new host....29

Working with an HSM....29
Set up Luna 7 HSM for Ranger KMS....30
Set up Luna 10.5 HSM Client for Ranger KMS....37
Integrating Ranger KMS DB with Google Cloud HSM....43
Integrating Ranger KMS with Google Cloud HSM....45
Integrating Ranger KMS DB with CipherTrust Manager HSM....48
Fresh installation - Configuring Ranger KMS DB to interact with Thales CipherTrust HSM....54
Migrating Ranger KMS DB Master Key To CipherTrust Manager HSM....56
Migrating the Master key from CipherTrust Manger HSM to Ranger KMS DB....58
Integrating Ranger KMS DB with SafeNet Keysecure HSM....58
Migrating the Master Key from Ranger KMS DB to Luna HSM....65
Migrating the Master Key from HSM to Ranger KMS DB....67

Ranger KMS overview

Apache Ranger Key Management Service (KMS) provides a centralized key management service that allows you to create, manage, and store encryption keys used for data encryption and decryption across various Hadoop ecosystem components. By integrating with Apache Ranger, it offers robust security features, including fine-grained authorization, auditing, and policies management for encryption keys.

Key features of Apache Ranger KMS include:

- Centralized key management

  Simplifies the management of encryption keys across your Hadoop ecosystem.
- Fine-grained authorization

  Allows administrators to define detailed access policies for encryption keys, ensuring only authorized users and applications can use them.
- Comprehensive auditing

  Tracks all key-related activities, providing detailed logs and reports for compliance and security audits.
- Seamless integration

  Works seamlessly with Hadoop Distributed File System (HDFS) encryption and other Hadoop ecosystem components, enhancing overall data security.

For information on how to install Ranger KMS, see *Setting Up Data at Rest Encryption for HDFS*.

Related Information

Setting Up Data at Rest Encryption for HDFS

Configuring Ranger KMS High Availability

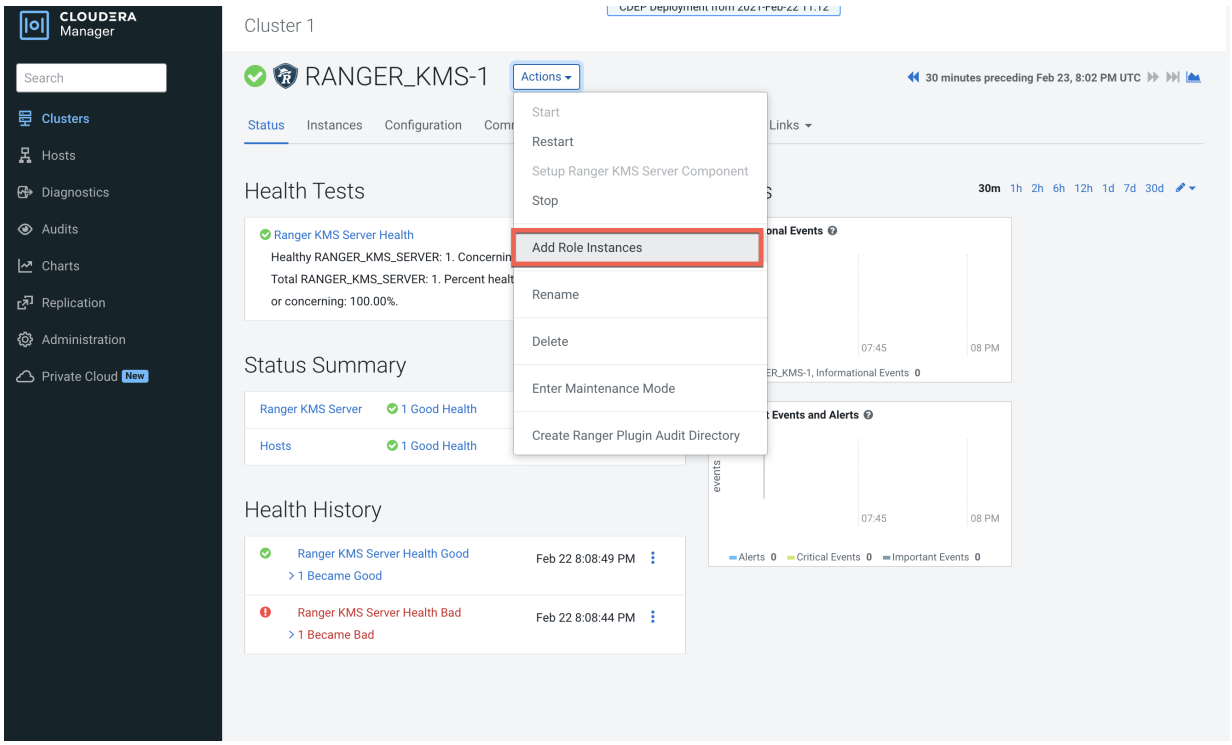
How to configure Ranger KMS high availability (HA) for Ranger KMS.

Configure High Availability for Ranger KMS with DB

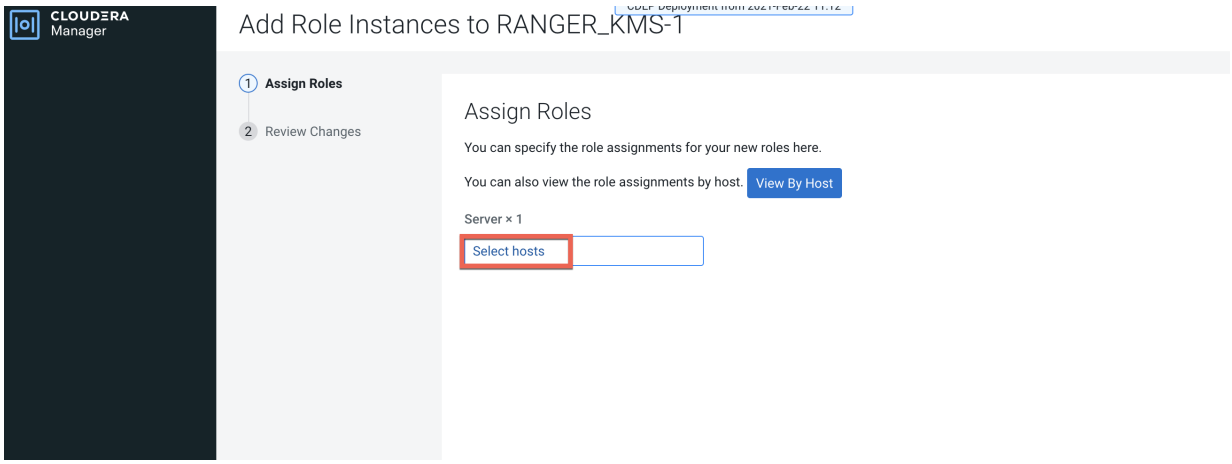
Use the following steps to configure high availability for Ranger KMS with an associated keystore database.

Procedure

1. In Cloudera Manager, select Ranger KMS, then select Actions > Add Role Instances.

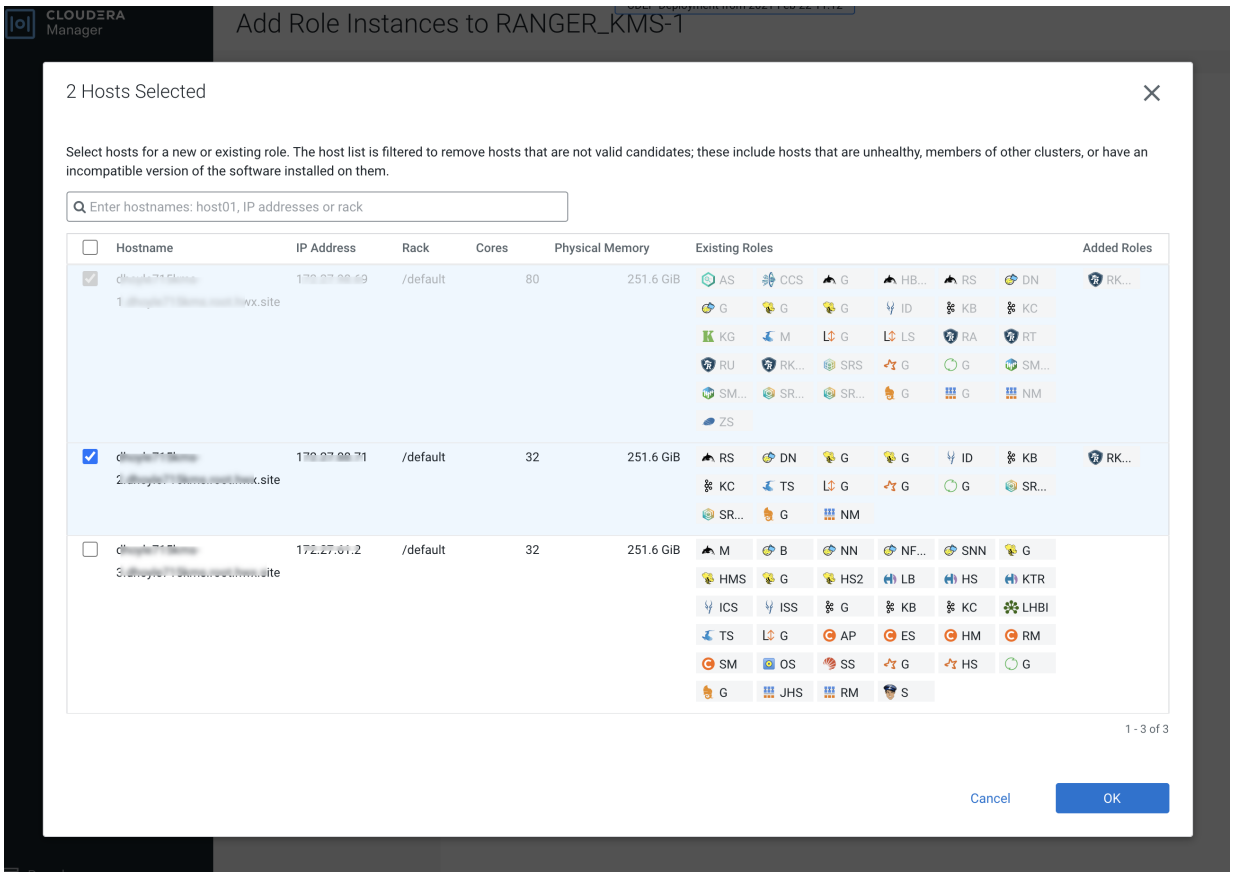


2. On the Assign Roles page, click Select hosts.

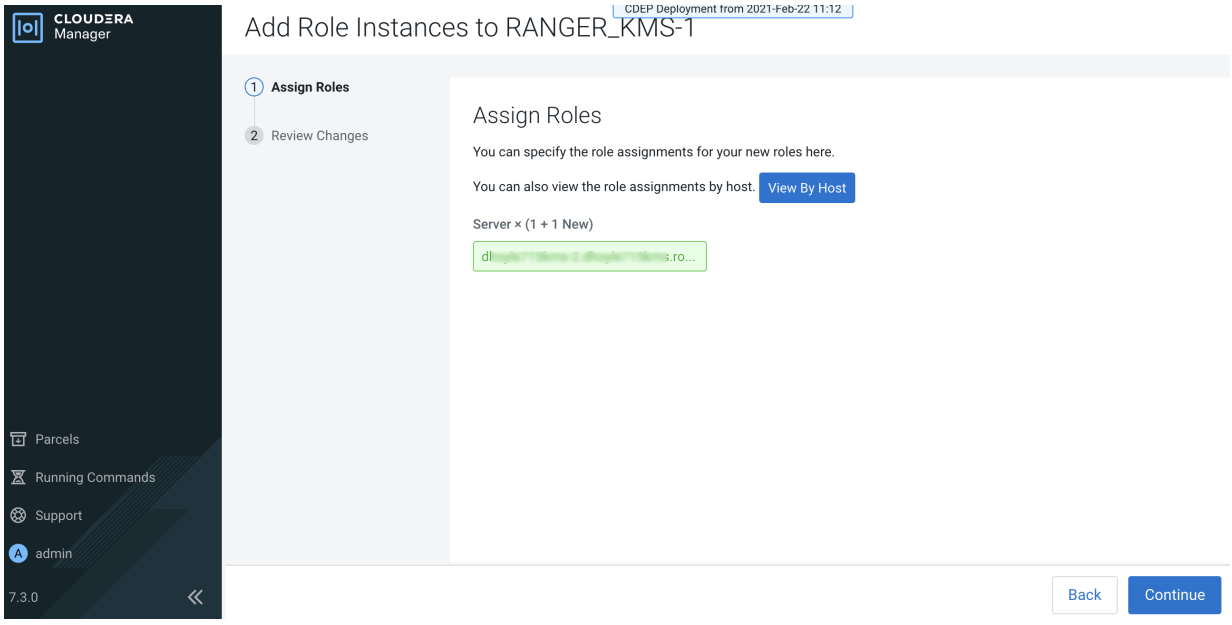


3. On the selected hosts page, select a backup Ranger KMS host. A Ranger KMS (RK) icon appears in the Added Roles column for the selected host. Click OK to continue.

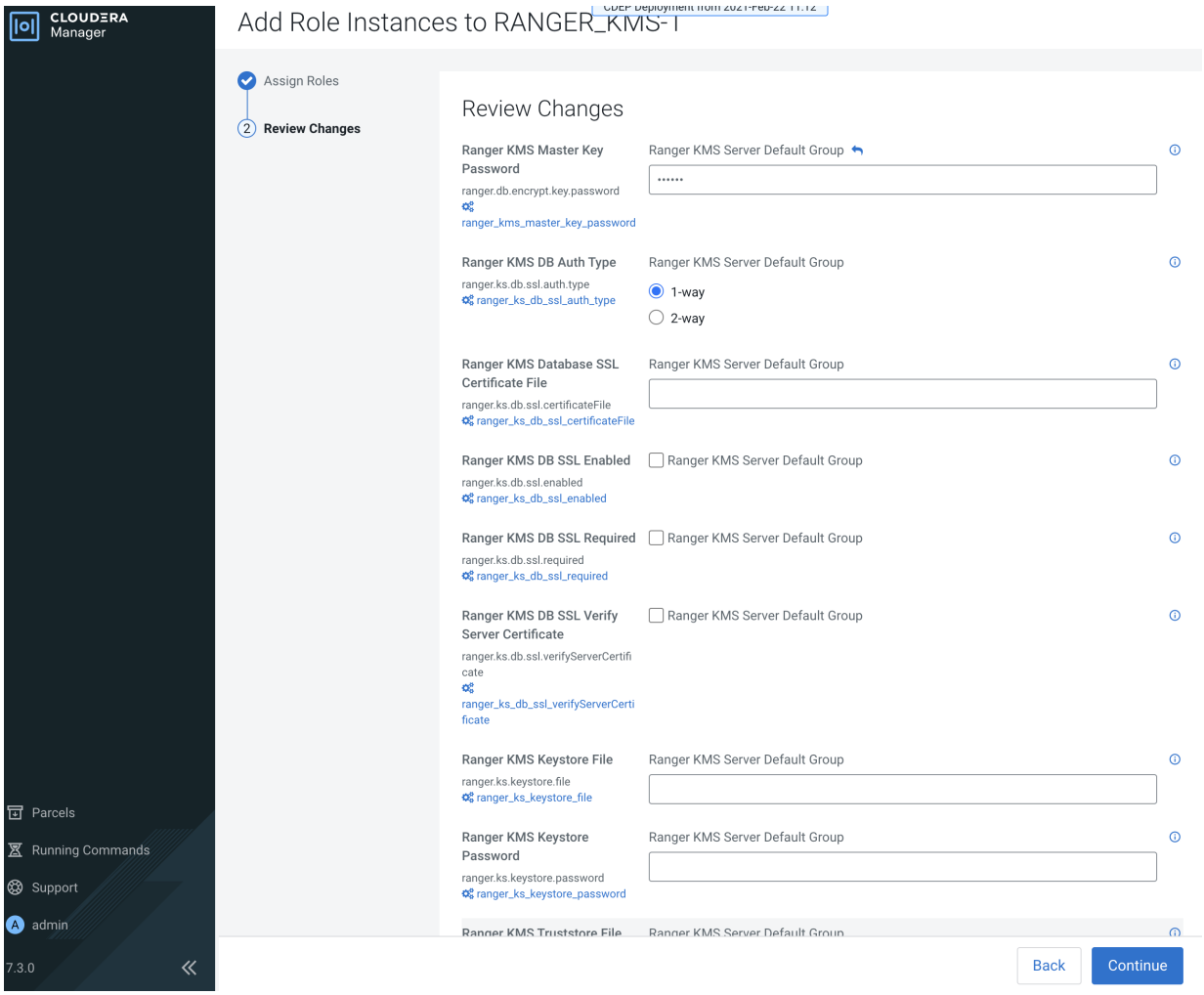
Note: These steps show how to add one additional backup Ranger KMS host, but you can use the same procedure to add multiple Ranger KMS hosts.



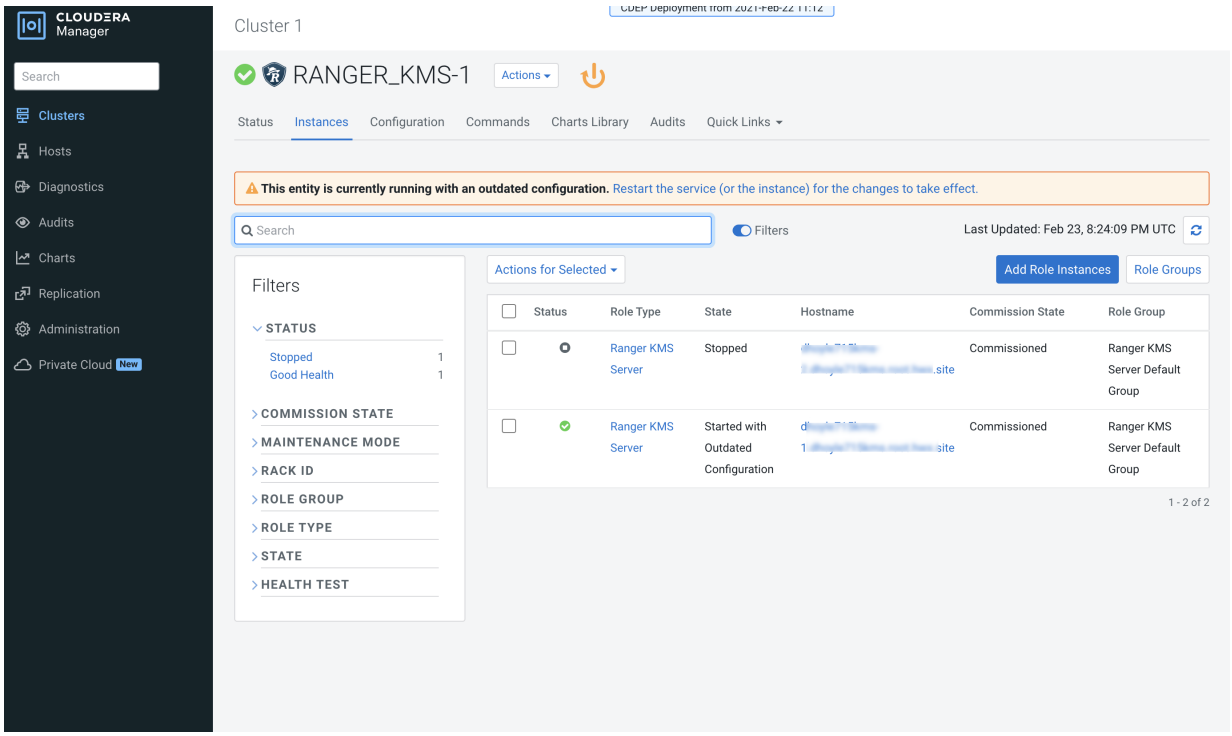
4. The Assign Roles page is redisplayed with the new backup host. Click Continue.



5. Review the settings on the Review Changes page, then click Continue.



6. The new role instance appears on the Ranger KMS page. If the new Ranger KMS instance was not started by the wizard, you can start the service by clicking Actions > Start in the Ranger KMS service.



7. In Cloudera Manager, select the Ranger service, click Ranger Admin Web UI, then log in as the Ranger KMS user (the default credentials are keyadmin/admin123). Click the Edit icon for the cm_kms service, then update the KMS URL property.

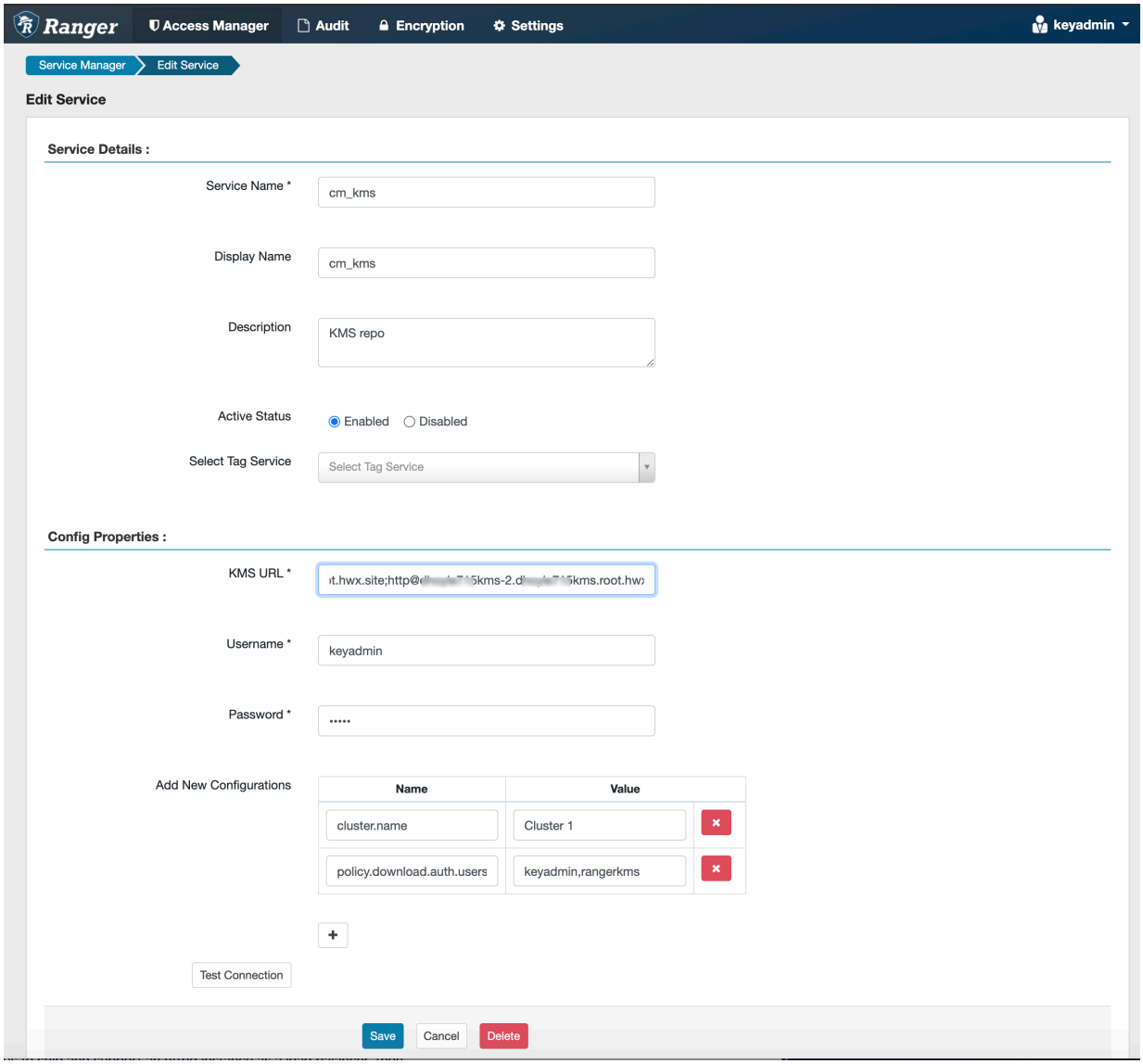
   - Add the new KMS host using the following format:

     kms://http@<kms_host1>;http@<kms_host2>:<kms_port>/kms
   - The default port is 9292. For example:

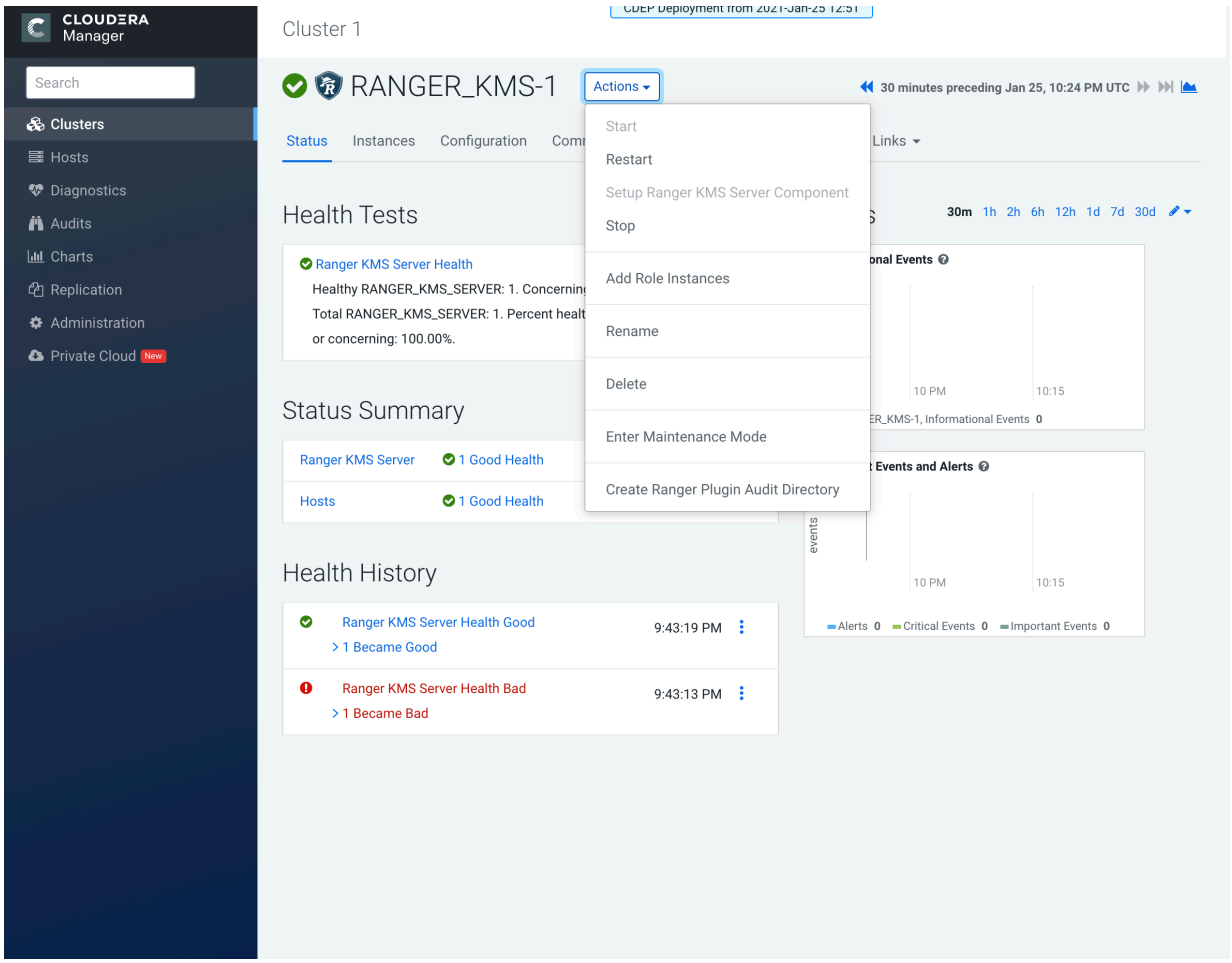
     kms://http@kms_host1;http@kms_host2:9292/kms
   - If SSL is enabled, use https and port 9494. For example:

     kms://http@kms_host1;http@kms_host2:9494/kms

   Click Test Connection to confirm the settings, then click Save to save your changes.



8. In Cloudera Manager, click the Ranger KMS service, then select Actions > Create Ranger Plugin Audit Directory.



9. In Cloudera Manager, select Ranger KMS, then click Configuration.

a) Use the Add (+) icons for the Ranger KMS Server Advanced Configuration Snippet (Safety Valve) for conf/kms-site.xml property to add the following properties, then click Save Changes.

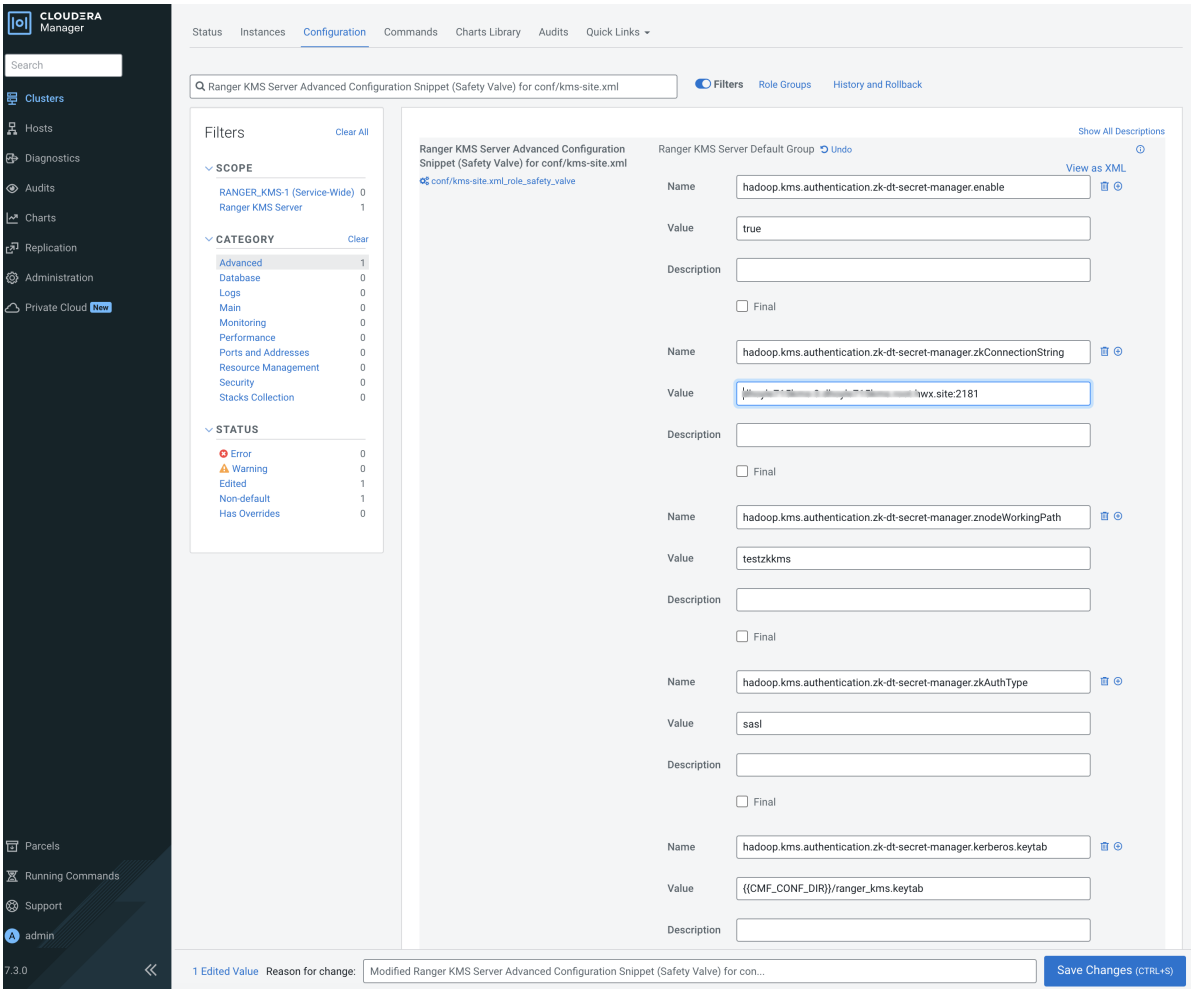
- hadoop.kms.authentication.zk-dt-secret-manager.enable = true
- hadoop.kms.authentication.zk-dt-secret-manager.zkConnectionString = <Zookeeper hostname>:2181

  Note: In a cluster with multiple ZooKeeper hosts, include them as a comma-separated list. For example: hadoop.kms.authentication.zk-dt-secret-manager.zkConnectionString = <ZK_hostname1>:2181,<ZK_hostname2>:2181,....,<ZK_hostnameN>:2181 .
- hadoop.kms.authentication.zk-dt-secret-manager.znodeWorkingPath = <provide a znode working path other than /zkdtsm to avoid collision>

  For example:

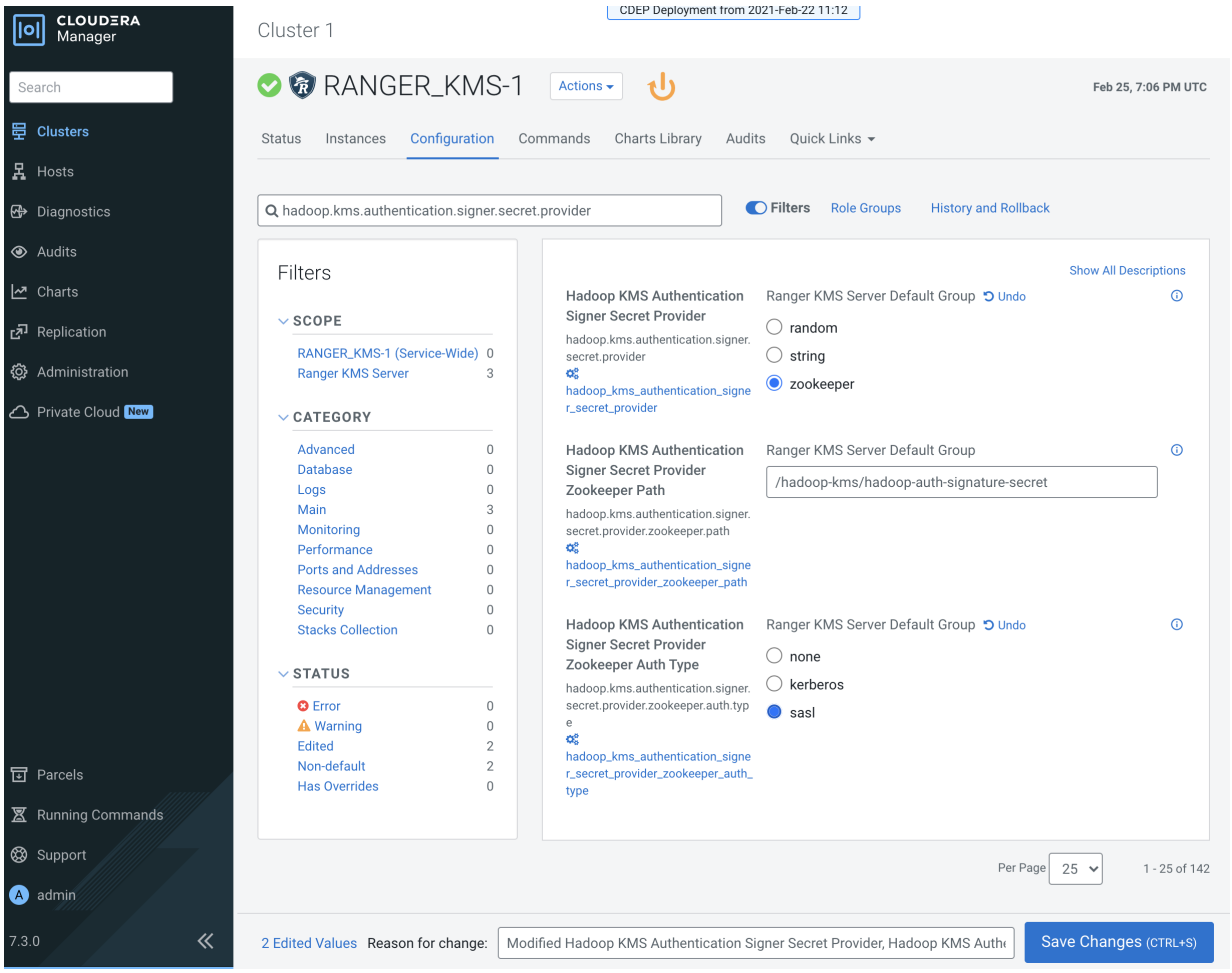
  hadoop.kms.authentication.zk-dt-secret-manager.znodeWorkingPath = testzkkms

  Note: Do not put a leading slash at the beginning of the znode working path.
- hadoop.kms.authentication.zk-dt-secret-manager.zkAuthType = sasl
- hadoop.kms.authentication.zk-dt-secret-manager.kerberos.keytab = {{CMF_CONF_DIR}}/ranger_kms.keytab

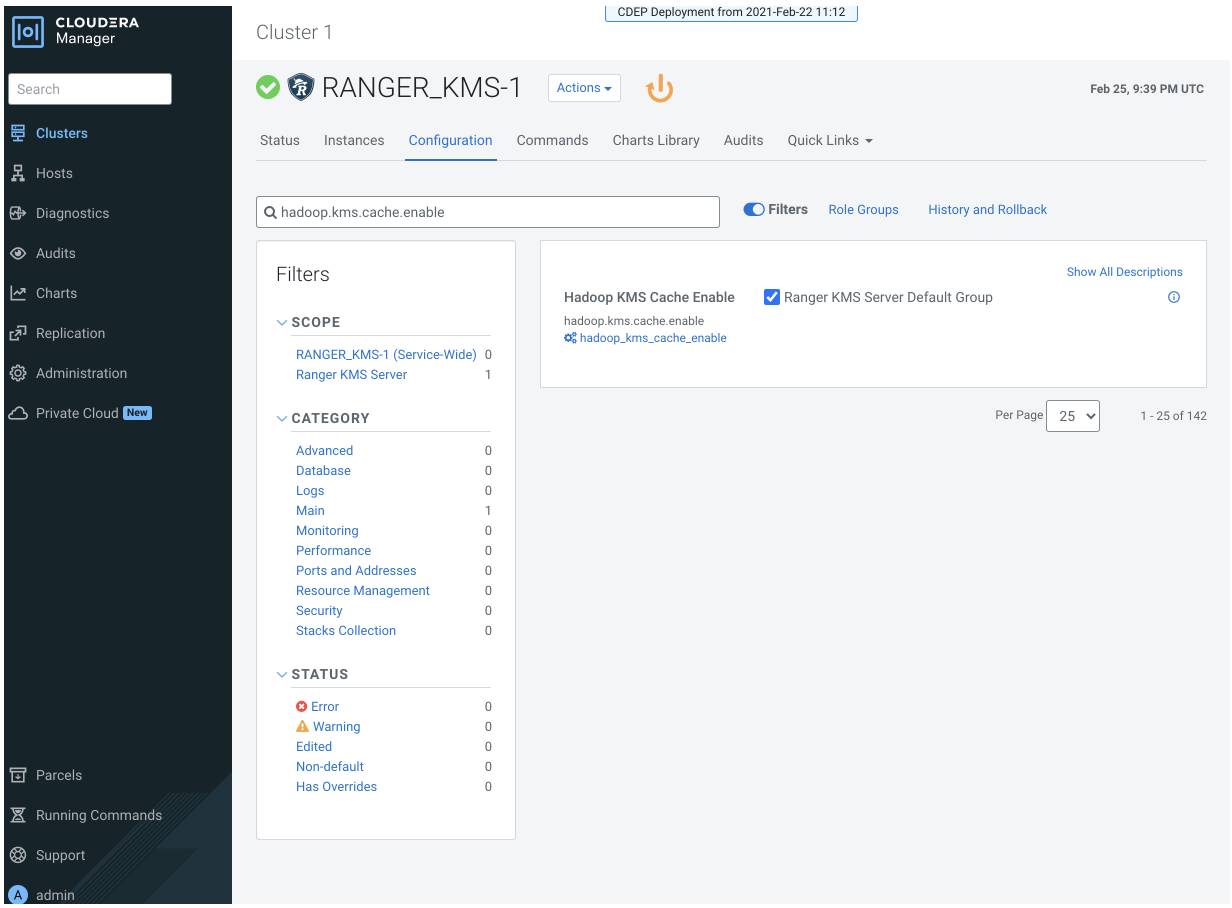


10. Update the following Ranger KMS configuration properties, then click Save Changes.

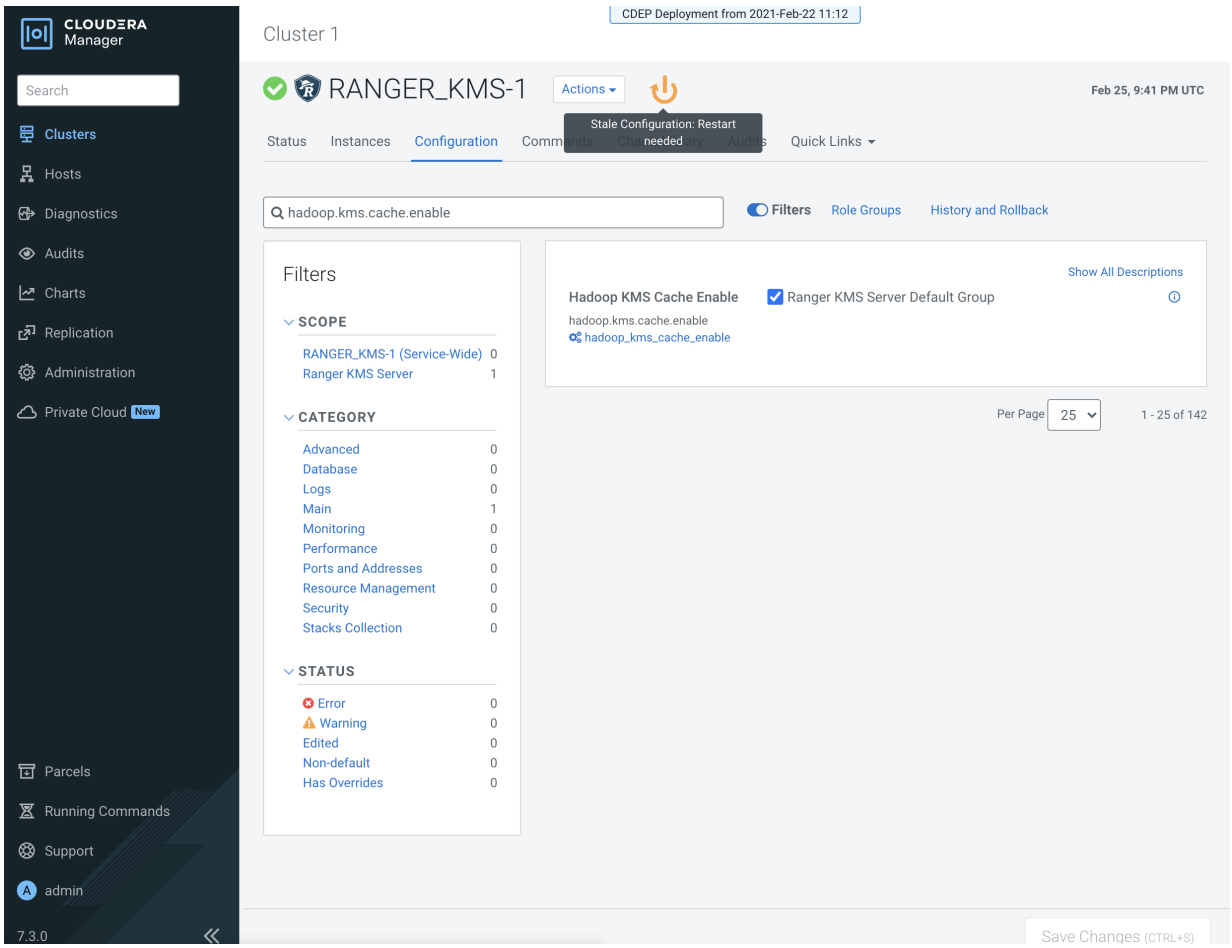
- hadoop.kms.authentication.signer.secret.provider = zookeeper
- hadoop.kms.authentication.signer.secret.provider.zookeeper.auth.type = sasl



11. Verify that the hadoop.kms.cache.enable property is set to the default value of true (the check box is selected).



12. Click the Stale Configuration Restart icon.



13. On the Stale Configurations page, click Restart Stale Services.
14. On the Restart Stale Services page, select the Re-deploy client configuration checkbox, then click Restart Now.
15. A progress indicator page appears while the services are being restarted. When the services have restarted, click Finish.

Rotating Ranger KMS access log files

How to configure properties that control access log file rotation in Ranger KMS service.

About this task

Ranger KMS access log files accrue in the following path: /var/log/ranger/kms/access_log.yyyy-mm-dd.log. By default, these files aren't removed which consumes free space in the /var/ directory. Currently, Ranger KMS access log files get rotated every hour, which amounts to 24 files per day.You can configure it to rotate every 24 hours using the safety valve. To do so, you must add a configuration property to the ranger-kms-site.xml file.

Procedure

1. In Cloudera Manager, select Ranger_KMS, then choose Configuration.
2. On Configuration, in Search, type ranger-kms-site.
3. In Ranger KMS Server Advanced Configuration Snippet (Safety Valve) for conf/ranger-kms-site.xml, click + (Add).
4. Add a key-value pair that configures the rotation of Ranger KMS access log files.

Name

ranger.accesslog.dateformat

Value

yyyy-MM-dd

Note: If not set, then the default value is yyyy-MM-dd.HH.

5. Click Save Changes.

   After saving changes, the Stale Configuration icon appears on the Cloudera Manager UI. Optionally, click Stale Configuration to view details.
6. Select Actions Restart .

Using the Ranger Key Management Service

Ranger Key Management Service (KMS) can be accessed by logging into the Ranger Web UI as the KMS administrator.

Role Separation

Ranger uses separate admin users for Ranger and Ranger KMS.

- The Ranger admin user manages Ranger access policies.
- The Ranger KMS admin user (keyadmin by default) manages access policies and keys for Ranger KMS, and has access to a different set of UI features than the Ranger admin user.

Using separate administrator accounts for Ranger and Ranger KMS separates encryption work (encryption keys and policies) from cluster management and access policy management.

