

Migrating keys from KTS to Ranger KMS

Date published:

Date modified:



Legal Notice

© Cloudera Inc. 2025. 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

Migrating keys from Key Trustee Server to Ranger KMS.....	4
Key migration.....	4
Key migration in UCL.....	11
Navigator Encrypt key migration with HSM.....	18
Updating Navigator Encrypt.....	24
Rollback of Ranger KMS DB to KTS.....	25

Migrating keys from Key Trustee Server to Ranger KMS

You can migrate keys from Key Trustee Server to Ranger KMS DB.

Important considerations before migrating the keys.

Ranger KMS KTS keys are case sensitive but Ranger KMS DB keys are case insensitive. This means, it is possible to have the following keys in KMS KTS:

- KEY1 // All in capital case
- key1 // All in small case
- Key1 // Mix of both

KMS DB always stores the key names in lower case, even if you provide the key name in uppercase. It will error out when attempting to create duplicate keys with different cases. During key migration, this may cause issues. For instance, only one of key listed above will be imported.

Impact:

- Scenario 1: If you have a combination of such keys (with different cases) , only one key will be migrated. The migration tool will ignore the other keys.
- Scenario 2: If you have keys in any cases but no duplicates ,then after migration it will be stored in lowercase and will be visible on UI in lowercase.

Therefore, it is important to check the case of the keynames before starting the migration.

Key migration

This procedure describes how to migrate keys from Key Trustee Server to Ranger KMS.

Before you begin



Note: Key migration is supported only from Cloudera Runtime 7.1.9 release onwards.

- Locate the keys in Key Trustee Server.
 - Login to Ranger UI with Key Admin credentials.
 - Go to Key Management -> Select Service, to view the HDFS encryption zone keys with service Ranger KMS KTS.

- If Navigator Encryption is setup, locate its keys.
 - SSH in to the active KTS node.
 - Login to Postgres 14 database for Cloudera Runtime version 7.1.9.
 - The 'keytrustee' user is created with 'nologin' by default. Update the keytrustee user in /etc/passwd before accessing the database by running the following command:

```
sed -i "/keytrustee:x:${ id -u keytrustee }):$( id -g keytrustee ):Keytrustee User:\var\lib\keytrustee:\sbin\nologin\c\keytrustee:x:${ id -u keytrustee }):$( id -g keytrustee ):Keytrustee User:\var\lib\keytrustee:\bin\bash" /etc/passwd
```

- Run the following commands:

```
select handle from deposit;
```

For Cloudera Runtime version 7.1.9:

```
# sudo -u keytrustee LD_LIBRARY_PATH=/opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/14.2/lib /opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/14.2/bin/psql -p 11381 keytrustee

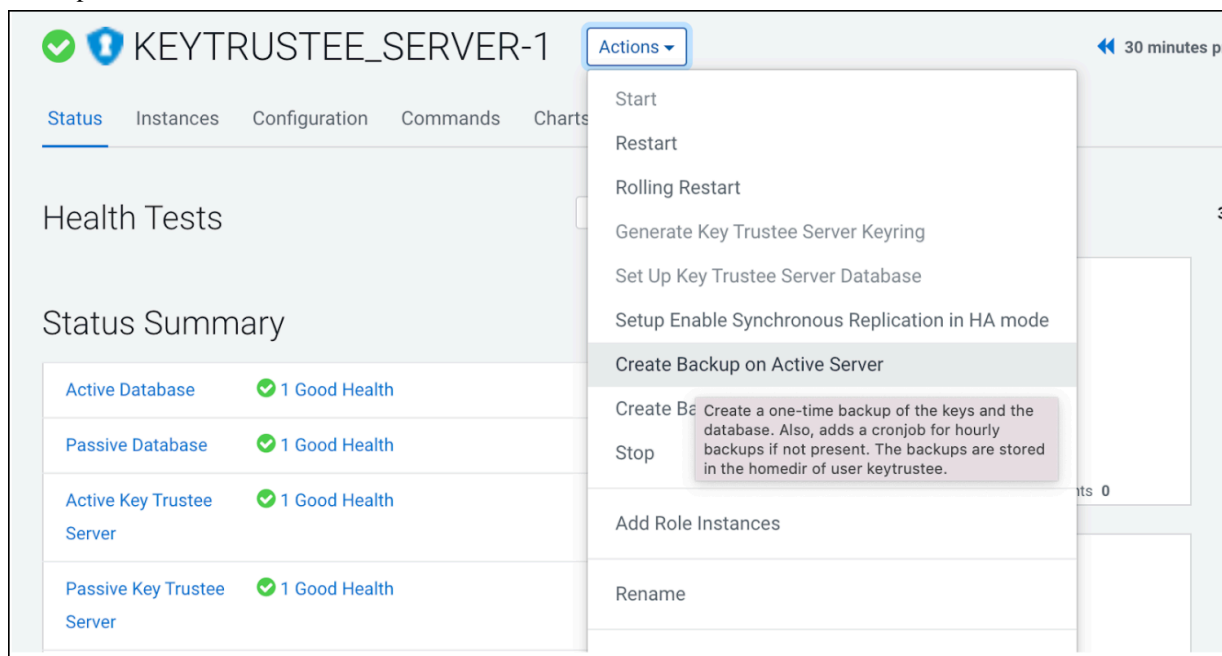
keytrustee=# select handle from deposit;
handle
-----
mykey1
mykey2

control

control
(6 rows)
```