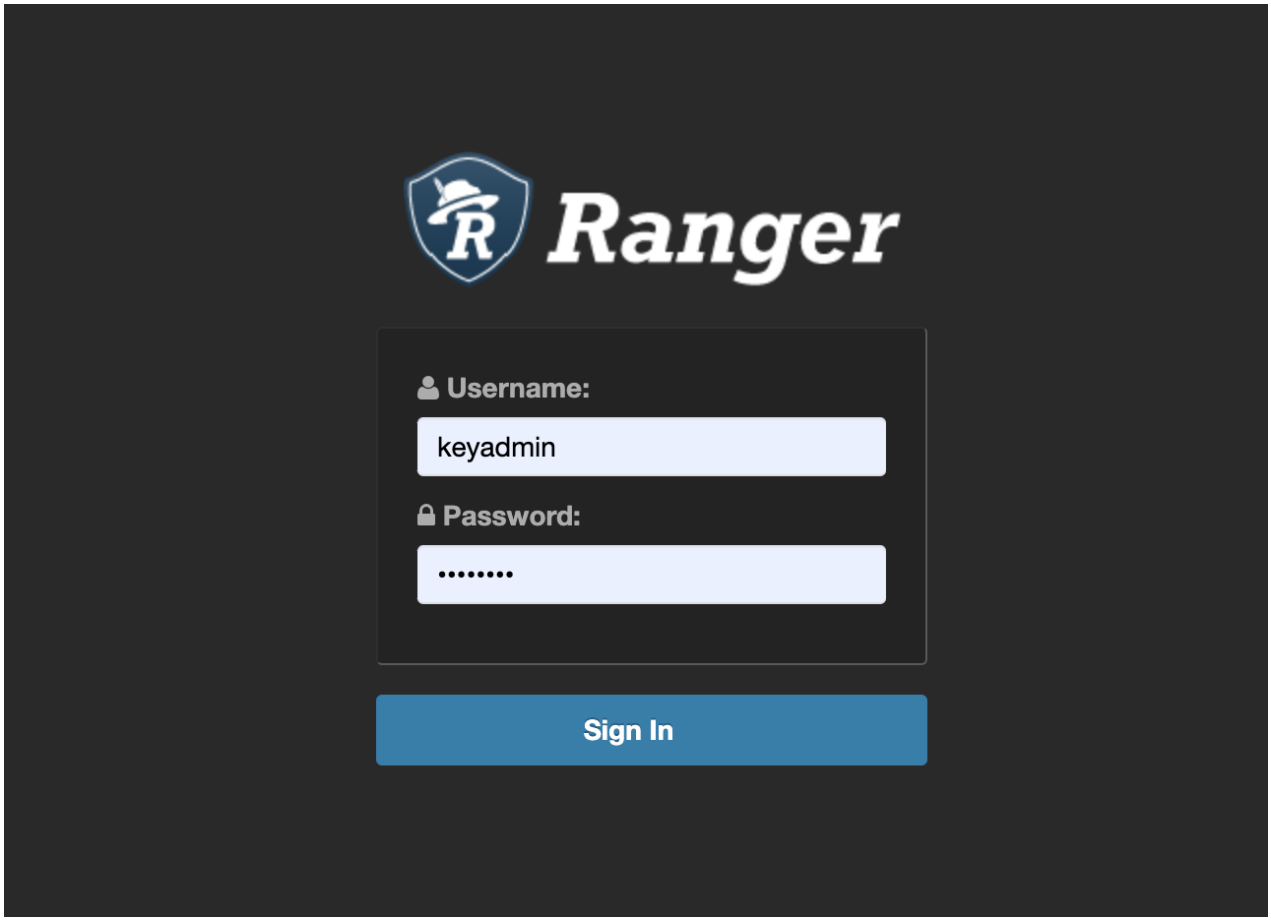
Note:

For more information about creating, deleting, listing, and rolling over existing keys using Ranger REST APIs, see https://ranger.apache.org/apidocs/resource_XKeyREST.html.

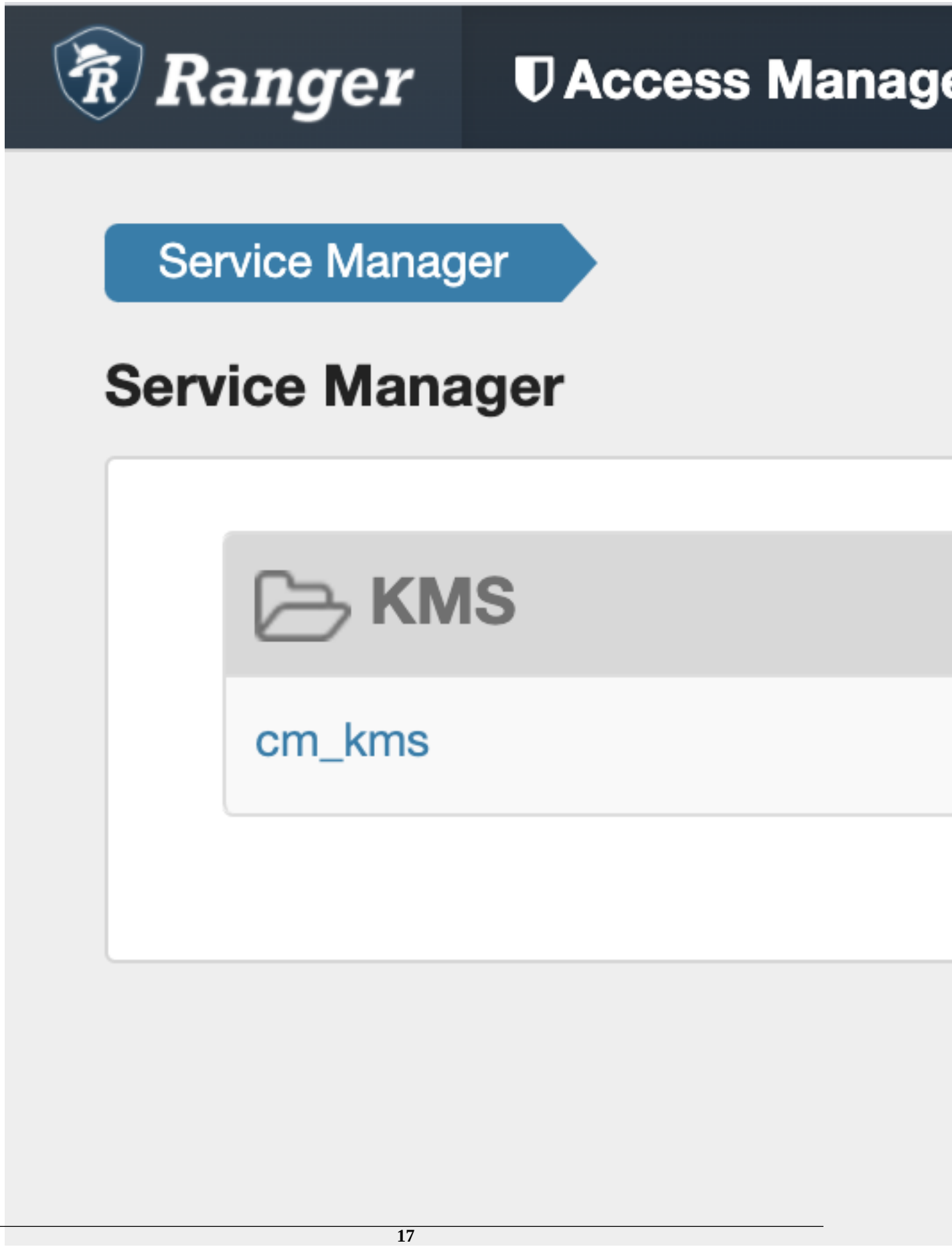
Accessing the Ranger KMS Web UI

How to access the Ranger Key Management Service (KMS) Web UI.

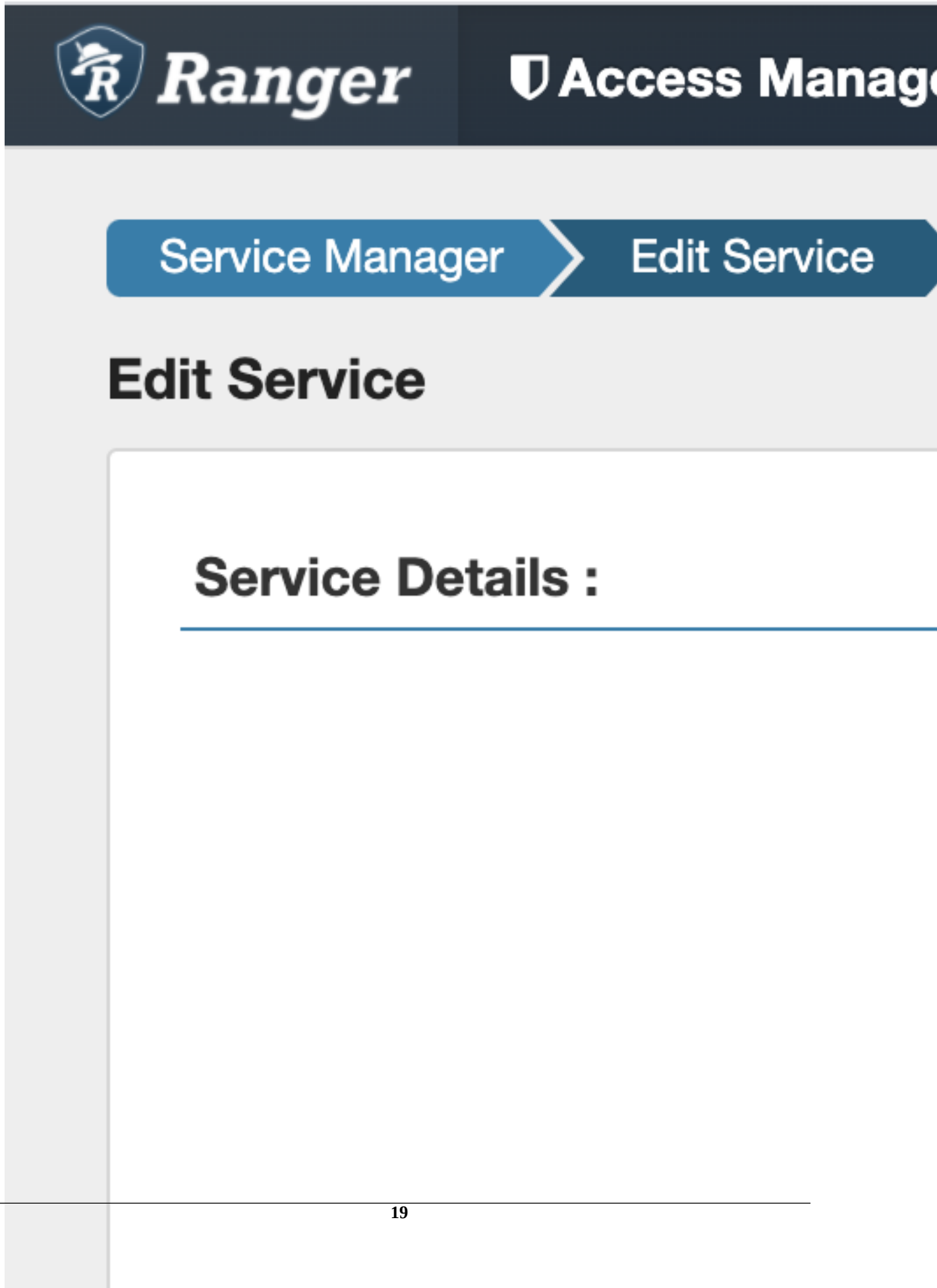
To access Ranger KMS, click the Ranger Admin Web UI link, enter your Ranger KMS admin user name and password, then click Sign In.



After logging in, the **Service Manager** page appears.


Ranger
Access Manage
Service Manager
Service Manager
KMS
cm_kms

To edit Ranger KMS repository properties, click the Edit icon for the service and update the settings on the **Edit Service** page.

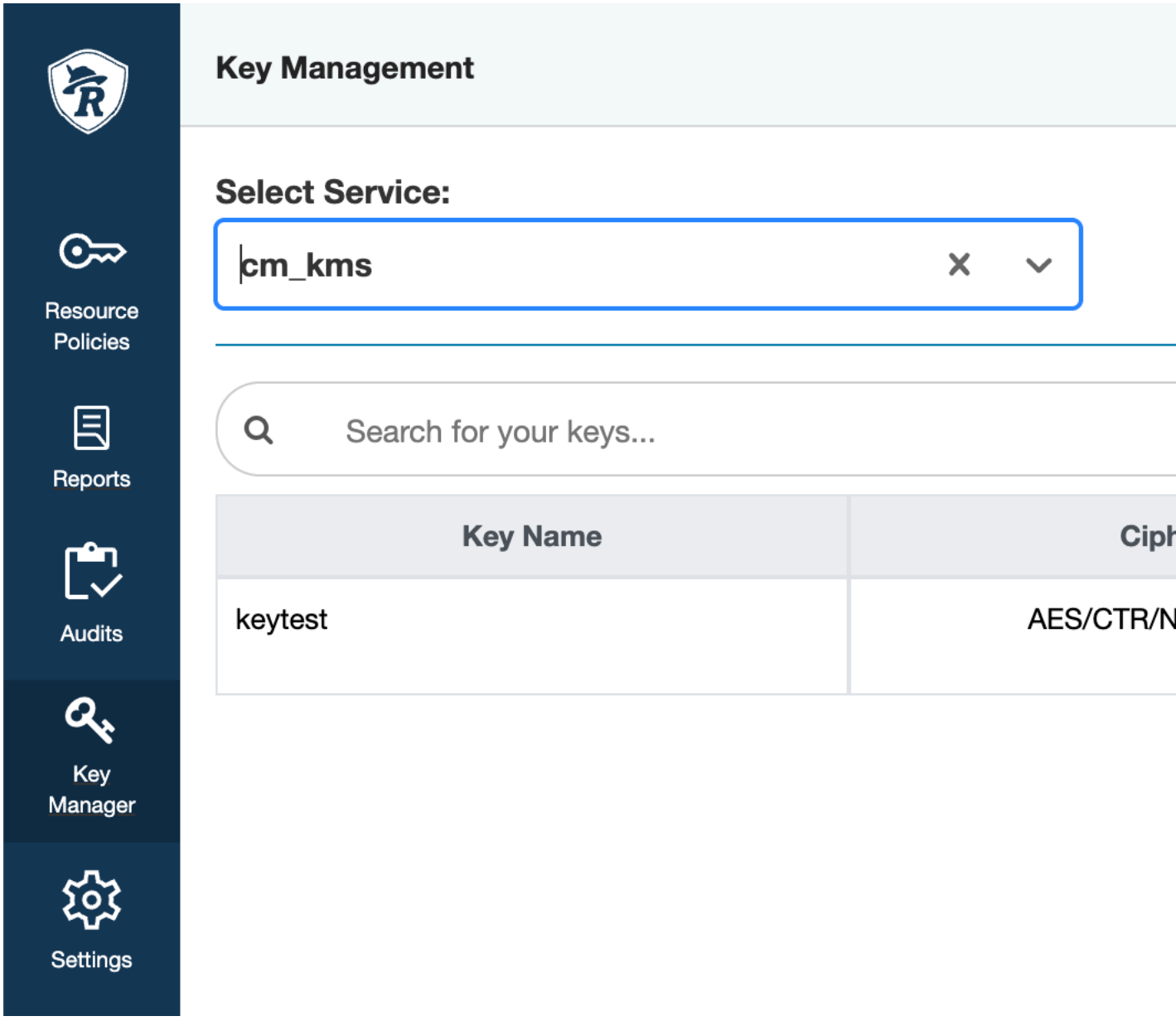

Ranger
Access Manag
Service Manager
Edit Service
Edit Service
Service Details :

List and Create Keys

How to list and create Ranger Key Management Service (KMS) keys.

List existing keys

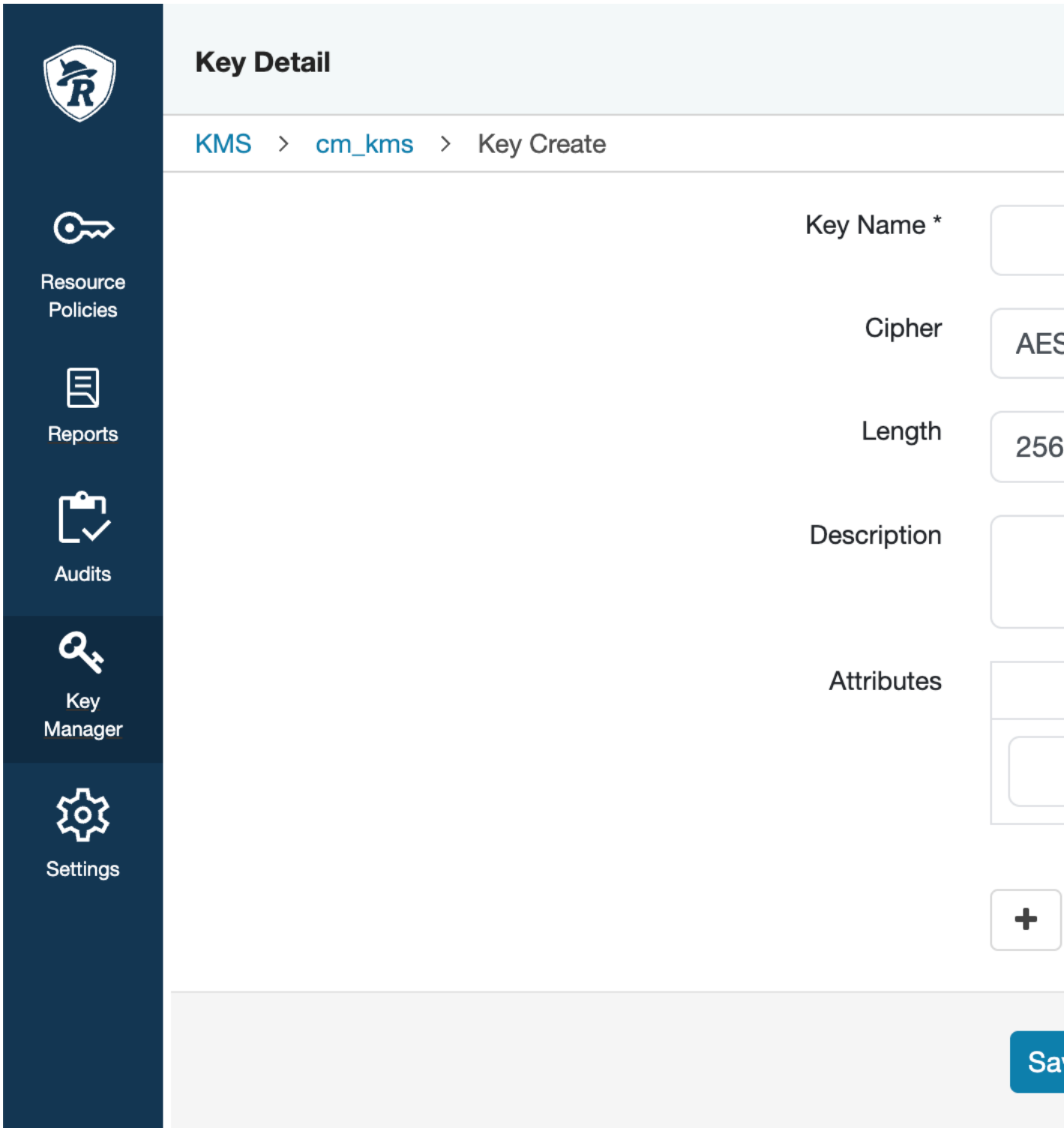
1. Log in to Ranger as the Ranger KMS admin user.
2. Click Encryption in the top menu to display the **Key Management** page.
3. Use the Select Service box to select a Ranger KMS service. The keys for the service appear.



Create a new key

1. Click Add New Key.
2. On the Key Detail page, add a valid key name.
3. Specify a cipher. Ranger KMS supports AES/CTR/NoPadding as the cipher suite.
4. Specify the key length: Default is 256 bits.

5. Add other attributes as needed, then click Save.



Roll Over an Existing Key

How to roll over an existing Ranger Key Management Service (KMS) key.

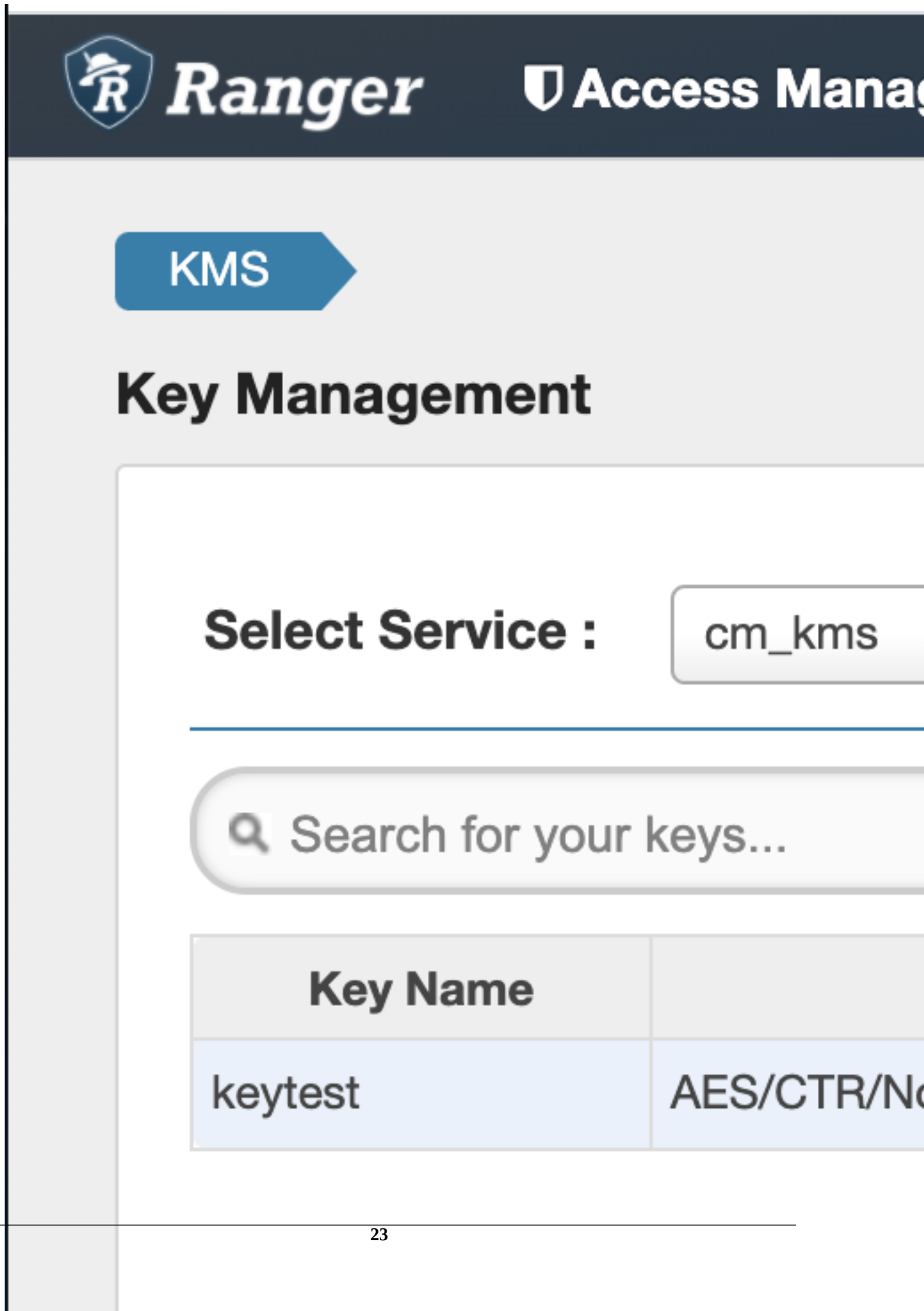
About this task

When you roll over (or rotate) a key, the key retains the same key name, but creates a new version of the key. This newly versioned key becomes the currentKey. After the key rotation, new files will have the file key encrypted by the current encryption zone (EZ) key for the encryption zone.

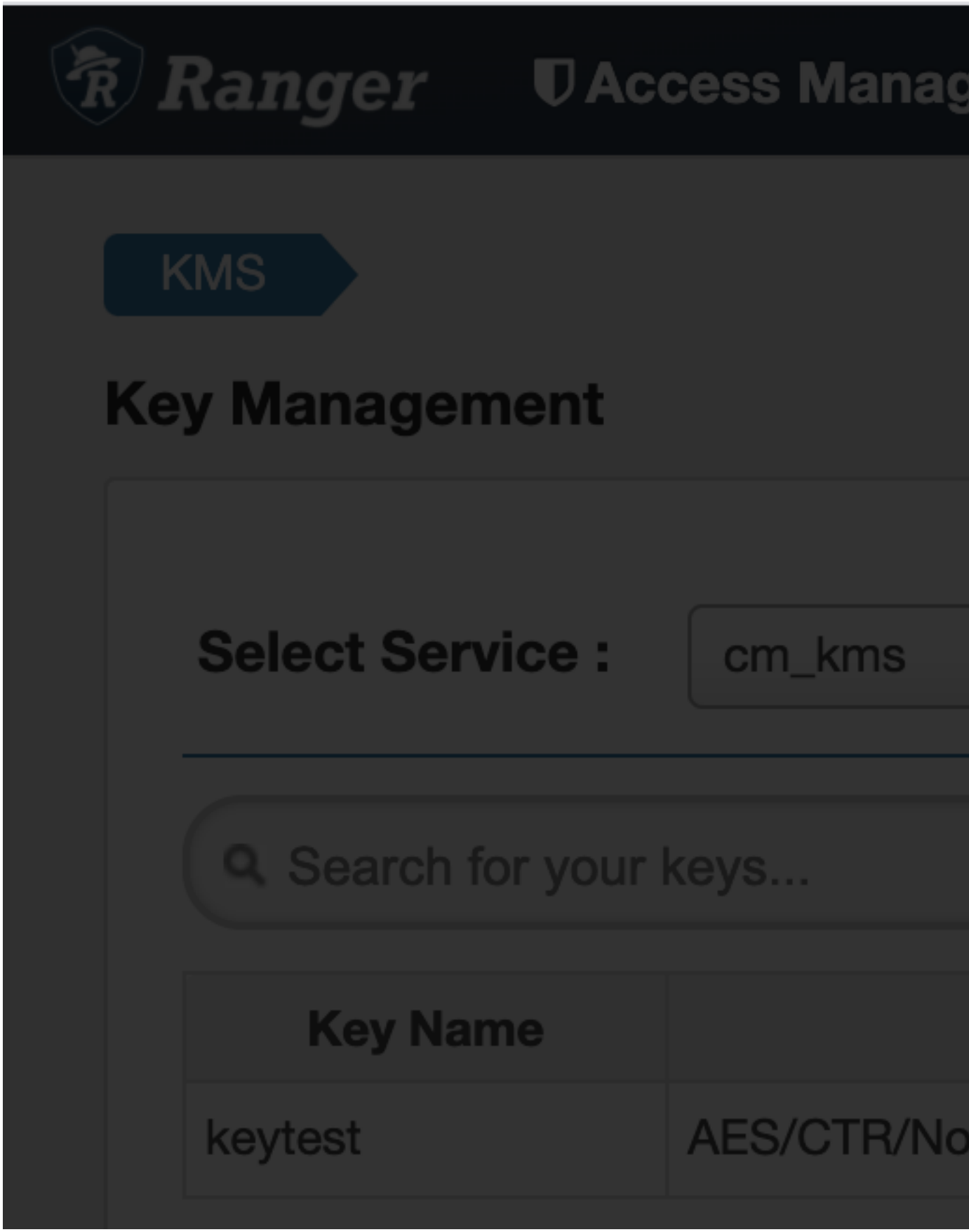
Note: Ranger KMS does not delete older versions of the key, as the older versions are used to decrypt the file keys that were encrypted with them.

Procedure

1. Log in to Ranger as the Ranger KMS admin user, click Encryption in the top menu, then select a Ranger KMS service.
2. To rotate a key, click the Rollover icon for the key in the Action column.


Ranger
Access Manag
KMS
Key Management
Select Service :
cm_kms
Search for your keys...
Key Name
keytest
AES/CTR/N

3. Click OK on the confirmation pop-up.



Delete a Key

How to delete a Ranger Key Management Service (KMS) key.

About this task

Important:

Deleting a key associated with an existing encryption zone will result in data loss.

Note:

- Encryption zone keys should be deleted from the Ranger UI or Hadoop Command line.
- Encryption keys should NOT be deleted in the HSM before deleting from the Ranger UI or Hadoop command line.

Procedure

1. Log in to Ranger as the Ranger KMS admin user, click Encryption in the top menu, then select a Ranger KMS service.
2. Click on the Delete icon for the key in the Action column.
3. Click OK on the confirmation pop-up.

Configuration of cryptographic parameters

This topic outlines the default out-of-the-box cryptographic settings used by Ranger KMS to generate, derive, and encrypt master keys and zone keys.

Note: The cryptographic parameter configurations are available since Cloudera Runtime 7.3.2.10000.

The cryptographic parameter configurations enables you to achieve the following goals:

- Data confidentiality: By encrypting the key material using compliant and strong key derivation algorithm and transformation algorithms.
- Data integrity: A check to protect or detect the stored encrypted bytes from being tampered during the decryption process.
- Authenticity: To verify the sender.

Master key configurations

The master key gets generated when Ranger KMS service starts for the very first time. Under this parameter framework, administrators can override out-of-the-box defaults within configuration profiles to align with organizational compliance standards.

Master key default configurations

When the Ranger KMS service starts for the first time, a master key is automatically generated and stored using the following default cryptographic parameters:

- Key Material Generation Algorithm: AES
- Key Size: 256 bit
- Key Derivation Function (KDF): PBKDF2WithHmacSHA256. This serves as the default KDF across both FIPS and non-FIPS environments.
- Cipher Transformation: AES/CTR/NoPadding

- Salt: Input is provided as configuration (or default) value and Ranger KMS further generates a new salt of required size using the provided input and MessageDigest API.
- Iteration Count: 1000

Hardware Security Module (HSM) Defaults

- Luna HSM: Stored master keys default to a key size of 256 bit (updated from the legacy hard-coded 128-bit setting).
- SafenetKeySecure: Stored master keys default to an algorithm of AES with a key size of 256 bit.

Master key configurable properties

The following properties can be customized for the Ranger KMS master key:

Property Name	Allowed Configurable Options / Formats	Description
ranger.kms.service.masterkey.password.cipher	• AES/CTR/NoPadding (Default) • AES/GCM/NoPadding • AES/CBC/PKCS5Padding • AES/CBC/PKCS7Padding (FIPS only)	Defines the complete Cipher value formatted as <Algorithm>/<MODE>/<PADDING>. KMS parses the first token for generation and the remaining parts for encryption/decryption.
ranger.kms.service.masterkey.password.size	256 (Recommended), 192, 128	Determines the cryptographic key length bit size.
ranger.kms.service.masterkey.password.encryption.algorithm	• PBKDF2WithHmacSHA256 (Recommended • PBEWithMD5AndTripleDES (Supported for backward compatibility but not recommended)	Configures the underlying Key Derivation Function (KDF) name used to derive strong SecretKeys.

Zone key configurations

Previously, cryptographic parameters for zone key handling were heavily hard-coded. This parameter model introduces a hierarchical inheritance structure where zone keys automatically adapt and mirror the master key's underlying cryptographic configuration. However, to provide precise control over high-volume data encryption keys, the framework allows selective configuration overrides for zone keys.

Zone key default configurations

Zone keys are wrapped using a password-based encryption approach (via the SealedObject API). By default, the zone key implementation inherits the master key's cryptographic parameters with a few legacy configuration overrides:

- Key Generation Algorithm: AES
- Key Size: 256 bit
- Key Derivation Function (KDF): Inherits the master key's KDF (PBKDF2WithHmacSHA256).
- Cipher Transformation: Inherits the master key's cipher (AES/CTR/NoPadding).
- Iteration Count: Default value is 20 (retained for backward compatibility).
- Salt Size: Inherits the master key’s salt size.



Note: Salt for zonekey is generated using the SecureRandom API.

Zone key configurable properties

While zone keys dynamically utilize the master key's parameter configurations by default, administrators can specifically override properties for zone keys using the dbks-site.xml file in Cloudera Manager Ranger KMS Configuration :

Parameter	Configuration Property Name	Configurable Options
Key Derivation Algorithm	ranger.kms.service.zonekey.encryption.algorithm	Accepts any KDF supported by the master key, which are as follows: • PBKDF2WithHmacSHA256 (Recommended) • PBEWithMD5AndTripleDES (Supported for backward compatibility but not recommended)
Cipher	ranger.kms.service.zonekey.encryption.cipher	Accepts any cipher transformation available to the master key, which are as follows: • AES/CTR/NoPadding (Default) • AES/GCM/NoPadding • AES/CBC/PKCS5Padding • AES/CBC/PKCS7Padding (FIPS only)
Iteration Count	ranger.kms.service.zonekey.iteration.count	Custom integer count value to optimize wrapping complexity.
Salt Size	ranger.kms.service.zonekey.salt.size	Custom byte size setting for generating unique zone key salts.

Securing the Key Management System (KMS)

Cloudera provides the following Key Mangement System (KMS) implementations: Ranger KMS with database and Ranger KMS with HSM. You can secure Ranger KMS using Kerberos, TLS/SSL communication, and access control lists (ACLs) for operations on encryption keys.

Cloudera Manager supports wizard-driven instructions for installing Ranger KMS with a database.

Enabling Kerberos Authentication for the KMS

You can use Cloudera Manager to enable Kerberos authentication for the KMS.

About this task

Minimum Required Role: Full Administrator

Procedure

1. Open the Cloudera Manager Admin Console and go to the KMS service.
2. Click Configuration.
3. Set the Authentication Type property to kerberos.
4. Click Save Changes.
5. Because Cloudera Manager does not automatically create the principal and keytab file for the KMS, you must run the Generate Credentials command manually.

   On the top navigation bar, go to Administration Security Kerberos Credentials and click Generate Missing Credentials

   Note: This does not create a new Kerberos principal if an existing HTTP principal exists for the KMS host.
6. Return to the home page by clicking the Cloudera Manager logo.
7. Click the icon that is next to any stale services to invoke the cluster restart wizard.
8. Click Restart Stale Services.

9. Click Restart Now.
10. Click Finish.

Configuring TLS/SSL for the KMS

You must configure specific TLS/SSL properties associated with the KMS.

About this task

Minimum Required Role: Configurator (also provided by Cluster Administrator, Full Administrator)

Procedure

1. Go to the KMS service.
2. Click Configuration.
3. In the Search field, type TLS/SSL to show the KMS TLS/SSL properties (in the Key Management Server Default Group Security category).
4. Edit the following TLS/SSL properties according to your cluster configuration.

Property	Description
Enable TLS/SSL for Key Management Server	Encrypt communication between clients and Key Management Server using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (TLS/SSL)).
Key Management Server TLS/SSL Server JKS Keystore File Location	The path to the TLS/SSL keystore file containing the server certificate and private key used for TLS/SSL. Used when Key Management Server is acting as a TLS/SSL server. The keystore must be in JKS format.
Key Management Server TLS/SSL Server JKS Keystore File Password	The password for the Key Management Server JKS keystore file.
Key Management Server Proxy TLS/SSL Certificate Trust Store File	The location on disk of the truststore, in .jks format, used to confirm the authenticity of TLS/SSL servers that Key Management Server Proxy might connect to. This is used when Key Management Server Proxy is the client in a TLS/SSL connection. This truststore must contain the certificates used to sign the services connected to. If this parameter is not provided, the default list of well-known certificate authorities is used instead.
Key Management Server Proxy TLS/SSL Certificate Trust Store Password	The password for the Key Management Server Proxy TLS/SSL Certificate Trust Store File. This password is not required to access the truststore; this field can be left blank. This password provides optional integrity checking of the file. The contents of truststores are certificates, and certificates are public information.

5. Click Save Changes.
6. Return to the home page by clicking the Cloudera Manager logo.
7. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.
8. Click Restart Stale Services.
9. Click Restart Now.
10. Click Finish.

Migrating Ranger Key Management Server Role Instances to a New Host

You can move the Ranger Admin and Ranger KMS database role instances for an existing Ranger KMS service from one host to another, using Cloudera Manager.



Note: This procedure applies only to the Ranger Key Management Server role instances.

In some cases–for example, after upgrading your servers–you may want to migrate a Ranger KMS Server role instance to a new host. This procedure describes how to move a Ranger KMS role instance from an existing cluster host to another cluster host.

Migrate the Ranger Admin role instance to a new host

To migrate the Ranger KMS role instances to a new host, first migrate the Ranger Admin role instance.

Procedure

1. Add a new Ranger Admin role instance on another node.

   Note: If you enabled manual SSL on this cluster, you must update the SSL configs when adding a new role.

2. Start the new Ranger Admin role instance.
3. Stop the initial Ranger Admin instance.
4. Delete the initial Ranger Admin instance.
5. Restart the cluster.

   Restarting the cluster removes the "stale" changes.

Migrate the Ranger KMS DB role instance to a new host

After migrating the Ranger Admin role instance to a new host, migrate the Ranger KMS DB role instance.

About this task

Only if Ranger KMS has a backend database for key storage, should you migrate the Ranger KMS DB role instance.

Procedure

1. Add a new Ranger KMS DB role instance on another node.

   Note: If you enabled manual SSL on this cluster, you must update the SSL configs when adding a new role.

2. Start the new Ranger KMS DB role instance.
3. Stop the initial Ranger KMS DB instance.
4. Delete the initial Ranger KMS DB instance.
5. Restart the cluster.
6. Login to Ranger Admin UI using keyadmin credentials.
7. Update the cm_kms service to use the kms url that refers to the new hostname.

Working with an HSM

How to integrate Cloudera data encryption components to provide enterprise data encryption solutions.

Ranger Key Mangement System

Consists of Ranger KMS providing enterprise-grade key management with a backend database that provides key storage.

1. Install Ranger KMS using Cloudera Manager Administration Security HDFS Encryption Wizard .

2. Install a seperate database to store keys.

   For more information, see related links.

Ranger KMS and HSM

Consists of Ranger KMS and database integrated with a backend hardware security module (HSM). In this solution, Ranger KMS provides enterprise-grade key management, and HSM provides encryption zone key protection. HSM stores only the encryption master key.

1. Install Ranger KMS using Cloudera Manager Administration Security HDFS Encryption Wizard .
2. Install a seperate database to store keys.
3. Obtain and integrate one of the following hardware security modules (HSM) supplied by a vendor.
   - Luna 7
   - CipherTrust
   - GCP Cloud HSM
   - Azure Key Vault

   For more information, see related links.

Set up Luna 7 HSM for Ranger KMS

How to integrate Cloudera Ranger Key Management System software with the Luna 7 HSM appliance supplied by SafeNet.

About this task

This task describes how to set up the Luna 7 hardware security moudule (HSM) supplied by SafeNet. The process inlcudes setting up Luna 7 HSM on a client (host) and using Cloudera Manager to add configuration properties that enable Ranger KMS and Luna 7 HSM to interact.



Note: Ranger KMS should have separate HSM partition per KMS setup. If the same HSM partition is configured in the second KMS cluster, it overwrites the existing master key in the HSM partition.

Before you begin

You must:

- Acquire the Luna 7 HSM from SafeNet.
- If the Luna HSM module is configured for FIPS mode, you must add the following additional configuration option to the Luna client:

  ```
  /usr/safenet/lunaclient/bin/configurator setValue -s Misc -e RSAKeyGenMe
  chRemap -v 1
  ```

- Have both Ranger KMS and a backend database to store keys installed in your environment.

See related topics for more information about installing Ranger KMS and a database to store keys.

Procedure

Set Up the Luna 7 Client

1. Download Luna 7 client on the host where Ranger KMS service resides.

   ```
   610-013144-006_SW_Client_SDK_SafeNet_HSM_7.3.0_Linux_RevA.tar
   ```

2. Untar the Luna 7 client.

```
tar -xf 610-013144-006_SW_Client_SDK_SafeNet_HSM_7.3.0_Linux_RevA.tar
```

the LunaClient_7.3.0-165_Linux/ folder gets created.

3. Navigate to the Luna client folder.

```
cd LunaClient_7.3.0-165_Linux/64/
```

4. In the Luna client folder, install Luna products and components.

```
bash install.sh
```

a) At the (y/n) prompt, choose y.

If you select no or n, this product will not be installed.

b) At the Products prompt, choose Luna products to be installed:

- [1]: Luna Network HSM
- [2]: Luna PCIe HSM
- [3]: Luna USB HSM
- [4]: Luna Backup HSM
- [N|n]: Next
- [Q|q]: Quit

Enter selection: 1, then enter selection n.

c) At the Components prompt, choose Luna Components to be installed

- [1]: Luna SDK
- [2]: Luna JSP (Java)
- [3]: Luna JCProv (Java)
- [B|b]: Back to Products selection
- [I|i]: Install
- [Q|q]: Quit

Enter selection: i, then enter selection Q.
Enter selection: 1,2,and 3 then type i.

5. Navigate to the Luna SA command directory.

```
cd /usr/safenet/lunaclient/bin
```

You should see the following:

ls

ckdemo cmu common configurator lunacm multitoken openssl.cnf plink pscp

salogin uninstall.sh vtl

6. Add a user to the hsmusers group.

```
sudo gpasswd --add kms hsmusers
```

7. Copy the Luna appliance server certificate to the client.

```
scp admin@<LunaBoxHostname>:server.pem
```

```
 scp e02paruser115@elab2.safenet-inc.com:server.pem .
(grant permission chmod 777 and chown kms:kms)
```

```
The authenticity of host 'elab2.safenet-inc.com (192.43.161.62)' can't be
established.
ECDSA key fingerprint is SHA256:Lz36zjWHh3BMtI9TVHUBGoHffxgA6azFtPSGRBC
kiYU.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'elab2.safenet-inc.com,192.43.161.62' (ECDSA) t
o the list of known hosts.
e02paruser115@elab2.safenet-inc.com's password:  SafeNetPSG95 (given by
 the luna hsm team)
  press enter
server.pem                                    100% 1155      1.1KB/s
          00:00
```

8. Confirm that server.pem is added to the client.

```
ls
```

```
ckdemo cmu common configurator lunacm multitoken openssl.cnf plink pscp
 salogin server.pem uninstall.sh vtl
```

server.pem is added

9. As the KMS user, register the server with the client.

```
su -l kms
./vtl addServer -n <LunaBoxHostname> -c server.pem
```

```
 ./vtl addserver -n elab2.safenet-inc.com -c server.pem
```

vtl (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.

New server elab2.safenet-inc.com successfully added to server list.

10. Generate a client certificate.

```
./vtl  createCert -n  <ClientHostname>
```

```
 ./vtl createcert -n e02paruser115
```

vtl (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.

Private Key created and written to: /usr/safenet/lunaclient/cert/client/e02paruser115Key.pem. Certificate created and written to: /usr/safenet/lunaclient/cert/client/e02paruser115.pem .

(grant permission chmod 777 and chown kms:kms)

11. Copy the client certificate to the server.

```
scp /usr/safenet/lunaclient/cert/client/<ClientHostname>.pem admin@<Luna
BoxHostname>:
```

```
scp /usr/safenet/lunaclient/cert/client/e02paruser115.pem e02paruser115@
elab2.safenet-inc.com:
e02paruser115@elab2.safenet-inc.com's password: SafeNetPSG95
e02paruser115.pem                                             100%
1172   201.7KB/s   00:00
```

12. Login to luna hsm.

```
ssh admin@<lunaboxhostname>
```

```
ssh e02paruser115@elab2.safenet-inc.com
e02paruser115@elab2.safenet-inc.com's password: SafeNetPSG95
Last login: Fri Jul 19 03:59:38 2019 from 114.143.87.94
Luna Network HSM Command Line Shell v7.3.0-165.
Copyright (c) 2018 SafeNet. All rights reserved.
[elab2] lunash:>
```

13. Register the client with the server, then assign the client to a server partition.

```
lunash:> client register -client <ClientHostname> -hostname <ClientHostn
ame>
```

```
client register -client e02paruser115 -hostname e02paruser115
```

14. Check the existing partitions.

```
lunash:> partition list
```

```
lunash:> partition list
                        Storage (bytes)
                        ---------------------------
                        Partition            Name                    Obj
ects   Total    Used    Free
                        ============================================
============================
                        1254277068838        elab2par058
 0  325896       0  325896
```

15. Assign client to the partition.

```
lunash:> client assignPartition -client <ClientHostname> -partition <Gat
ewayPartition>
```

```
lunash:> client assignPartition -client e02paruser115 -partition elab2pa
r058
```

16. client show -client e02paruser115

```
ClientID:     e02paruser115
Hostname:     e02paruser115
Partitions:   "elab2par058"
```

17. Log out from the Luna HSM.

```
lunash:> exit
```

18. Set the read permissions for the certificate files in the following directories.



Note: Make sure to log in as root user.

```
chmod a+r /usr/safenet/lunaclient/cert/server/*.pem
chmod a+r /usr/safenet/lunaclient/cert/client/*.pem
```

```
(grant permission chmod 777 and chown kms:kms to above .pem files)
```

19. Verify that the client is connected to its assigned partition.



Note: Make sure to log in as kms user.

```
cd /usr/safenet/lunaclient/bin/
./vtl verify
```

```
[root@os-mv-711-1 bin]# ./vtl verify
vtl (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.

The following Luna SA Slots/Partitions were found:

Slot            Serial #                    Label
==========================================================================
===
0               1254277068842             elab2par115
```

20. ./lunacm

```
./lunacm
```

```
[root@os-mv-711-1 bin]# ./lunacm
lunacm (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights rese
rved.

Available HSMs:

Slot ID ->           0
Label ->             elab2par115
Serial # ->          1254277068842
Model ->             LunaSA 7.3.0
Firmware version ->  7.3.0
Configuration ->     Luna User Partition with SD (PW) Key Export with Cl
eaning Mode
Slot Description ->  Net Token Slot

Current Slot ID:  0
```

21. role login -n co

```
enter password: pwd123
```

22. par con

If Master Key RangerKMSKey exists, then the following will be visible:

```
lunacm:>par con
    The 'Crypto Officer' is currently logged in.
    Looking for objects accessible to the 'Crypto Officer'.

    Object List:

    Label:            RangerKMSKey
    Handle:           131
    Object Type:      Symmentric Key
    Object UID:       ba8e00002e00000554380800
    Number of Objects: 1
```

```
Command Result: No Error
Else
lunacm:>par con

    The 'Crypto Officer' is currently logged in.
    Looking for objects accessible to the 'Crypto Officer'.

    No objects viewable to 'Crypto Officer' are currently stored in the
 partition.
Command Result: No Error
```

23. Navigate to the following directory on the Gateway.

```
# cd /usr/safenet/lunaclient/jsp/lib/
(grant permission chmod 777 and chown kms:kms to all the at this location)
```

24. Copy the Luna .JAR files over to the Gateway.

```
cp libLunaAPI.so Luna*.jar {JAVA_HOME}/jre/lib/ext/
```

25. Set the file permissions for the JDK library as follows:

```
chmod a+r {JAVA_HOME}/jre/lib/
```

26. Open the following file in a text editor:

```
vim {JAVA_HOME}/jre/lib/security/java.security
```

a) Add these two lines:

```
security.provider.6=com.safenetinc.luna.provider.LunaProvider
com.safenetinc.luna.provider.createExtractableKeys=true
```

replacing the line highlighted below:

```
Java SDK/JRE 1.6.x or 1.7.x installation to read as follows:
security.provider.1=sun.security.provider.Sun
security.provider.2=sun.security.rsa.SunRsaSign
security.provider.3=sun.security.ec.SunEC
security.provider.4=com.sun.net.ssl.internal.ssl.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=com.safenetinc.luna.provider.LunaProvider
security.provider.7=sun.security.jgss.SunProvider
security.provider.8=com.sun.security.sasl.Provider
security.provider.9=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.10=sun.security.smartcardio.SunPCSC
```

27. Set the file permissions for the Luna client as follows:

```
chmod -R 777 /usr/safenet
chown kms:kms
```

Set KMS Configuration Properties.