Procedure

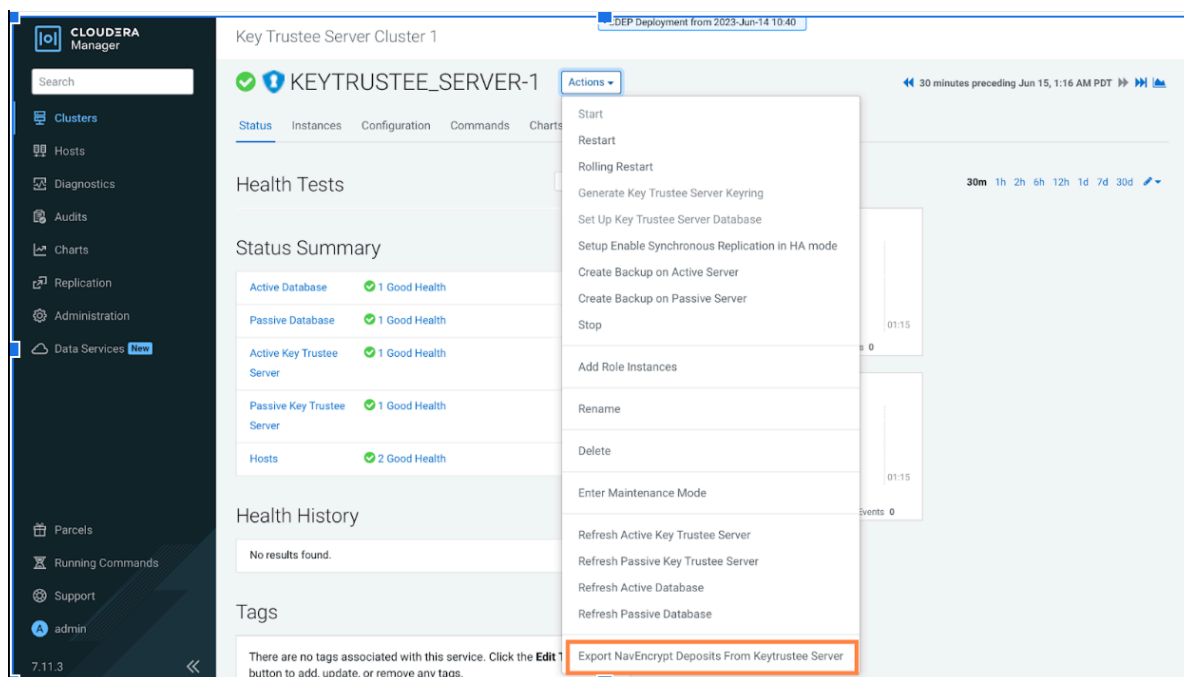
1. Backup the KTS database.



The backup will be created at /var/lib/keytrustee/ on Active KTS node.