28. In Cloudera Manager Ranger KMS Configs edit the following properties:

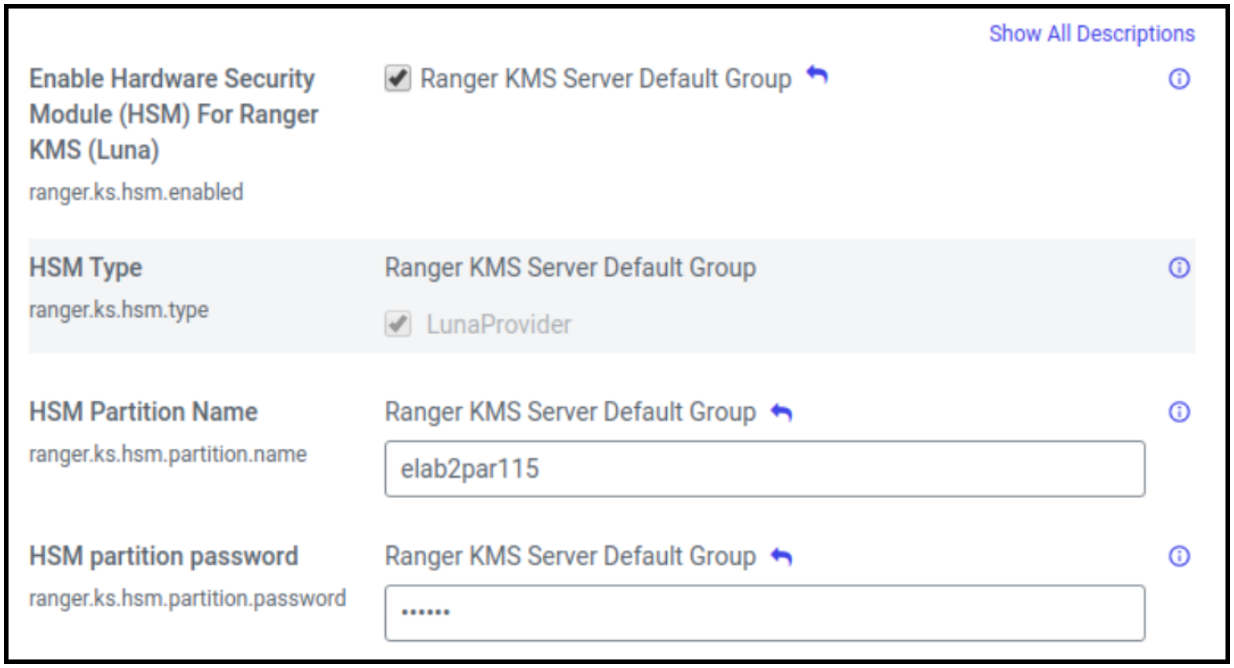
Note: For Cloudera Manager 7.1.1 and Cloudera Manager 7.1.2 you must add properties to the dbks-site.xml, also known as

Ranger KMS Server Advanced Configuration Snippet (Safety Valve) for conf/dbks-site.xml .

```
ranger.ks.hsm.type = LunaProvider
ranger.ks.hsm.enabled = true
ranger.ks.hsm.partition.name=elab2par115
ranger.ks.hsm.partition.password=pwd123
(CM-7.1.1 & CM-7.1.2 password will be in plain text)
```

Note: For Cloudera Manager 7.1.3 and higher, update the configuration as shown in:

Figure 1: Adding Ranger KMS Configuration for Luna 7 HSM



29. Restart Ranger KMS from Cloudera Manager.
30. Login to Luna client and validate whether the master key is successfully created.

```
cd /usr/safenet/lunaclient/bin/
./lunacm
role login -n co

enter password: pwd123

par con
```

```
lunacm:>par con
    The 'Crypto Officer' is currently logged in.
    Looking for objects accessible to the 'Crypto Officer'.

    Object List:

    Label:              RangerKMSKey
    Handle:             131
    Object Type:        Symmentric Key
    Object UID:         ba8e00002e00000554380800
```

```
    Number of Objects: 1

Command Result: No Error
```

Results

Ranger KMS is successfully started.

What to do next

You can now create Encryption zone keys using hadoop command or from Ranger UI using credentials of keyadmin user.

Set up Luna 10.5 HSM Client for Ranger KMS

How to integrate Cloudera Ranger Key Management System software with the Luna 10.5 HSM appliance supplied by SafeNet.

About this task

This task describes how to set up the Luna 10.5 hardware security moudule (HSM) supplied by SafeNet. The process inlcudes setting up Luna 10.5 HSM on a client (KMS host) and using Cloudera Manager to add configuration properties that enable Ranger KMS and Luna 10.5 HSM to interact.

Note: Ranger KMS should have separate HSM partition per KMS setup. If the same HSM partition is configured in the second KMS cluster, it overwrites the existing master key in the HSM partition.

Before you begin

You must:

- Acquire the Luna v10.5 client, HSM Software Version v7.3.0, and HSM Firmware v7.3.0 from SafeNet.
- Have both Ranger KMS and a backend database to store keys installed in your environment.

See related topics for more information about installing Ranger KMS and a database to store keys.

Procedure

Set Up the Luna 10.5 Client

1. Download Luna 10.5 client on the host where Ranger KMS service resides.

   ```
   610-000397-006_SW_Linux_Luna_Client_V10.5.0_RevA.tar
   ```

2. Untar the Luna 10.5 client.

   ```
   tar -xf 610-000397-006_SW_Linux_Luna_Client_V10.5.0_RevA.tar
   ```

   The LunaClient_10.5.0-*_Linux/ folder gets created.

3. Navigate to the Luna client folder.

   ```
   cd LunaClient_10.5.0-*_Linux/64/
   ```

4. In the Luna client folder, install Luna products and components.

   ```
   bash install.sh
   ```

   a) At the (y/n) prompt, choose y.

   If you select no or n, this product will not be installed.

b) At the Products prompt, choose Luna products to be installed:

- [1]: Luna Network HSM
- [2]: Luna PCIe HSM
- [3]: Luna USB HSM
- [4]: Luna Backup HSM
- [N|n]: Next
- [Q|q]: Quit

Enter selection: 1, then enter selection n.

c) At the Components prompt, choose Luna Components to be installed

- [1]: Luna SDK
- [2]: Luna JSP (Java)
- [3]: Luna JCProv (Java)
- [B|b]: Back to Products selection
- [I|i]: Install
- [Q|q]: Quit

Enter selection: 1,2,and 3 then type i.

5. Navigate to the Luna SA command directory.

```
cd /usr/safenet/lunaclient/bin
```

You should see the following:

ls

ckdemo cmu common configurator lunacm multitoken openssl.cnf plink pscp

salogin uninstall.sh vtl

6. Add a user to the hsmusers group.

```
sudo gpasswd --add kms hsmusers
```

7. Copy the Luna appliance server certificate to the client.

```
scp admin@<LunaBoxHostname>:server.pem .
```

8. Confirm that server.pem is added to the client.

```
ls
```

server.pem is added

9. As the KMS user, register the server with the client.

```
./vtl addServer -n <LunaBoxHostname> -c server.pem
```

10. Generate a client certificate.

```
./vtl  createCert -n  <ClientHostname>
```

Private Key created and written to: /usr/safenet/lunaclient/cert/client/e02paruser115Key.pem. Certificate created and written to: /usr/safenet/lunaclient/cert/client/e02paruser115.pem .

(grant permission chmod 777 and chown kms:kms)

11. Copy the client certificate to the server.

```
scp /usr/safenet/lunaclient/cert/client/<ClientHostname>.pem admin@<Luna
BoxHostname>:
```

12. Login to luna hsm with the password.

```
ssh admin@<lunaboxhostname>
```

13. Register the client with the server, then assign the client to a server partition.

```
lunash:> client register -client <ClientHostname> -hostname <ClientHostn
ame>
```

14. Check the existing partitions.

```
lunash:> partition list
```

Example:

```
lunash:> partition list
                        Storage (bytes)
                        ---------------------------
                        Partition            Name                   Obj
ects   Total    Used    Free
                        ===============================================
============================
                        1254277068838        elab2par058
 0  325896       0  325896
```

15. Assign client to the partition.

```
lunash:> client assignPartition -client <ClientHostname> -partition <Gat
ewayPartition>
```

Example :

```
lunash:> client assignPartition -client e02paruser115 -partition elab2pa
r058
```

16. client show -client e02paruser115
Example:

```
ClientID:     e02paruser115
Hostname:     e02paruser115
Partitions:   "elab2par058"
```

17. Log out from the Luna HSM.

```
lunash:> exit
```

18. Set the read permissions for the certificate files in the following directories.



Note: Make sure to log in as root user.

Example :

```
chmod a+r /usr/safenet/lunaclient/cert/server/*.pem
chmod a+r /usr/safenet/lunaclient/cert/client/*.pem
```

```
(grant permission chmod 777 and chown kms:kms to above .pem files)
```

19. Verify that the client is connected to its assigned partition.



Note: Make sure to log in as kms user.

```
cd /usr/safenet/lunaclient/bin/
./vtl verify
```

```
[root@os-mv-711-1 bin]# ./vtl verify


The following Luna SA Slots/Partitions were found:

Slot            Serial #                    Label
==========================================================================
==
0                1254277068842             elab2par115
```

Troubleshooting : If you get following error : Application "vtl" has detected "locale::facet::_S_create_c_locale name not valid" , then

```
export LC_ALL="C"
```

and re-execute the command.

20. ./lunacm

```
./lunacm
```

```
[root@os-mv-711-1 bin]# ./lunacm


Available HSMs:

Slot ID ->           0
Label ->             elab2par115
Serial # ->          1254277068842
Model ->             LunaSA 7.3.0
Firmware version ->  7.3.0
Configuration ->     Luna User Partition with SD (PW) Key Export with Cl
eaning Mode
Slot Description ->  Net Token Slot

Current Slot ID:  0
```

21. role login -n co

```
enter password: passwrd123
```

22. par con

If Master Key RangerKMSKey exists, then the following will be visible:

```
lunacm:>par con
    The 'Crypto Officer' is currently logged in.
    Looking for objects accessible to the 'Crypto Officer'.

    Object List:
```

```
    Label:             RangerKMSKey
    Handle:            131
    Object Type:       Symmentric Key
    Object UID:        ba8e00002e00000554380800
    Number of Objects: 1

Command Result: No Error
Else
lunacm:>par con

    The 'Crypto Officer' is currently logged in.
    Looking for objects accessible to the 'Crypto Officer'.

    No objects viewable to 'Crypto Officer' are currently stored in the
 partition.
Command Result: No Error
```

23. Navigate to the following directory on the Gateway.

```
# cd /usr/safenet/lunaclient/jsp/lib/
(grant permission chmod 777 and chown kms:kms to all the at this location)
```

24. Copy the Luna .JAR files over to the classpath.

```
cp Luna*.jar /opt/cloudera/parcels/CDH/lib/ranger-kms/ews/webapp/lib/
(grant permission chmod 777 and chown kms:kms to Luna jar at this location
.)
```

25. To add the .so files to the java.library.path, execute the following command and note down the already existing value for -Djava.library.path.

```
ps -ef | grep “kms”
```

26. Open the following file in a text editor:

```
vim {JAVA_HOME}/conf/security/java.security
```

a) Add these two lines:

```
security.provider.10=com.safenetinc.luna.provider.LunaProvider
com.safenetinc.luna.provider.createExtractableKeys=true
```

For example,

```
Java SDK/JRE 17.x installation to read as follows:
security.provider.1=SUN
security.provider.2=SunRsaSign
security.provider.3=SunEC
security.provider.4=SunJSSE
security.provider.5=SunJCE
security.provider.6=SunJGSS
security.provider.7=SunSASL
security.provider.8=XMLDSig
security.provider.9=SunPCSC
security.provider.10=com.safenetinc.luna.provider.LunaProvider
com.safenetinc.luna.provider.createExtractableKeys=true
```

27. Set the file permissions for the Luna client as follows:

```
chmod -R 777 /usr/safenet
chown -R kms:kms /usr/safenet
```

Set Ranger KMS configuration properties in Cloudera Manager.

28. Add the .so files to the java.library.path:

a) Go to Cloudera Manager Ranger KMS Configuration .
b) Search for the Ranger KMS Client Java Opts property.
c) Enter the following value and click Save Changes:

```
-Djava.library.path=<existing-native-file-path>:/usr/safenet/lunaclient/
jsp/lib/
```

Note: The <existing-native-file-path> value comes from step 25.

29. In Cloudera Manager Ranger KMS Configuration edit the following properties:

Note: For Cloudera Manager 7.1.1 and Cloudera Manager 7.1.2 you must add properties to the dbks-site.xml, also known as

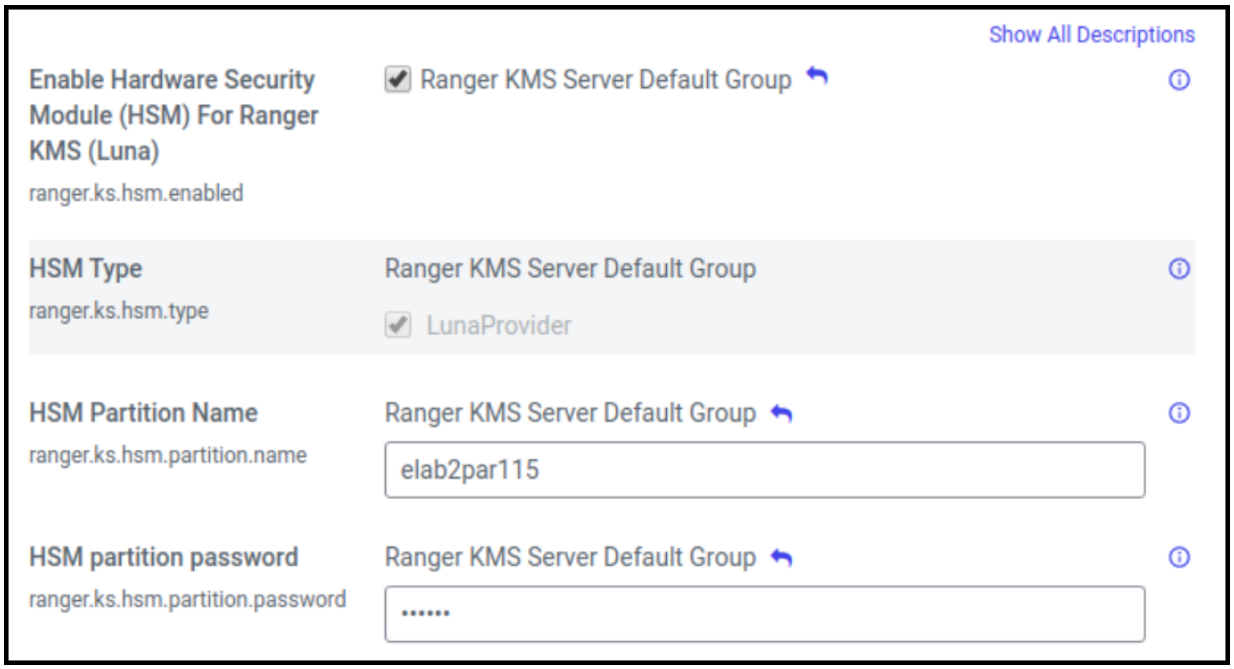
Ranger KMS Server Advanced Configuration Snippet (Safety Valve) for conf/dbks-site.xml .

```
ranger.ks.hsm.type = LunaProvider
ranger.ks.hsm.enabled = true
ranger.ks.hsm.partition.name=elab2par115
ranger.ks.hsm.partition.password=passwrd123
(CM-7.1.1 & CM-7.1.2 password will be in plain text)
```

Note: For Cloudera Manager 7.1.3 and higher, update the configuration as shown in:

Example :

Figure 2: Adding Ranger KMS Configuration for Luna 10.5 HSM



30. Restart Ranger KMS from Cloudera Manager.
31. Login to Luna client and validate whether the master key is successfully created.

```
cd /usr/safenet/lunaclient/bin/
./lunacm
role login -n co

enter password: passwrd123

par con
```

Example :

```
lunacm:>par con
    The 'Crypto Officer' is currently logged in.
    Looking for objects accessible to the 'Crypto Officer'.

    Object List:

    Label:              RangerKMSKey
    Handle:             131
    Object Type:        Symmentric Key
    Object UID:         ba8e00002e00000554380800
    Number of Objects: 1

Command Result: No Error
```

Results

Ranger KMS is successfully started.

What to do next

You can now create Encryption zone keys using hadoop command or from Ranger UI using credentials of keyadmin user.

Integrating Ranger KMS DB with Google Cloud HSM

How to integrate Ranger KMS DB with Google Cloud HSM.

About this task

This task describes how to integrate Ranger KMS DB with Google Cloud Platform (GCP) Hardware Security Module (HSM). This process includes setting up the GCP HSM service on a client (host), configuring Ranger KMS with GCP, or migrating the Master Key storage from the KMS database to the Google Cloud HSM.

Before you begin

- Ensure you can log in to the Google Cloud Console using your accout. (Requires Google account access).
- Ensure you have Java installed.
- You must create a back up of the Ranger KMS DB before executing the migration script to migrate to GCP.

Procedure

Set Up Google Cloud HSM

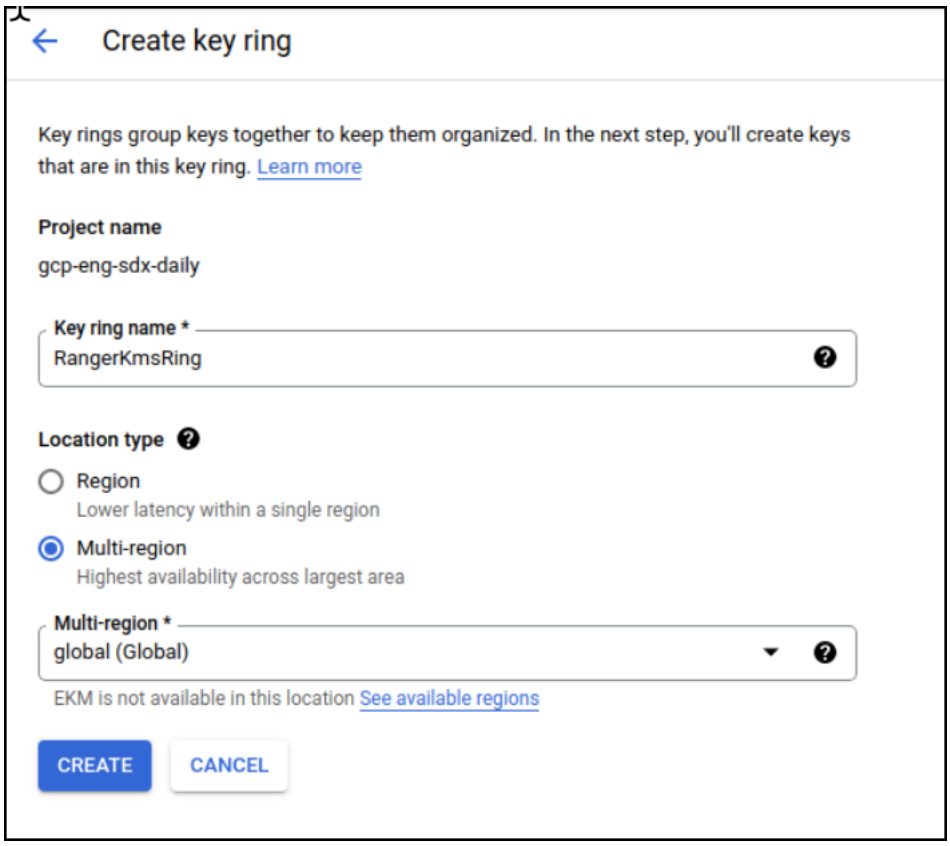
1. Log in to Google Cloud console using Cloudera account.
2. Create the service account by selecting or creating the Project.
3. Create the key.

4. Download and save the key in JSON format.

   Note: Record the project ID, Location ID and save the JSON file.

5. In GCP Console Key Management create the key ring.

Figure 3: Creating a key ring in Google Cloud Platform

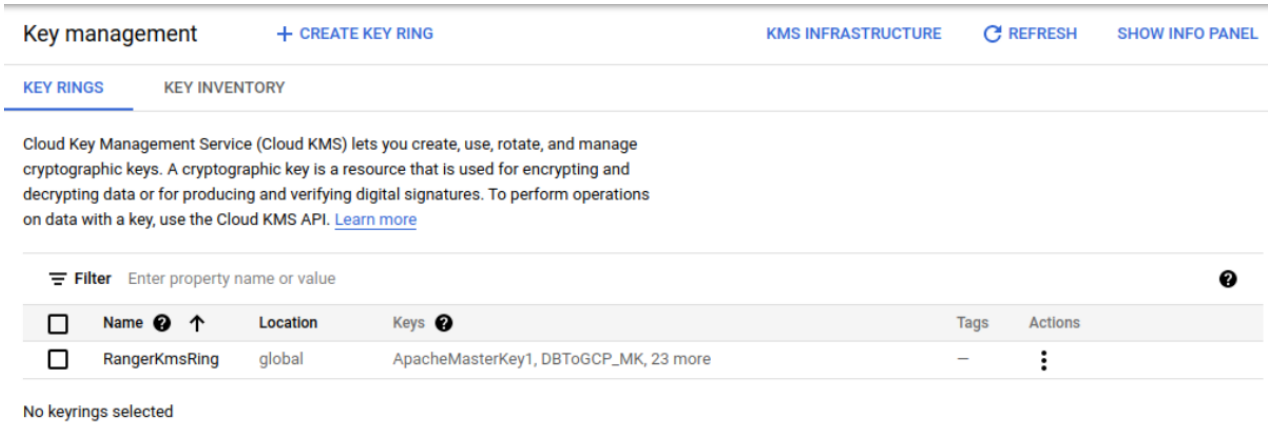


This example shows a project gcp-eng-sdx-daily, region Global, and key ring RangerKMSRing.

Results

The key ring is created.

Figure 4: RangerKMSRing created



Integrating Ranger KMS with Google Cloud HSM

This document provides technical instructions for configuring Apache Ranger Key Management Service (KMS) to use Google Cloud HSM for key management. It covers both a fresh installation scenario and an upgrade/migration scenario for existing installations.

Note: This feature is available since Cloudera Runtime 7.3.2.100 and 7.3.2.10000 releases.

Prerequisites and Google Cloud HSM setup

Before configuring Ranger KMS, you must establish the necessary infrastructure and access tokens within your Google Cloud Platform (GCP) environment:

1. Create a service account:
   a. Log in to the Google Cloud Console.
   b. Select your project and create a dedicated service account.
   c. Download its authentication key in JSON format.

   This file is required for authentication.

   Note: Record your Project ID, Location ID, and the file path of the downloaded credential JSON file.
2. Create a key ring:
   a. In the GCP Console search bar, search for Key Management (Cloud KMS).
   b. Create a new key ring (for example, RangerKmsRing).

Important: Since you are utilizing Cloudera credentials, these account setup and key ring creation steps are already completed.

Fresh installation: Configuring Ranger KMS with GCP

Follow these steps if you are performing a clean configuration of Ranger KMS to natively use GCP for master key generation:

1. Stop the KMS service.

   Ensure that the Ranger KMS service is completely stopped before applying configurations.
2. Configure the Safety Valve:

a. Log in to Cloudera Manager.
b. Navigate to Ranger KMS Configuration and search for dbks-site.xml.
c. Add the following property definitions into the dbks-site.xml safety valve:

```
<property>
  <name>ranger.kms.gcp.enabled</name>
  <value>true</value>
</property>
<property>
  <name>ranger.kms.gcp.keyring.id</name>
  <value>RangerKmsRing</value>
</property>
<property>
  <name>ranger.kms.gcp.cred.file</name>
  <value>/opt/cloudera/parcels/CDH/lib/ranger-kms/your_cred_file.json</
value>
</property>
<property>
  <name>ranger.kms.gcp.project.id</name>
  <value>gcp-eng-sdx-daily</value>
</property>
<property>
  <name>ranger.kms.gcp.location.id</name>
  <value>global</value>
</property>
<property>
  <name>ranger.kms.gcp.masterkey.name</name>
  <value>ApacheMasterKey1</value>
</property>
```

Alternatively, you can copy the downloaded GCP service account credential JSON file to the following directory on your KMS host:

```
/opt/cloudera/parcels/CDH/lib/ranger-kms
```

3. Save the configuration changes and start the KMS service.

   The master key will automatically be created inside your Google Cloud HSM ecosystem.
4. Verify that the standard KMS zone encryption and decryption tasks successfully route through the newly established GCP master key.

Upgrade scenario: Migrating master key storage from KMS database to Google Cloud HSM

Follow these steps to migrate an existing Ranger KMS master key stored in a local relational database to Google Cloud HSM:

1.  **Important:** Take a full backup of your Ranger KMS database before proceeding.
2. Move to your Ranger KMS home directory:

   ```
   /opt/cloudera/parcels/CDH/lib/ranger-kms
   ```
3. SSH to the KMS host and run the following command:

   ```
   ls -ltrh /var/run/cloudera-scm-agent/process/ | grep kms
   ```

   You will get kms process directories.

   a. Identify the latest directory entry from the timestamps provided.