2. Export the Navigator Encrypt keys.

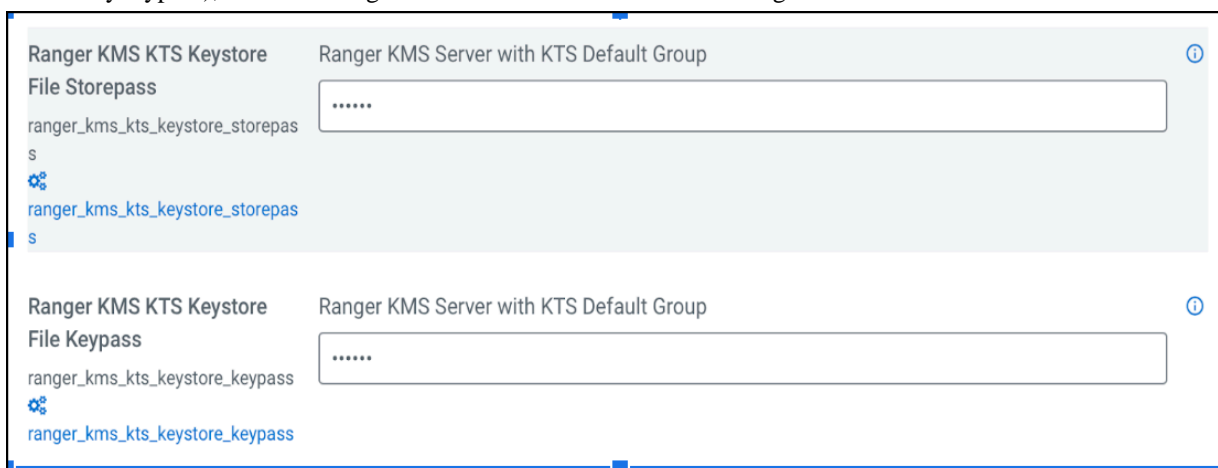
- a) Go to Cloudera Manager Key Trustee Server. Click on Actions Export NavEncrypt Deposits from Keytrustee Server.



This will generate the CSV required to import Navigator Encrypt keys in Ranger KMS DB after migration. The deposits.csv file will be created at /var/lib/keytrustee/.keytrustee.

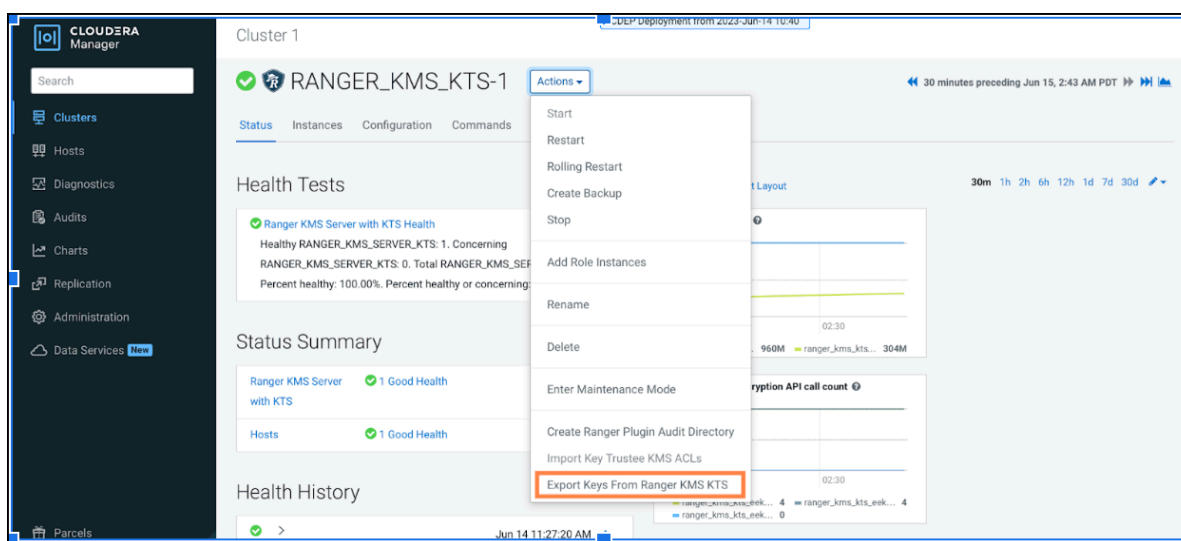
3. Back up the Ranger KMS KTS directory and generate a keystore file of existing encryption zone keys.

The keystore file is protected using Store password (default value : mystorepass) and Key password (default value : mykeypass), that are configurable in RANGER KMS KTS configuration.




Note: Record the key password and store password as they are required after migration while importing keys in Ranger KMS DB.

- a) Go to Cloudera Manager Ranger KMS KTS Actions and select Export keys from Ranger KMS KTS

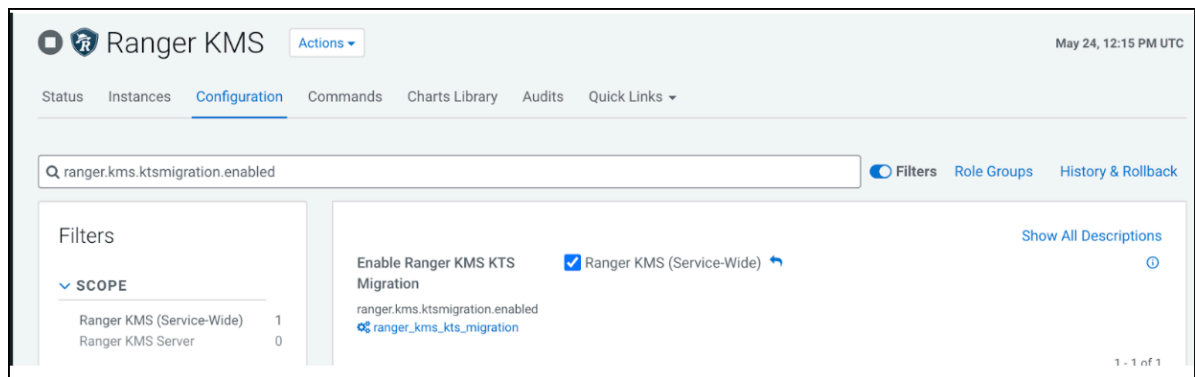


The service GPG keys backup and keystore file will be created at /var/lib/kms-keytrustee on Ranger KMS KTS node.

4. Stop HDFS and Ranger KMS KTS.
5. Delete the Ranger KMS KTS service from Cloudera Manager UI.
6. Add the Ranger KMS service from Cloudera Manager UI and follow the steps as per wizard. For more info, see related links for 'Configuring a database for Ranger or Ranger KMS' and 'Installing Ranger KMS backed by a Database and HA'

7. Enable the migration flag and complete the wizard.

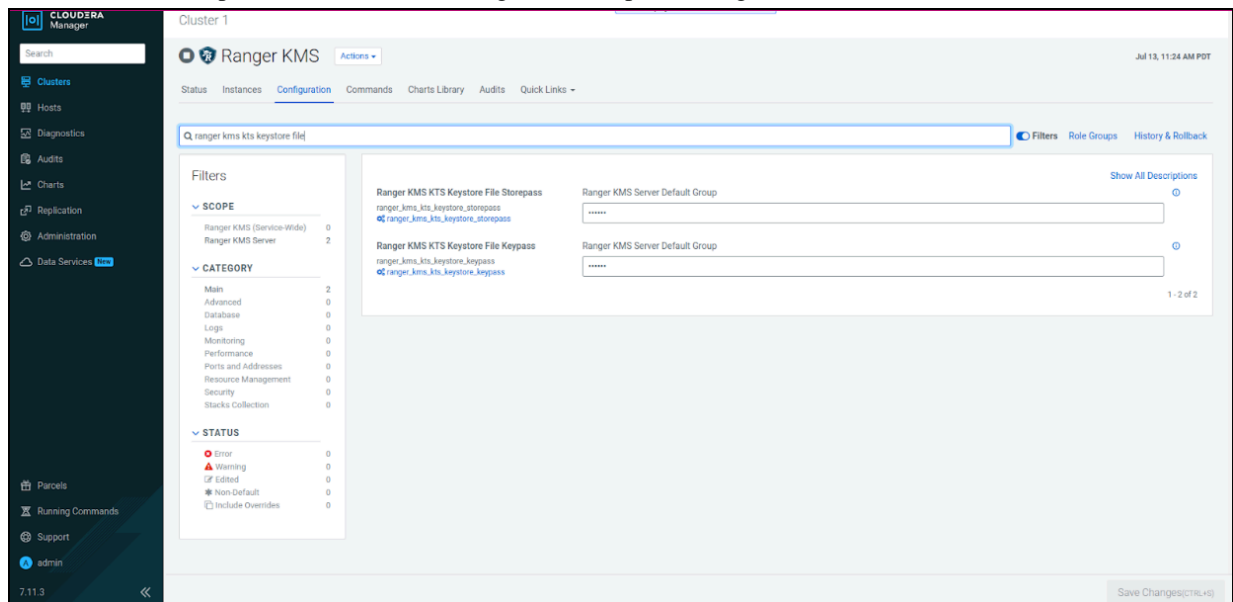
- a) Go to Cloudera Manager Ranger KMS Configuration and check **Enable Ranger KMS KTS Migration**.