4. Export the following variable (process number and password will vary in the following snippet):

```
export JAVA_HOME=/usr/java/<jdk-version>-cloudera/
export RANGER_KMS_HOME=/opt/cloudera/parcels/CDH/lib/ranger-kms
export RANGER_KMS_CONF=/var/run/cloudera-scm-agent/process/1546338415-ran
ger_kms-RANGER_KMS_SERVER/conf
export SQL_CONNECTOR_JAR=/usr/share/java/postgresql-connector-java.jar
export HADOOP_CREDSTORE_PASSWORD=sbpiuybywid78552ory169p6
```



Important: For HADOOP_CREDSTORE_PASSWORD, you need to read the proc.json file, which is stored at /var/run/cloudera-scm-agent/process/1546338415-ranger_kms-RANGER_KMS_SERVER/proc.json (process number will vary). Search for HADOOP_CREDSTORE_PASSWORD and take its value for export.

5. Migrate the master key to GCP by using the MigrateMKeyStorageDbToGCP.sh utility script.

   Use the following command for this purpose:

```
bash MigrateMKeyStorageDbToGCP.sh <key_name_for_gcp> <gcp_project_name>
<gcp_key_ring_name>  <gcp_key_ring_location> <cred json file>
```

   Upon successful execution, the following message appears:

```
Getting Master KeyMaster Key from Ranger KMS DB has been successfully mi
grated to GCP.
```

6. Now navigate to Cloudera Manager Ranger KMS Configuration , and search for dbks-site.xml.
7. Add the following in the Ranger KMS Server Advanced Configuration Snippet (Safety Valve) for conf/dbks-site.xml and save the changes:

```
<property>
    <name>ranger.kms.gcp.enabled</name>
    <value>true</value>
</property>

<property>
    <name>ranger.kms.gcp.keyring.id</name>
    <value>RangerKmsRing</value>
</property>

<property>
    <name>ranger.kms.gcp.cred.file</name>
    <value>/full/path/to/downloadedCredJsonFile.json</value>
</property>

<property>
    <name>ranger.kms.gcp.project.id</name>
    <value>gcp-eng-sdx-daily</value>
</property>

<property>
    <name>ranger.kms.gcp.location.id</name>
    <value>global</value>
</property>
<property>
   <name>ranger.kms.gcp.masterkey.name</name>
   <value>vbmasterkeyupmar201</value>
</property>
```

   Alternatively, you can copy the cred json file in the following location: /opt/cloudera/parcels/CDH/lib/ranger-kms

8. Add the following in the Ranger KMS Service Environment Advanced Configuration Snippet (Safety Valve) and save the changes:

```
Key: JAVA_OPTS
Value: --add-opens java.base/java.util=ALL-UNNAMED
```

9. Restart the Ranger KMS service, and encrypt and decrypt new zones.

   You should be able to access previously encrypted zones as well.

Integrating Ranger KMS DB with CipherTrust Manager HSM

How to integrate Ranger KMS DB with CipherTrust Manager HSM.

About this task

This task describes how to integrate Ranger KMS DB with CipherTrust Manager Hardware Security Module (HSM). This process includes configuring the NAE port in Thales Cipher Trust Manager, configuring Ranger DB KMS to interact with Thales CipherTrust HSM, or migrating Ranger KMS DB Master Key To CipherTrust Manager HSM, and migrating the master key from CipherTrust Manager HSM to Ranger KMS DB.

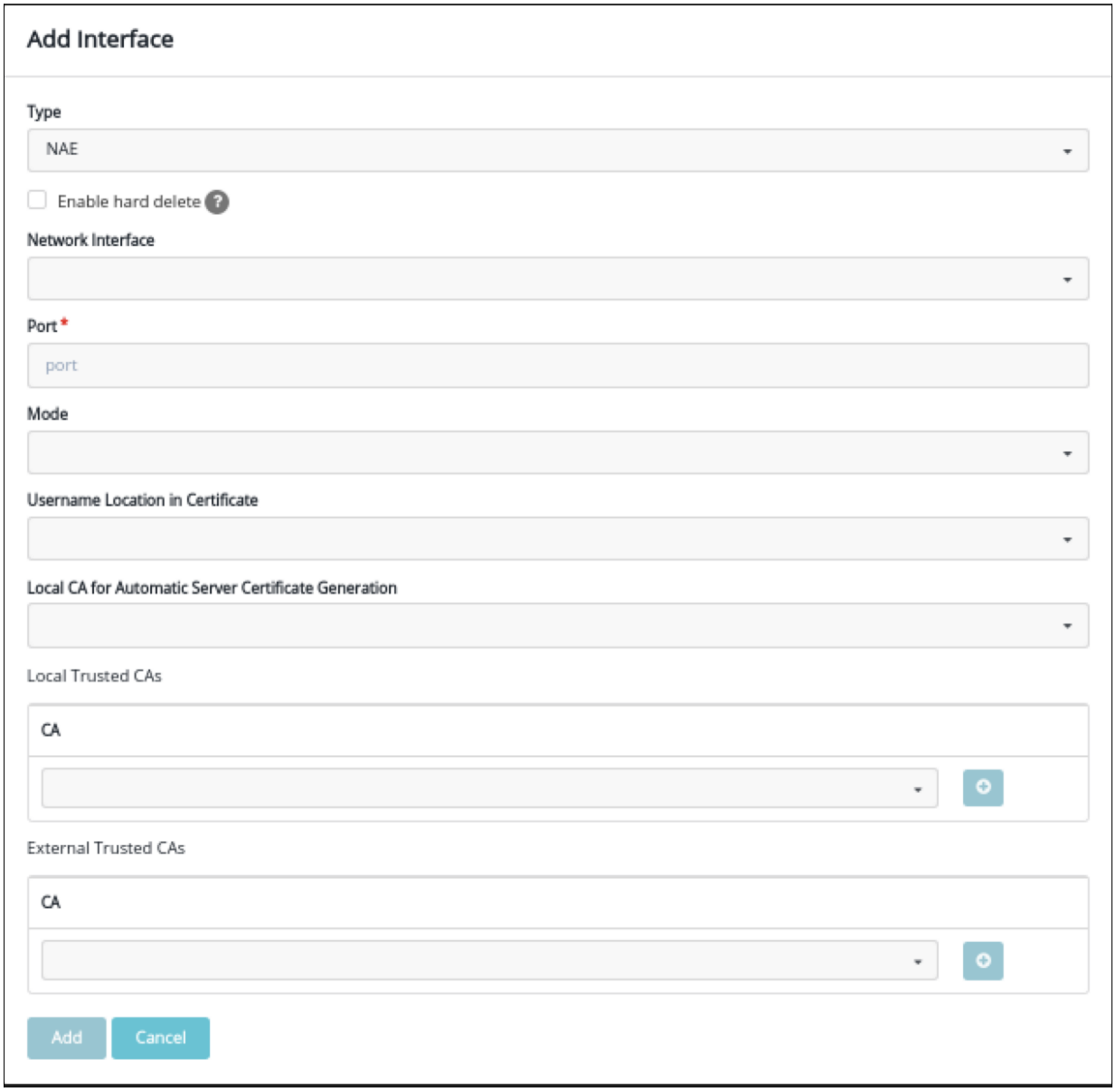
Before you begin

- Ensure you have Thales CipherTrust Manger installed in your enivronment.
- Ensure you have Java installed.

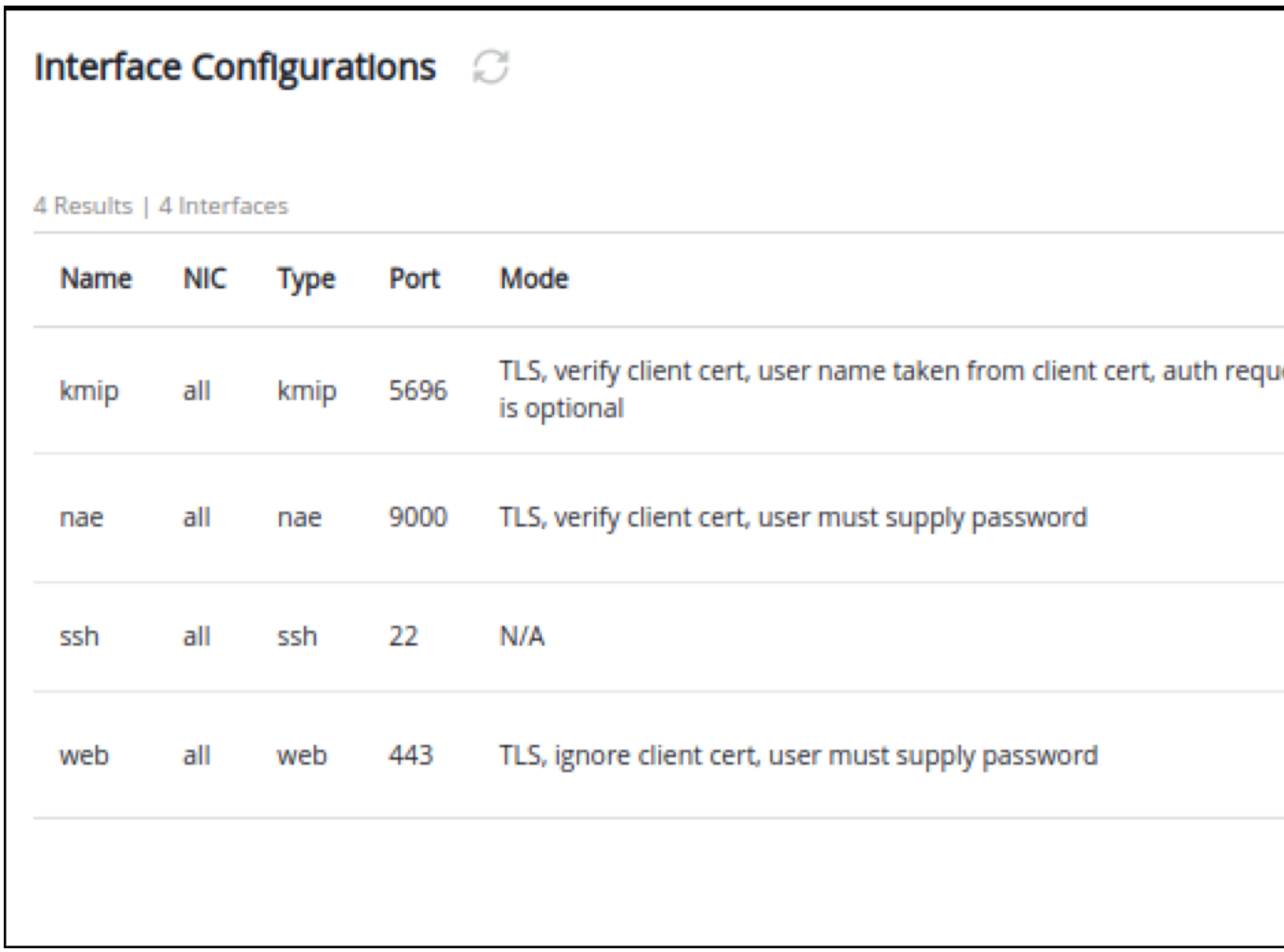
Procedure

Configure NAE port in Thales CipherTrust Manager

1. Log in to Thales CipherTrust Manager.
2. In CipherTrust Manager Admin Settings , select Add Interface.
3. In Type, Select NAE (default).
4. In Network Interface, selectAll.
5. In Port, type a value for the port number.
   9000
6. In Mode, select one of the following options to match your environment:
   - No TLS,user must supply password.
   - TLS, Ignore client cert. user must supply password.
7. Click Add.



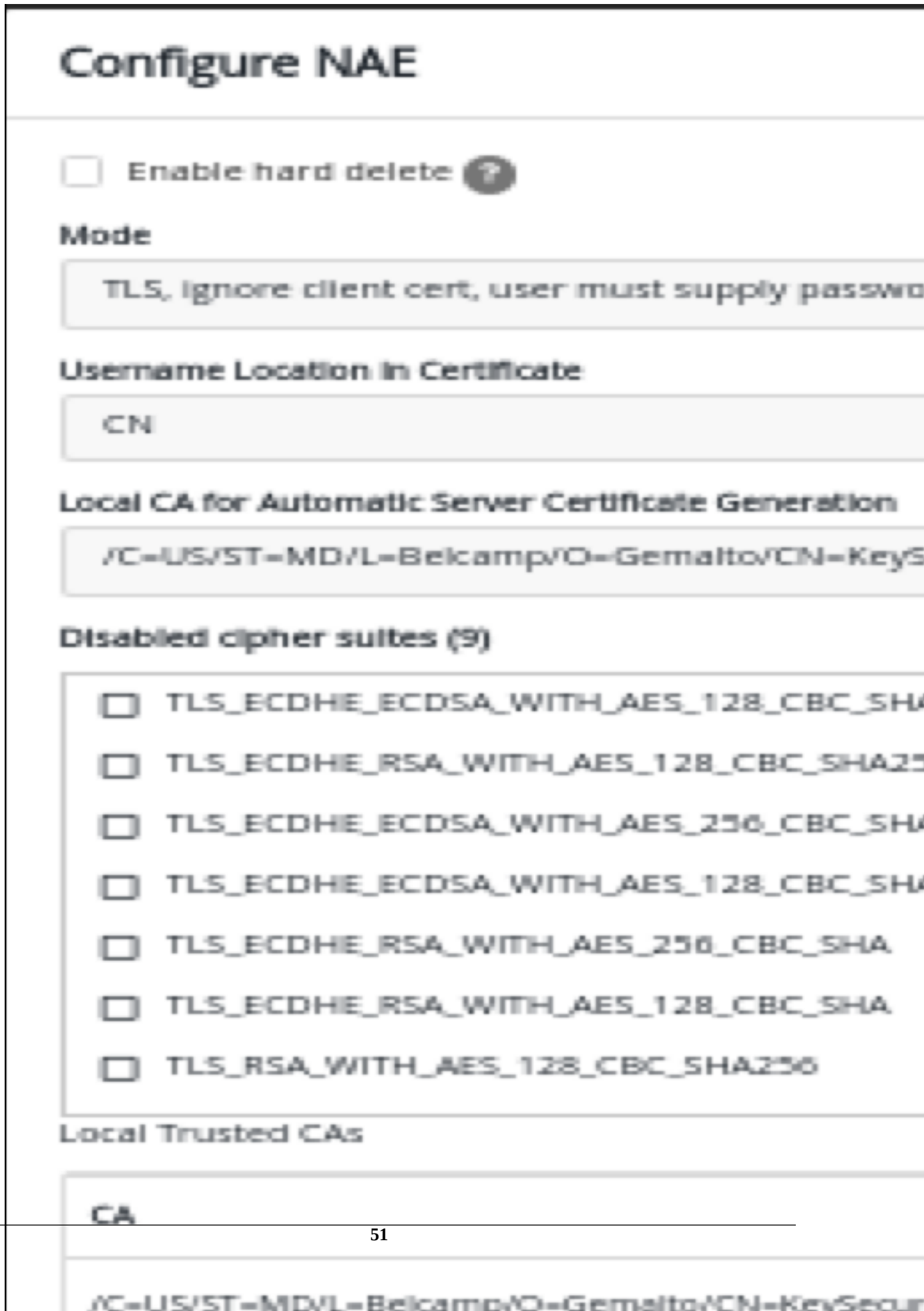
8. If selected mode is TLS, ignore client cert, user must supply password while adding interface, then click on Edit and Download Current Certificate as shown in the images below. Else, skip this step.



Interface Configurations

4 Results | 4 Interfaces

Name	NIC	Type	Port	Mode
kmip	all	kmip	5696	TLS, verify client cert, user name taken from client cert, auth requ is optional
nae	all	nae	9000	TLS, verify client cert, user must supply password
ssh	all	ssh	22	N/A
web	all	web	443	TLS, ignore client cert, user must supply password


Configure NAE
Enable hard delete
Mode
TLS, ignore client cert, user must supply passwo
Username Location in Certificate
CN
Local CA for Automatic Server Certificate Generation
/C=US/ST=MD/L=Belcamp/O=Gemalto/CN=KeyS
Disabled cipher suites (9)
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA25
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
Local Trusted CAs
CA
/C=US/ST=MD/L=Belcamp/O=Gemalto/CN=KeySecur

9. After the certificate is downloaded (for example, Certificate_nae.txt), copy it to Ranger KMS server Create a directory on Ranger KMS serverhost under /etc/security.

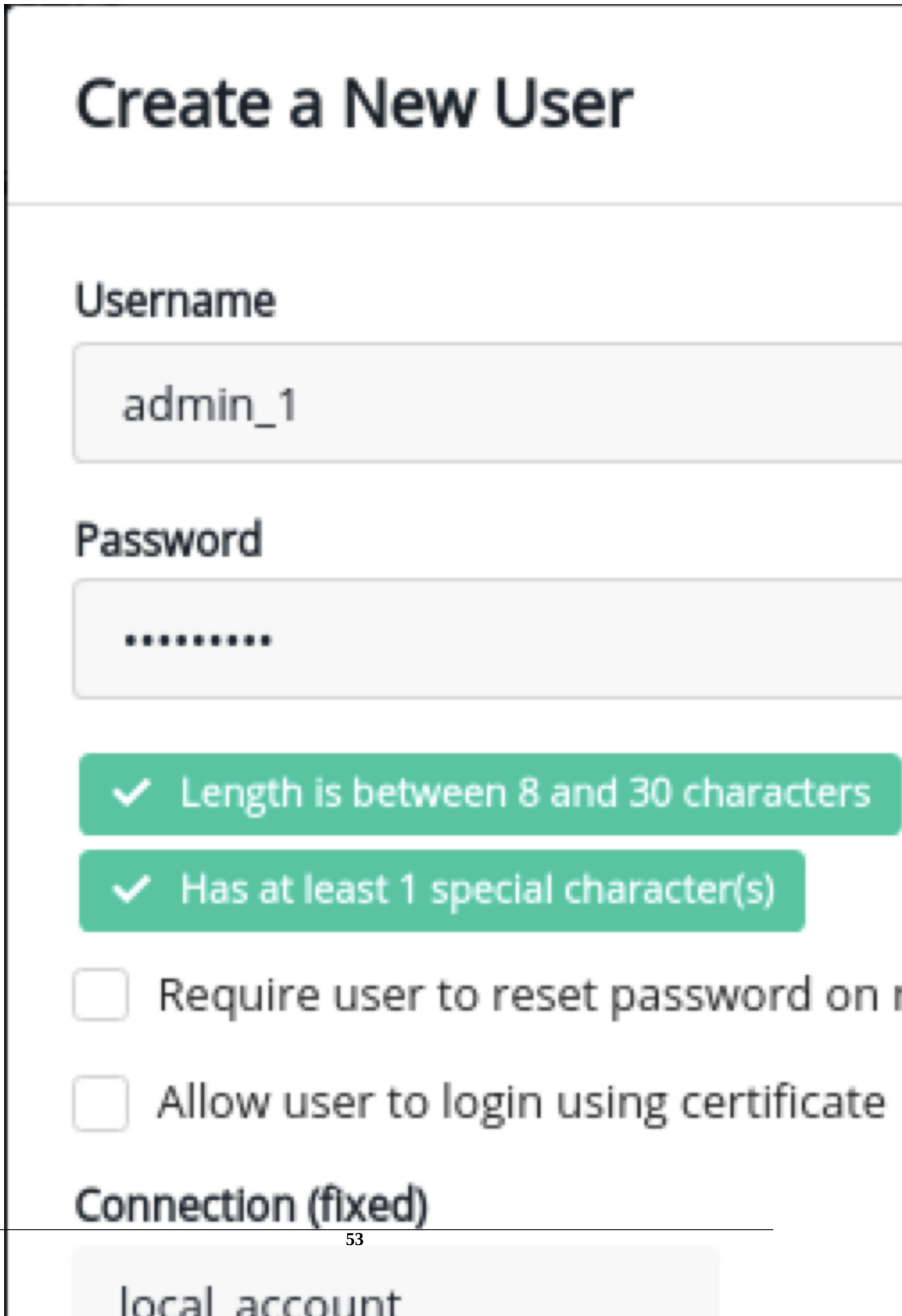
```
mkdir etc/security/serverKeys
```

and scp the downloaded certificate to this directory. Ensure that the user has required access to the file:

```
chown kms:kms etc/security/serverKeys/Certificate_nae.txt
```

```
chmod 755 etc/security/serverKeysCertificate_nae.txt
```

10. Create a user.
    a) Go to Access Management Users , click Create New User .
    b) In Create a New User, provide a username, password, and any required information.
    c) Click Create.


Create a New User
Username
admin_1
Password
Length is between 8 and 30 characters
Has at least 1 special character(s)
Require user to reset password on
Allow user to login using certificate
Connection (fixed)
local_account

Fresh installation - Configuring Ranger KMS DB to interact with Thales CipherTrust HSM

These are the steps required to configure Ranger KMS DB to interact with Thales CipherTrust HSM. These steps need to be performed only when the cluster is ready with all the required services and no encryption zone keys are present.

Procedure

1. SSH into the Ranger KMS host.
2. Create a directory and copy the files IngrianNAE.properties, libIngPKCS11.so and sunpkcs11.cfg from Thales CipherTrust Manager to the directory.

   ```
   mkdir -p /opt/safenetConf/64/8.3.1
   ```

   and then copy the above mentioned files under this location.

   Note: Ensure that you have provided read-write access (chmod 775) to Ranger KMS service users (chown kms:kms) to all the 3 files.

3. Update the IngrianNAE.properties file and initialize the following properties.

   ```
   NAE_IP = Hostname / CipherTrust Manager IP address
   ```

   ```
   NAE_Port=9000 (should be the same port provided on CipherTrust instance
   under Device tab)
   ```

   ```
   Protocol=tcp (valid values ssl or tcp, should be the same protocol provi
   ded on CipherTrust instance)
   ```

4. Initialize the following property only when the protocol of the CipherTrust Manager instance is SSL. This is the location of the certificate downloaded from CipherTrust Manager and copied on Ranger KMS host.