Important: You must keep this property enabled even after migration. Disabling this property impacts the encryption zone key retrieval.

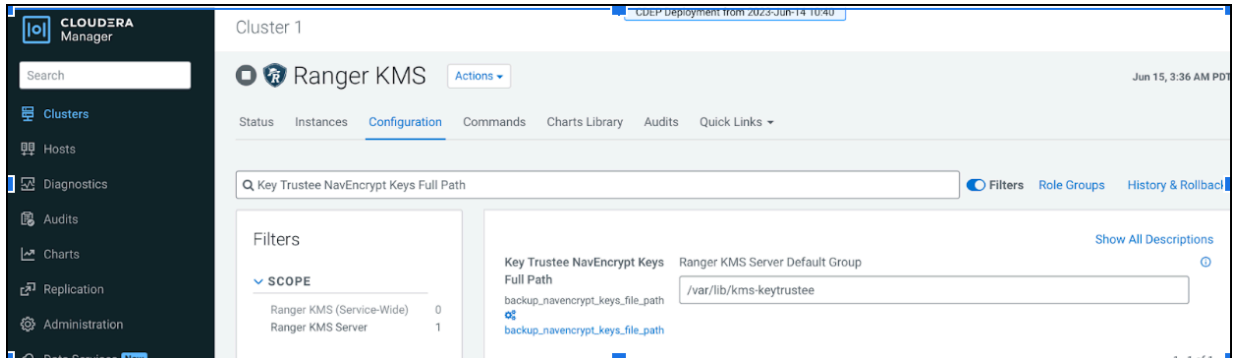
8. Configure the Key password (default value : mykeypass) and Store password (default value : mystorepass) in Ranger KMS configuration.

These are the same passwords that were configured in Step 3 in Ranger KMS KTS.



9. If Navigator Encrypt is configured on the cluster, copy deposits.csv file to the Ranger KMS node, and grant permission kms:kms.

The location is configurable using the property Key Trustee NavEncrypt Keys Full Path.



```
# scp root@dsktstokms-4.vpc.cloudera.com:/var/lib/keytrustee/.keytrustee/
deposits.csv /var/lib/kms-keytrustee
```

```
100% 10KB 8.2MB/s 00:00
# ls -ltr /var/lib/kms-keytrustee/deposits.csv
-rw-r--r-- 1 root root 10401 Jun 15 03:34 /var/lib/kms-keytrustee/depos
its.csv
# chown kms:kms /var/lib/kms-keytrustee/deposits.csv
# ls -ltr /var/lib/kms-keytrustee
total 64
-rw-r--r-- 1 kms kms 20480 Jun 14 11:22 kms_bak_dsktstokms-3_vpc_cloude
ra_com_2023-06-14_11-22-42.tar
-rw-r--r-- 1 kms kms 352 Jun 14 11:22 kt_bak_dsktstokms-3_vpc_cloudera
_com_2023-06-14_11-22-42.log
-rw-r--r-- 1 kms kms 20480 Jun 15 03:20 kms_bak_dsktstokms-3_vpc_cloud
era_com_2023-06-15_03-20-54.tar
-rw-r--r-- 1 kms kms 352 Jun 15 03:20 kt_bak_dsktstokms-3_vpc_clouder
a_com_2023-06-15_03-20-54.log
drwxr-xr-x 3 kms kms 55 Jun 15 03:21 keytrustee
-rw-r--r-- 1 kms kms 10401 Jun 15 03:34 deposits.csv
```

If you want to migrate the KMS hosts, then also copy the migratedKeyStore.jceks file to the Ranger KMS node.

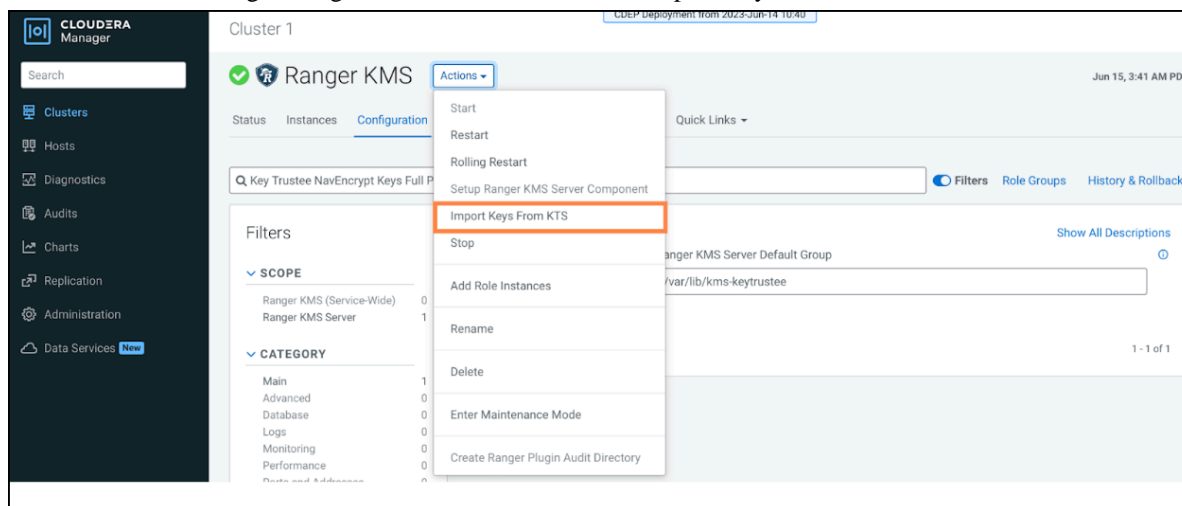
```
# scp root@dsktstokms-4.vpc.cloudera.com:/var/lib/keytrustee/.keytrustee/
migratedKeyStore.jceks /var/lib/kms-keytrustee/keytrustee

#ls -ltr /var/lib/kms-keytrustee/keytrustee
-rw-r--r-- 1 kms kms 1210 Jun 15 03:21 migratedKeyStore.jceks
```

10. Start Ranger KMS.

11. Import the keys from the keystore file and deposits.csv file for NavEncrypt.

- a) Go to Cloudera Manager Ranger KMS Actions and select Import Keys from KTS.



12. Restart Ranger KMS.

13. Start HDFS from Cloudera Manager UI.

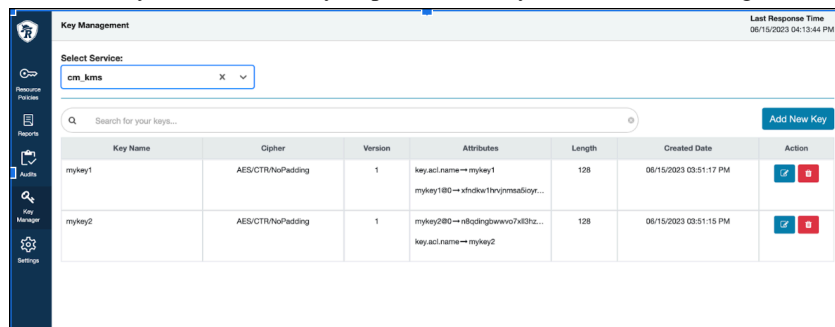
14. Stop the Key Trustee Server from Cloudera Manager UI.