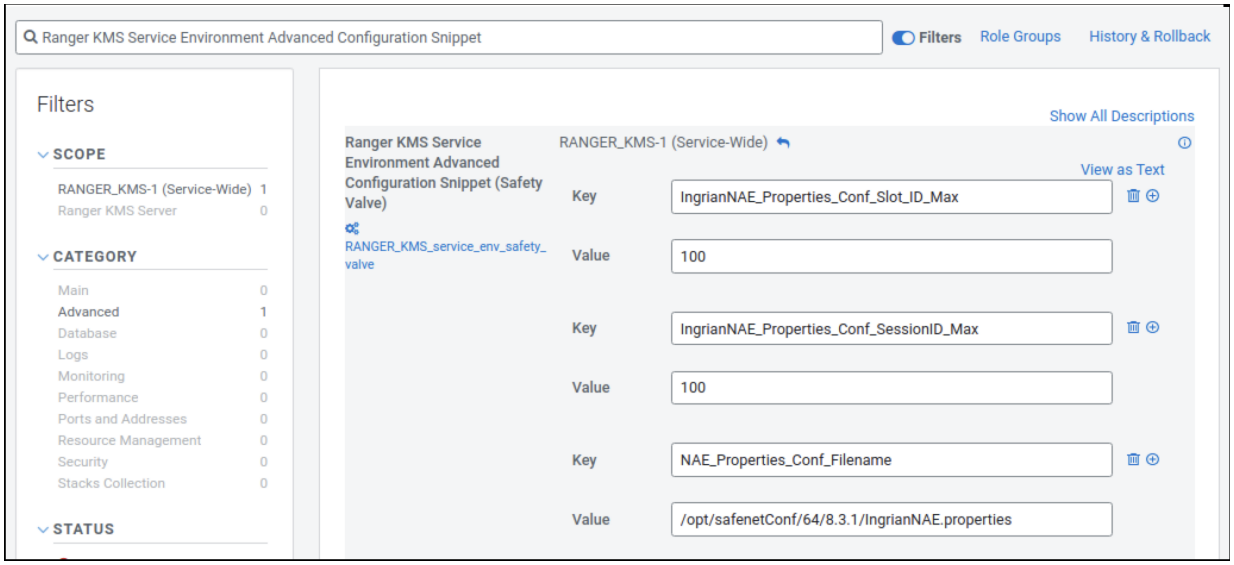
   ```
   CA_File=/etc/security/serverKeys/Certificate_nae.txt
   ```

5. Go to Cloudera Manager Ranger KMS Configuration Search for Ranger KMS Server Advanced Configuration Snippet and click on + icon to add the properties.
6. Add the following properties:

   ```
   IngrianNAE_Properties_Conf_Slot_ID_Max=100
   ```

   ```
   IngrianNAE_Properties_Conf_SessionID_Max=100
   ```

   ```
   NAE_Properties_Conf_Filename=/opt/safenetConf/64/8.3.1/IngrianNAE.proper
   ties
   ```



7. Save changes.
8. Search for the following properties and set the values as follows:

```
ranger.kms.keysecure.enabled = true

ranger.kms.keysecure.UserPassword.Authentication = true

ranger.kms.keysecure.masterkey.name = MasterKey1

ranger.kms.keysecure.login.username = keysecure-username

ranger.kms.keysecure.login.password = keysecure-password

ranger.kms.keysecure.masterkey.size = 256

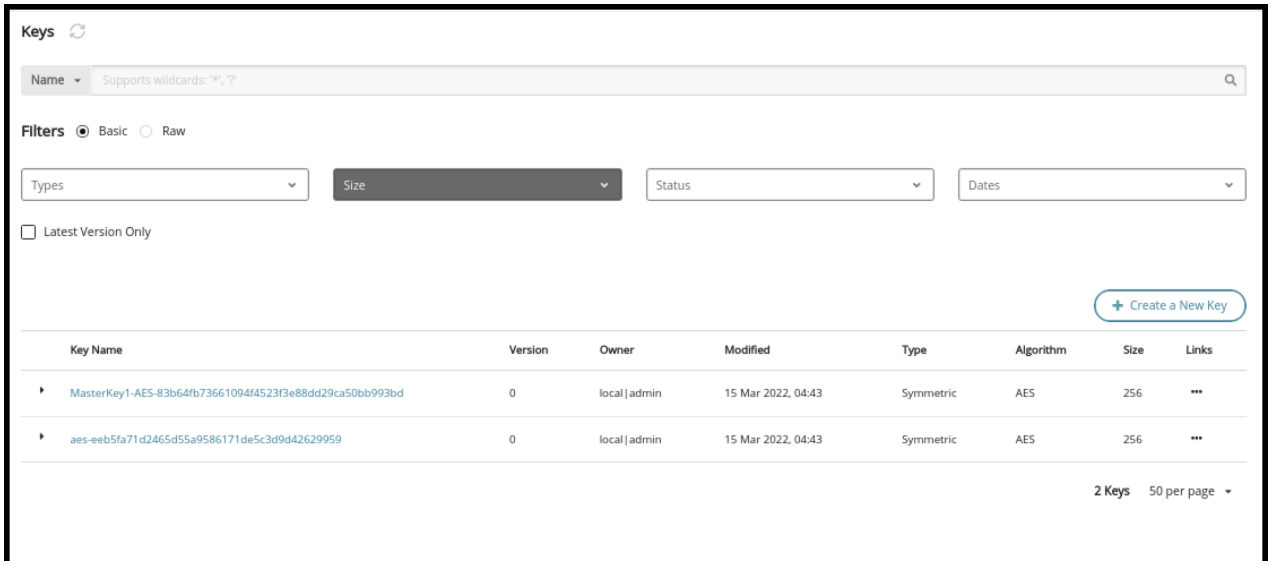
ranger.kms.keysecure.sunpkcs11.cfg.filepath = /opt/safenetConf/64/8.3.1/
sunpkcs11.cfg
```

9. Save changes.
10. Restart Ranger KMS.

Results

The master key with alias MasterKey1 is created in CipherTrust Manager.

Example



What to do next

Once Ranger KMS has started successfully, verify zone key creation, and zone encryption/decryption.

Migrating Ranger KMS DB Master Key To CipherTrust Manager HSM

How to migrate the Ranger KMS DB Master Key to CipherTrust Manager HSM. These steps need to be performed when the cluster is ready with all the required services and encryption zone keys are present in the Ranger KMS DB.

Before you begin

Ensure that you have configured CipherTrust Manager.

Procedure

1. Stop Ranger KMS service.
2. SSH into the Ranger KMS host.
3. Create a directory and copy the files IngrianNAE.properties, libIngPKCS11.so and sunpkcs11.cfg from Thales CipherTrust Manager to the directory

   ```
   mkdir -p /opt/safenetConf/64/8.3.1
   ```

   and then copy the above mentioned files under this location.

   Note: Ensure that you have provided read-write access (chmod 775) to Ranger KMS service users (chown kms:kms) to all the 3 files.

4. Update the IngrianNAE.properties file and initialize the following properties.

   ```
   NAE_IP = Hostname / CipherTrust Manager IP address
   ```

   ```
   NAE_Port=9000 (should be the same port provided on CipherTrust instance
   under Device tab)
   ```

   ```
   Protocol=tcp (valid values ssl or tcp, should be the same protocol provi
   ded on CipherTrust instance)
   ```

5. Initialize the following property only when the protocol of the CipherTrust Manager instance is SSL. This is the location of the certificate downloaded from CipherTrust Manager and copied on the Ranger KMS host.

```
CA_File=/etc/security/serverKeys/Certificate_nae.txt
```

6. Go to the Ranger KMS home directory.

```
cd /opt/cloudera/parcels/CDH/lib/ranger-kms
```

7. Export the following variables and ensure the variables are exported successfully using the env|grep NAE command.

```
export IngrianNAE_Properties_Conf_Slot_ID_Max=100
export IngrianNAE_Properties_Conf_SessionID_Max=100
export NAE_Properties_Conf_Filename=/opt/safenetConf/64/8.3.1/IngrianNAE.p
roperties
```

8. Go to Ranger KMS home directory and export the environment variables.

```
export JAVA_HOME = /usr/java/<jdk-version>-cloudera
export RANGER_KMS_HOME = /opt/cloudera/parcels/CDH/lib/ranger-kms
export RANGER_KMS_CONF = /var/run/cloudera-scm-agent/process/current_procs
ess_dir-RANGER_KMS_SERVER/conf
export SQL_CONNECTOR_JAR = /path/to/SQL-Connector.jar
export HADOOP_CREDSTORE_PASSWORD = hadoop_cred_store_password
```

9. Run the DBMKTOKEYSECURE.sh script.

```
./DBMKTOKEYSECURE.sh masterKey-name keysecure-UserName keysecure-passwor
d /path/to/sunpkcs/config/file/sunpkcs11.cfg
```

Master Key from Ranger KMS DB has been successfully imported into CipherTrust Manager.

10. Go to Cloudera Manager Ranger KMS Configuration , search for the following properties, and set the values.

```
ranger.kms.keysecure.enabled = true
ranger.kms.keysecure.UserPassword.Authentication = true
ranger.kms.keysecure.masterkey.name = masterkeyname (must be same name u
sed above with migration utility ./DBMKTOKEYSECURE.sh )
ranger.kms.keysecure.login.username = keysecure-username
ranger.kms.keysecure.login.password = keysecure-password  (User details
are craeted on KeySecure while adding the user)
ranger.kms.keysecure.masterkey.size = 256
ranger.kms.keysecure.sunpkcs11.cfg.filepath = /opt/safenetConf/64/8.3.1/su
npkcs11.cfg
```

11. Search for Ranger KMS Service Environment Advanced Configuration Snippet , click on the + icon, and add the following environment variables.

```
IngrianNAE_Properties_Conf_Slot_ID_Max=100

IngrianNAE_Properties_Conf_SessionID_Max=100

NAE_Properties_Conf_Filename=/opt/safenetConf/64/8.3.1/IngrianNAE.properti
es
```

12. Save changes.
13. Restart Ranger KMS.

Results
Master Key from Ranger KMS DB has been successfully migrated to CipherTrust Manager.

What to do next
Once Ranger KMS has started successfully, verify zone key creation, and zone encryption/decryption.

Migrating the Master key from CipherTrust Manger HSM to Ranger KMS DB
How to migrate the Master key from CipherTrust HSM to Ranger KMS DB. These steps need to be performed when the cluster is ready with all the required services and encryption zone keys are present in the Ranger KMS DB.

Before you begin
Ensure that Ranger KMS service is running.

Procedure
1. Find the KMS current process directory.

```
ps -ef | grep proc_rangerkms
```

2. Stop Ranger KMS service from Cloudera Manager.
3. Go to the Ranger KMS home directory and export the following environment variables.

```
export JAVA_HOME = /usr/java/<jdk-version>-cloudera
export RANGER_KMS_HOME = /opt/cloudera/parcels/CDH/lib/ranger-kms
export RANGER_KMS_CONF = /var/run/cloudera-scm-agent/process/current_procs
ess_dir-RANGER_KMS_SERVER/conf
export SQL_CONNECTOR_JAR = /path/to/SQL-Connector.jar
export HADOOP_CREDSTORE_PASSWORD = hadoop_cred_store_password

export IngrianNAE_Properties_Conf_Slot_ID_Max = 100
export IngrianNAE_Properties_Conf_SessionID_Max = 100
export NAE_Properties_Conf_Filename = /path/to/ingrian_props_file/IngrianN
AE.properties
```

4. If the Ranger KMS DB table contains any old master key, then delete the entry from ranger_masterkey table manually and proceed to the next step.
5. Run the ./KEYSECUREMKTOKMSDB script by passing the master key password.

```
./KEYSECUREMKTOKMSDB masterKeyPassword
```

6. Once the script runs successfully, go to Cloudera Manager Ranger KMS Configuration , search for dbks-site.xml, and update the value of ranger.kms.keysecure.enabled to false.
7. Search for ranger.db.encrypt.key.password and set its value to the master-key password which you used above while invoking the migration script.
8. Restart Ranger KMS.

What to do next
Once Ranger KMS has started successfully, verify zone key creation, and zone encryption/decryption.

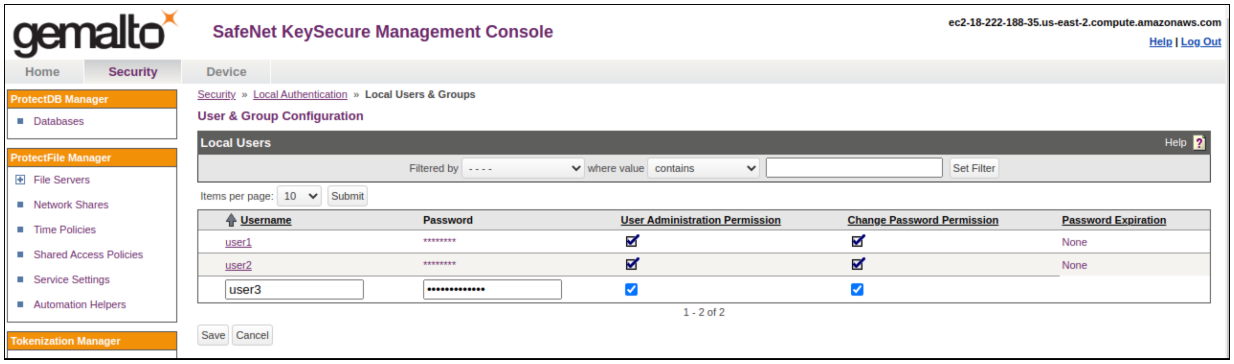
Integrating Ranger KMS DB with SafeNet Keysecure HSM
How to integrate Ranger KMS DB with SafeNet Keysecure HSM.

About this task

This task describes how to integrate Ranger KMS DB with Safenet Keysecure Hardware Security Module (HSM). This process includes setting up the SafeNet KeySecure Management Console, and configuring Ranger KMS to communicate with the Keysecure instance.

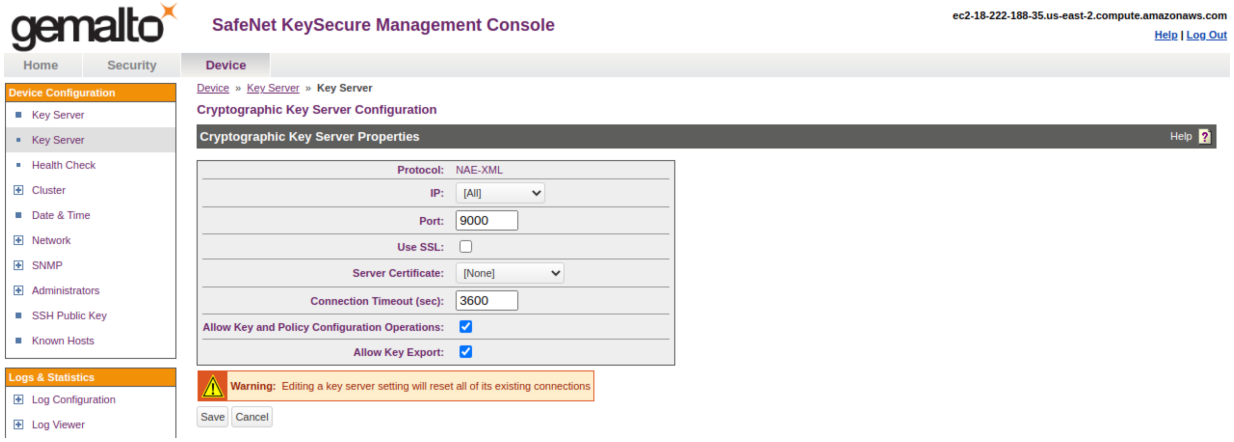
Creating the user on SafeNet keysecure

1. Log in to keysecure as an user with admin privileges.
2. Go to the Security tab.
3. Go to the Users & Groups section.
4. Click Local Authentication, and click Add to add a new user.
5. Check both User Administration Permission and Change Password Permission when adding the new user.
6. Save changes.



Creating device on SafeNet KeySecure

1. Log in to Keysecure with user having admin privilges.
2. Go to Device NAE-XML protocol.
3. Click Properties Edit .
4. Select Allow Key and Policy Configuration Operations and Allow Key Export.

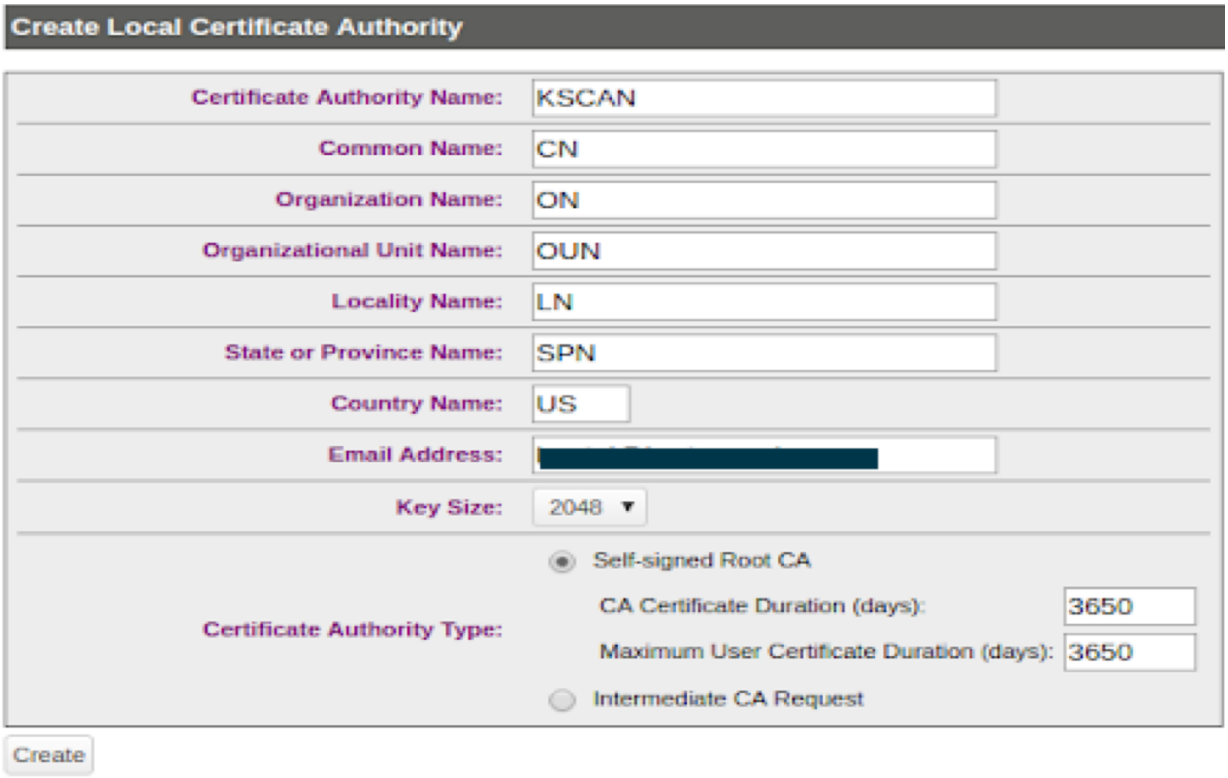


5. Save changes.

Configure SSL on Safenet Keysecure (NAE-XML)

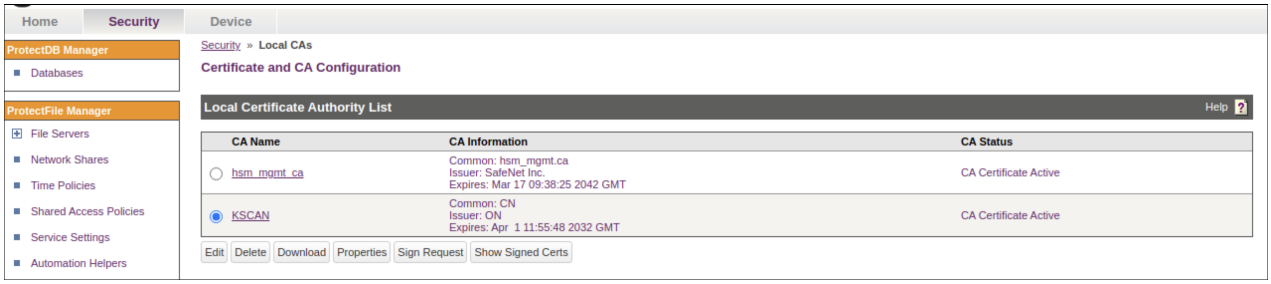
Creating a local CA

1. Log in to the Management Console as an administrator with Certificate Authority (CA) access control.
2. Navigate to the Security, CAs & SSL Certificates section and click Local CA's.
3. Enter the required details and select Self-signed Root CA as the Certificate Authority Type.



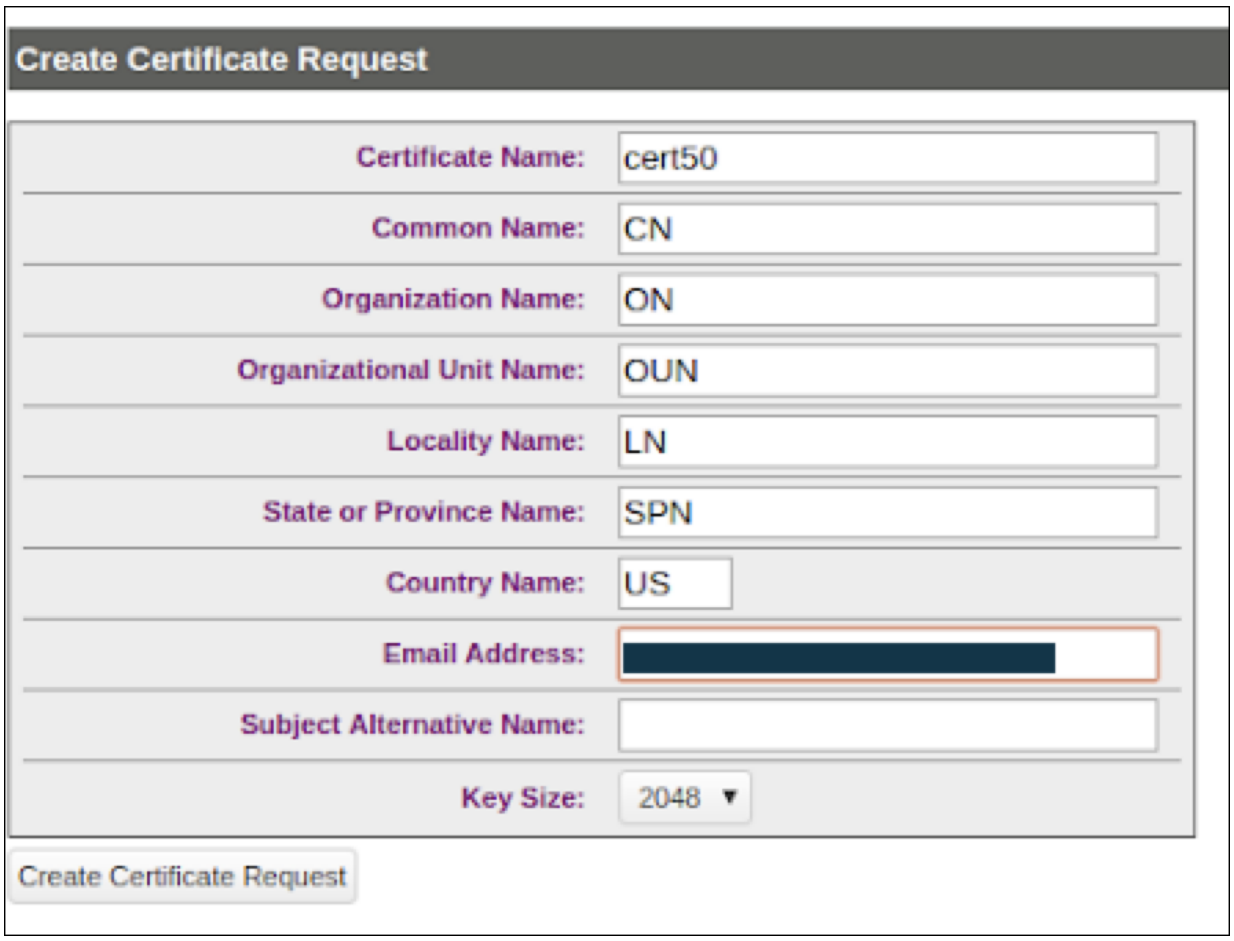
4. Click Create.

The Local CA is visble.



Creating a Server Certificate Request on the Management Console

1. Log on to the Management Console as an administrator with Certificate Authority (CA) access control.
2. Go to the Security tab and on the left side panel.
3. Navigate to the Device CAs & SSL Certificates section.
4. Click SSL certificates and modify the fields as needed.



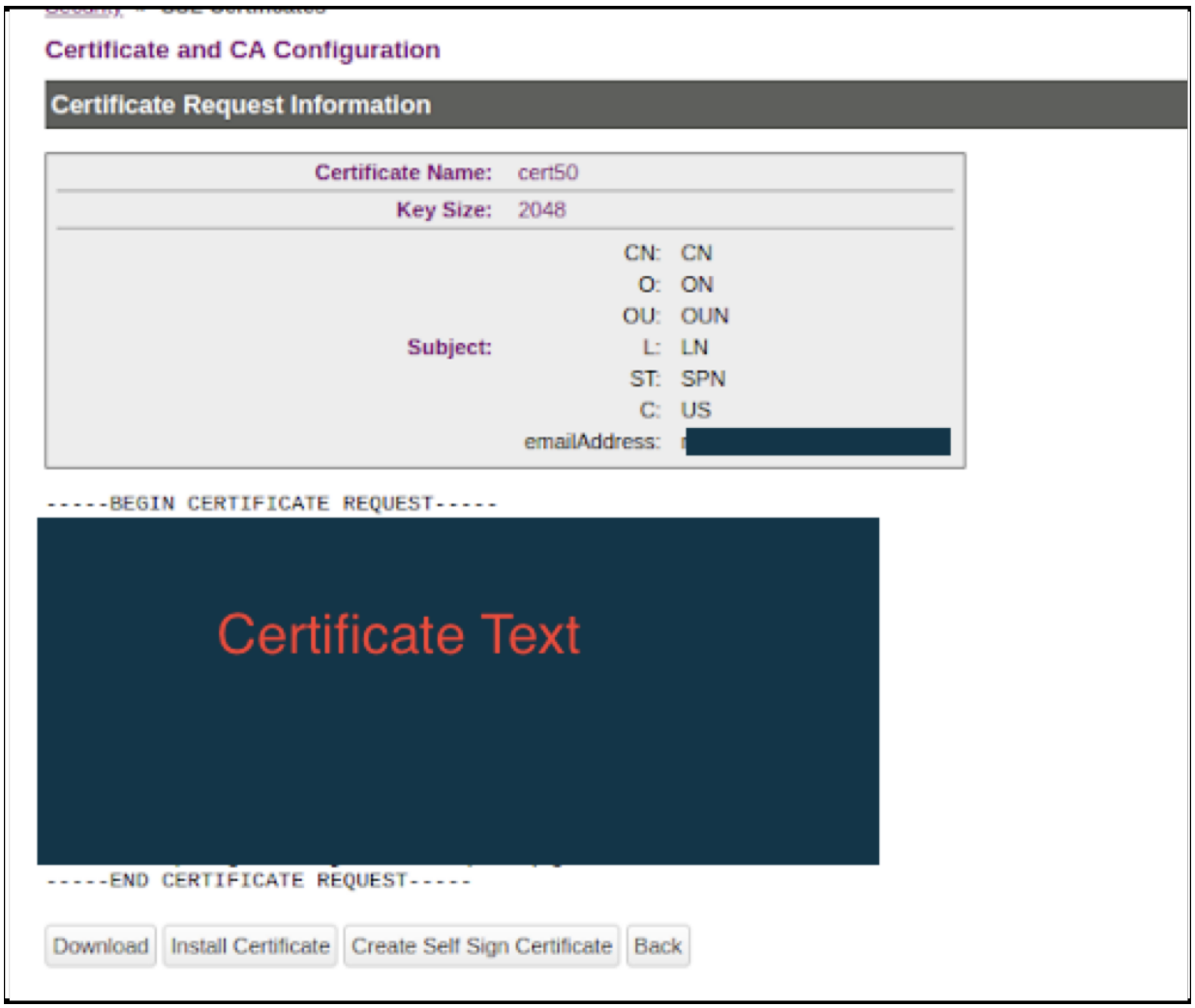
5. Click Create Certificate Request.

   This creates the certificate request and places it in the Certificate List section of the Certificate and CA Configuration page. The new entry shows that the Certificate Purpose is Certificate Request and that the Certificate Status is Request Pending.

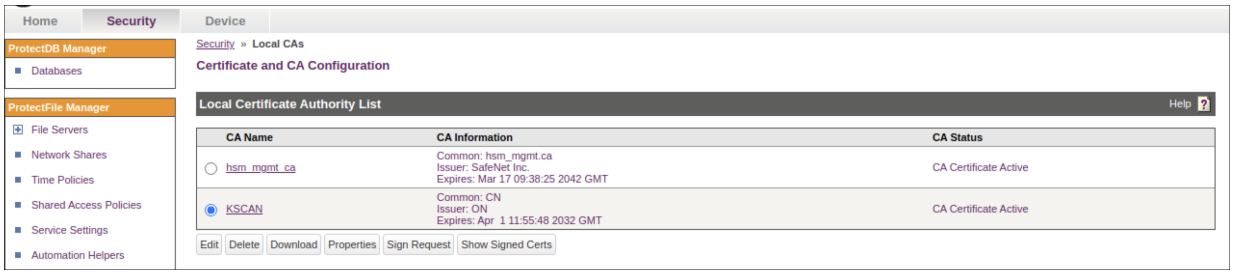


Signing a Server Certificate Request with a Local CA

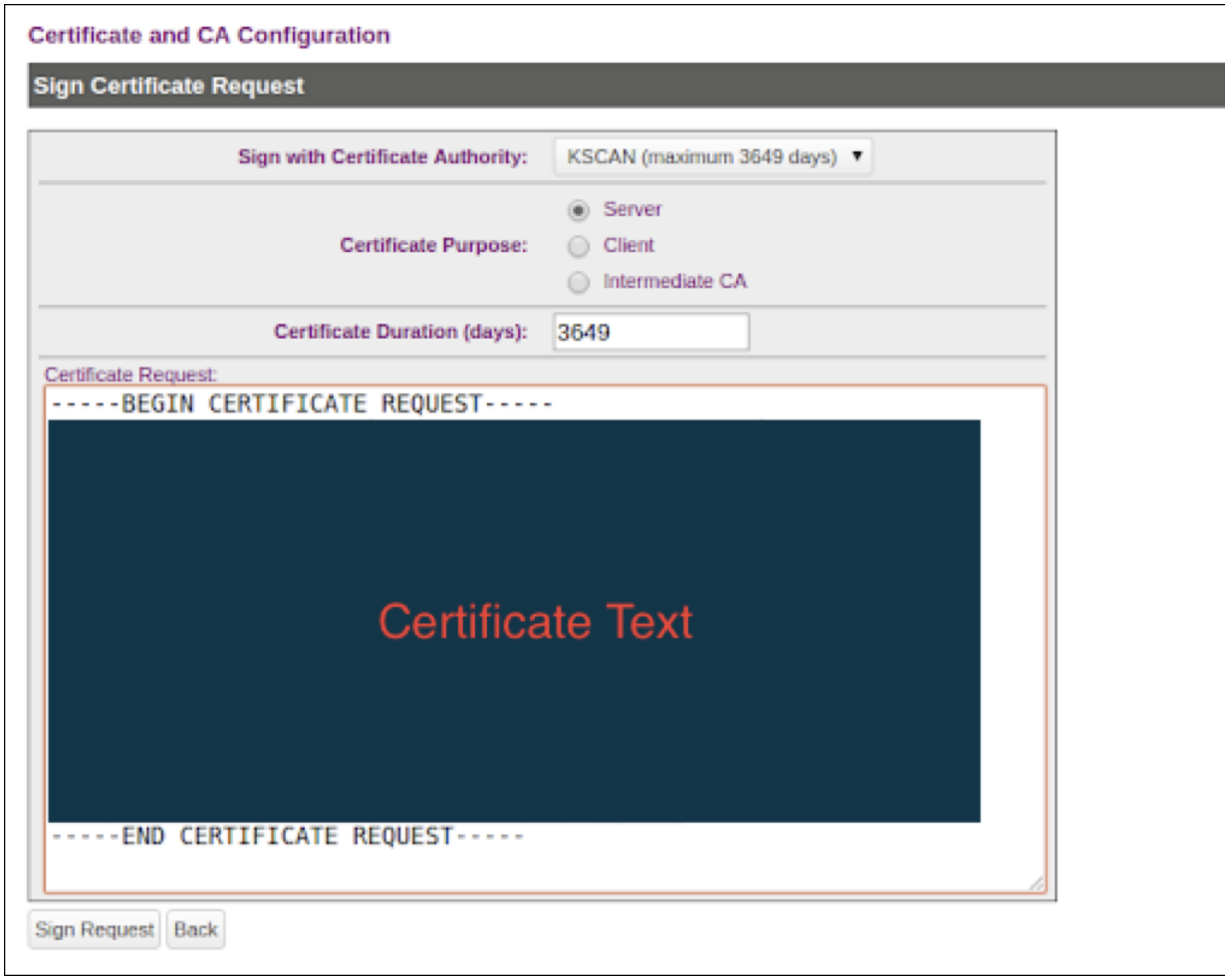
1. Log on to the Management Console as an Administrator with Certificates and Certificate Authorities (CA) access controls.
2. Navigate to the Security Tab -> Device, CAs and SSL Certificates section.
3. Click SSL Certificates.
4. Select the certificate request (cert50) and click Properties.



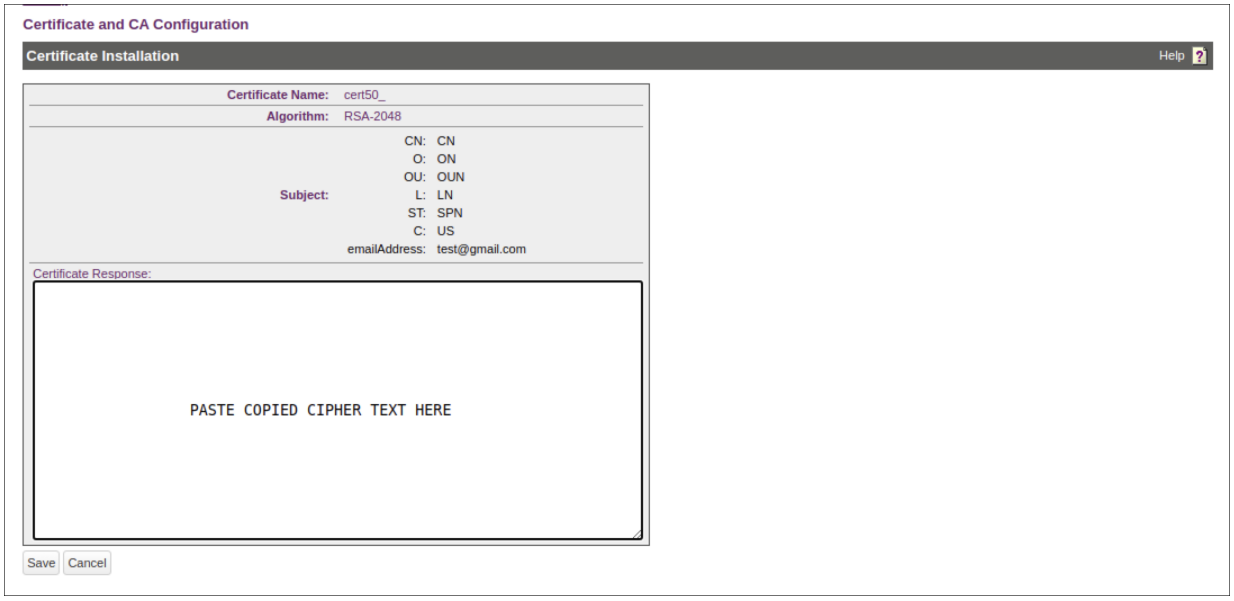
5. Copy the text of the certificate request. The copied text must include the header (-----BEGIN CERTIFICATE REQUEST-----) and footer (-----END CERTIFICATE REQUEST-----).
6. Navigate to the Security Tab -> Device, CAs & SSL Certificates section.
7. Click Local CAs and select the CA name from the list.
8. Click Sign Request to access the Sign Certificate Request section.



9. On the Sign Certificate Request screen, select Server as certificate Purpose.
10. Enter the validity of the certificate for Certificate Duration (days).
11. Paste the copied text from the server certificate request, including the header and footer in Certificate Request.

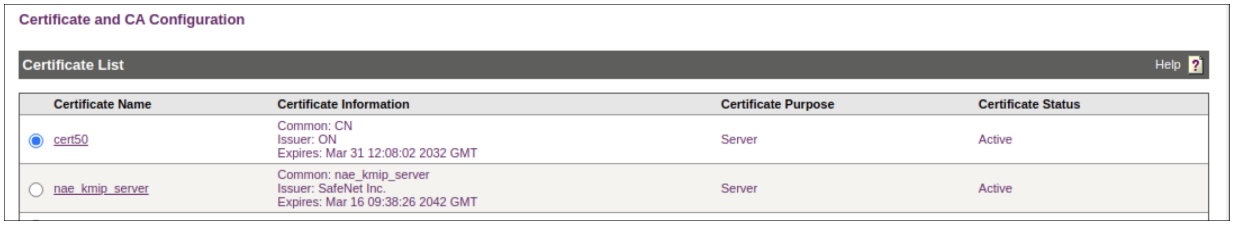


12. Click Sign Request. This takes you to the CA Certificate Information section.
13. Copy the actual (for eaxample, KSCAN) certificate text. The copied text must include the header (-----BEGIN CERTIFICATE-----) and footer (-----END CERTIFICATE-----).
14. Navigate back to the Certificate List section (Device, CAs & SSL Certificates) and click SSL Certificates.
15. Select your certificate request and click properties.
16. Click Install Certificate.
17. Paste the certificate as the Certificate Response.



18. Click Save.

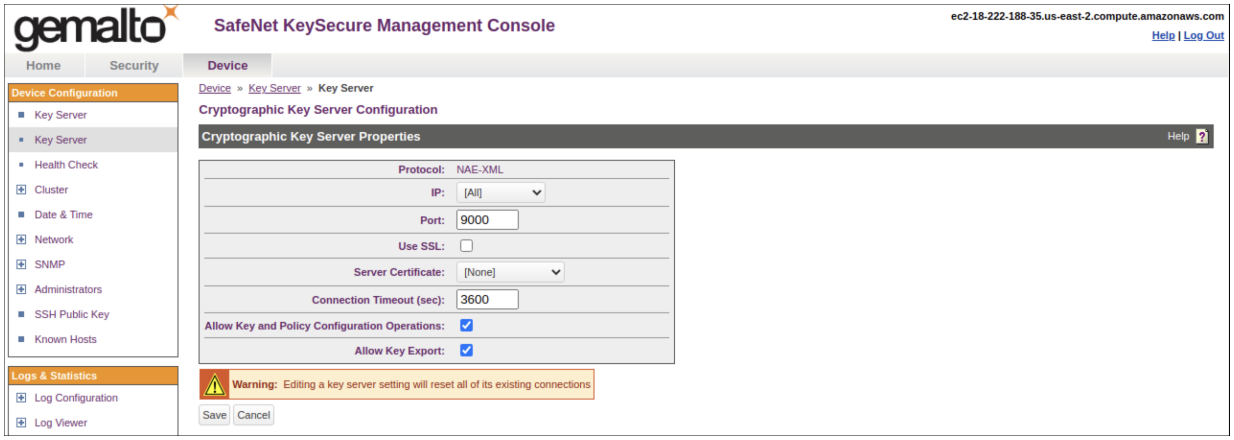
The Management Console takes you to the Certificate List section. The section shows that the Certificate Purpose is Server and that the Certificate Status is Active.



Enable SSL on Keysecure (NAE-XML)

After SSL has been configured in Safenet KeySecure, perform the following steps.

1. Log in to keysecure with admin privileges.
2. Go to the Device tab and click NAE-XML -> properties -> edit.



3. Select Use SSL.
4. Select the Server Certificate from the given drop-down list (for example, cert50).
5. Save changes.

Migrating the Master Key from Ranger KMS DB to Luna HSM

How to migrate the master key from Ranger KMS DB to Luna HSM.

Procedure

1. Go to the Ranger KMS directory.
   Example:

   ```
   cd /opt/cloudera/parcels/CDH/lib/ranger-kms
   ```

2. Export the below variables

   ```
   export JAVA_HOME=/usr/java/<jdk-version>-cloudera
   ```

   ```
   export RANGER_KMS_HOME=/opt/cloudera/parcels/CDH/lib/ranger-kms
   ```

3. Get the active directiory for rangerkms process and copy the conf directory

   ```
   ps -ef | grep rangerkms
   ```

   From the output of the above command, get the value of the rangerkms conf directory.

   ```
   export RANGER_KMS_CONF=/var/run/cloudera-scm-agent/process/xxxx-ranger_k
   ms-RANGER_KMS_SERVER/conf
   export SQL_CONNECTOR_JAR=/opt/cloudera/cm/lib/postgresql-42.1.4.jre7.jar
   ```

4. Get the active directory for rangerkms process and copy the active directory path.

   ```
   ps -ef | grep rangerkms
   ```

5. Open proc.json and get the value for HADOOP_CREDSTORE_PASSWORD

   ```
   vim /var/run/cloudera-scm-agent/process/xxxx-ranger_kms-RANGER_KMS_SERVER/
   proc.json
   export HADOOP_CREDSTORE_PASSWORD=hadoop_credstore_pwd
   ```

6. Run the following command:

   ```
   ${JAVA_HOME}/bin/java -Djava.library.path=/usr/safenet/lunaclient/jsp/lib/
    --add-opens java.xml.crypto/com.sun.org.apache.xml.internal.security.util
   s=ALL-UNNAMED -cp "${RANGER_KMS_HOME}/cred/lib/*:${RANGER_KMS_CONF}:${RA
   NGER_KMS_HOME}/ews/webapp/WEB-INF/classes/lib/*:${SQL_CONNECTOR_JAR}:${R
   ANGER_KMS_HOME}/ews/webapp/config:${RANGER_KMS_HOME}/ews/lib/*:${RANGER_
   KMS_HOME}/ews/webapp/lib/*:${RANGER_KMS_HOME}/ews/webapp/META-INF:${RANG
   ER_KMS_CONF}/*" org.apache.hadoop.crypto.key.DB2HSMMKUtil LunaProvider <
   partition-name>
   ```

7. Enter the partition password.
8. Login to the Luna client and validate if the master key is successfully migrated.

   ```
   cd /usr/safenet/lunaclient/bin/
   ./lunacm
   role login -n co
            enter password: passwrd123
   par con
   ```

If Master Key RangerKMSKey exists, then the following will be visible:

```
lunacm:>par con

                        The 'Crypto Officer' is currently logged in.
                        Looking for objects accessible to the 'Crypto Off
icer'.

                        Object List:

                        Label:              RangerKMSKey
                        Handle:             131
                        Object Type:        Symmentric Key
                        Object UID:         ba8e00002e00000554380800

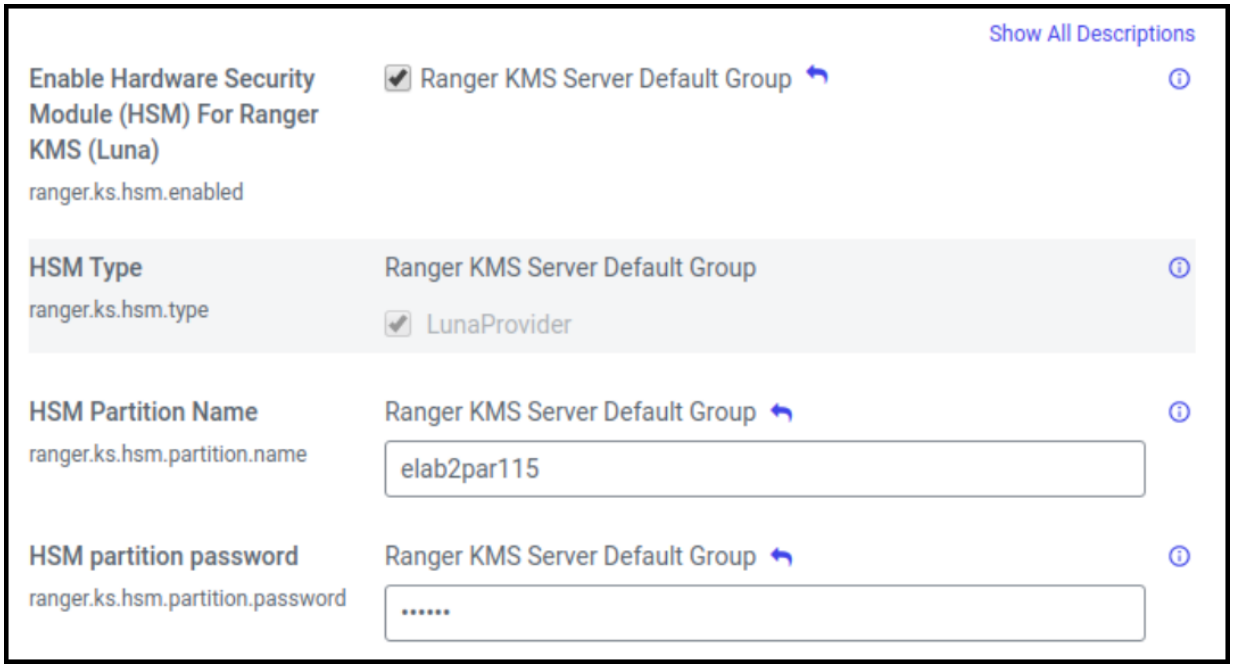
                        Number of Objects: 1

                        Command Result: No Error
```

9. In Cloudera Manager Ranger KMS Configs edit the following properties:

```
ranger.ks.hsm.type = LunaProvider
                        ranger.ks.hsm.enabled = true
                        ranger.ks.hsm.partition.name=<partition-name>
                        ranger.ks.hsm.partition.password=<passwrd123>
```

Figure 5: Adding Ranger KMS Configuration for Luna HSM



10. Restart Ranger KMS from Cloudera Manager.

What to do next

Ensure Ranger KMS is running with HSM enabled. If you do not require, delete the master key row from the database table “ranger_masterkey”, as the master key has already been migrated to the HSM.

Migrating the Master Key from HSM to Ranger KMS DB

How to migrate the master key from Luna HSM to Ranger KMS DB.

Procedure

1. Go to the Ranger KMS directory.
   Example:

   ```
   cd /opt/cloudera/parcels/CDH/lib/ranger-kms
   ```

2. Export the below variables:

   ```
   export JAVA_HOME=/usr/java/<jdk-version>-cloudera
   ```

   ```
   export RANGER_KMS_HOME=/opt/cloudera/parcels/CDH/lib/ranger-kms
   ```

3. Get the active directiory for rangerkms process and copy the conf directory:

   ```
   ps -ef | grep rangerkms
   ```

   From the output of the above command, get the value of the rangerkms conf directory.

   ```
   export RANGER_KMS_CONF=/var/run/cloudera-scm-agent/process/xxxx-ranger_k
   ms-RANGER_KMS_SERVER/conf
   export SQL_CONNECTOR_JAR=/opt/cloudera/cm/lib/postgresql-42.1.4.jre7.jar
   ```

4. Get the active directory for rangerkms process and copy the active directory path.

   ```
   ps -ef | grep rangerkms
   ```

5. Open proc.json and get the value for HADOOP_CREDSTORE_PASSWORD:

   ```
   vim /var/run/cloudera-scm-agent/process/xxxx-ranger_kms-RANGER_KMS_SERVER/
   proc.json
   export HADOOP_CREDSTORE_PASSWORD=hadoop_credstore_pwd
   ```

6. Run the following command:

   ```
   ${JAVA_HOME}/bin/java -Djava.library.path=/usr/safenet/lunaclient/jsp/lib/
    --add-opens java.xml.crypto/com.sun.org.apache.xml.internal.security.util
   s=ALL-UNNAMED -cp "${RANGER_KMS_HOME}/cred/lib/*:${RANGER_KMS_CONF}:${RA
   NGER_KMS_HOME}/ews/webapp/WEB-INF/classes/lib/*:${SQL_CONNECTOR_JAR}:${R
   ANGER_KMS_HOME}/ews/webapp/config:${RANGER_KMS_HOME}/ews/lib/*:${RANGER_
   KMS_HOME}/ews/webapp/lib/*:${RANGER_KMS_HOME}/ews/webapp/META-INF:${RANG
   ER_KMS_CONF}/*" org.apache.hadoop.crypto.key.HSM2DBMKUtil LunaProvider <
   partition-name>
   ```

7. Run the following command:

   ```
   ./HSMMK2DB.sh <provider> <HSM_PARTITION_NAME>
   ```

   Example :

   ```
   ./HSMMK2DB.sh LunaProvider
   ```

8. Enter the partition password when requested.
9. Login to the database that Ranger KMS is using, and validate whether master key is successfully migrated.

If the Ranger KMS database is Postgres, then

```
su - postgres
psql
Password : cloudera
\l
Find rangerkms db

 \c rangerkms
 select * from ranger_masterkey;
```

10. Login to Cloudera Manager and disable the HSM:

```
ranger.ks.hsm.enabled = false
```

11. Restart Ranger KMS.
12. Delete the master key from the partition.

```
/usr/safenet/lunaclient/bin/
./lunacm
lunacm:>role login -n co
enter password: **********
lunacm:>par con
lunacm:>par clear
proceed
```