Note: Do not remove KTS from Cloudera Manager UI.

Results

After the keys are successfully imported, the keys are visible on Ranger UI.



The Navigator Encrypt keys will be visible in Ranger KMS DB.

```
# mysql -u root -p

MariaDB [(none)]> use rangerkms;

MariaDB [rangerkms]> show tables;

+-----+
| Tables_in_rangerkms |
+-----+
| navencrypt_deposit   |
| ranger_keystore      |
| ranger_masterkey     |
+-----+
3 rows in set (0.00 sec)
```

What to do next

Post migration, after you verify that all the keys are migrated, delete KTS service from Cloudera Manager UI.

Related Information

[Configuring a database for Ranger or Ranger KMS](#)

[Installing Ranger KMS backed by a Database and HA](#)

Key migration in UCL

Learn how to migrate keys from Key Trustee Server (KTS) to Ranger KMS if you have already upgraded the Cloudera Manager version and want to upgrade the Cloudera version to 7.3.1 and above.

About this task

To upgrade your Cloudera version to 7.3.1 and above, you first need to upgrade to Cloudera Runtime 7.1.9. After you upgrade to Cloudera Runtime 7.1.9, you need to migrate your keys from KTS to Ranger KMS. After you migrate your keys, you can upgrade the Cloudera version to 7.3.1 and above from Cloudera Manager.

If you have upgraded your Cloudera Manager version, and proceed to upgrade your Cloudera version to 7.3.1 and above without migrating your keys from KTS to Ranger KMS, the following validation appears on the upgrade wizard page:

```
<service_name> in cluster <cluster_name> has been discontinued since 7.3.1 release, please follow this Migration doc to migrate your keys and install Ranger KMS to proceed. Note that if you are using an HSM, you cannot proceed with an upgrade at this time.
```



Note: If you want to migrate Navigator Encrypt keys with HSM, you need to do it manually before upgrading the Cloudera version to 7.3.1 and above. For details, see [Navigator Encrypt key migration with HSM](#).

Please read through and understand the root cause.

Root Cause

Services mentioned in the cluster are no longer supported in Cloudera 7.3.1 and have to be uninstalled before upgrading to the new version. Before uninstalling any services, you must be diligent to migrate any necessary data and keys stored in the respective services. If KTS service is installed in a separate cluster other than the one you intend to upgrade, it is recommended but not compulsory to uninstall this service.

Upgrade Cluster for Cluster 1

1 Getting Started
2 Shutdown Cluster
3 Upgrade Cluster
4 Summary

Getting Started

Current Version: Cloudera Runtime 7.1.9

Upgrade to Version: Cloudera Runtime 7.3.1-1.cdh7.3.1.p0.56640568

This wizard supports all major, minor, and maintenance upgrades to version 7.13.1 or smaller.
Check out the new [Upgrade Documentation](#) if you have not done so.

Before you proceed, be sure to check out the [Cloudera Enterprise Requirements and Supported Versions](#)

- Supported Operating Systems
- Supported Databases
- Supported JDK Versions

All checks must pass (✓) before you can continue.

[Run All Checks Again](#)

✗ Configuration Check

- RANGER_KMS_KTS-1 present in cluster Cluster 1 has been discontinued since 7.3.1 release, please follow this [migration document](#) to migrate your keys and install Ranger KMS to proceed.
- KEYTRUSTEE_SERVER-1 present in cluster Key Trustee Server Cluster 1 has been discontinued since 7.3.1 release, please follow this [migration document](#) to migrate your keys and install Ranger KMS to proceed.
- The RANGER_KMS_KTS-1 service cannot be upgraded: the Ranger KMS with Key Trustee Server service type is not supported for versions between CDH 7.2.0 and CDH 7.3.1. You might need to install a CSD that supports these versions, or remove this service.

All Configuration Issues

- ✓ Hosts Health Check
- ✓ Services Health Check
- ✓ Download and Distribute Parcel

[Cancel](#) [Back](#) [Continue](#)

Before you begin

Confirm the presence for keys in HDFS and Navigator Encrypt. Perform the following steps to locate the keys in KTS:

1. Login to Ranger UI with key admin credentials.
2. Go to Key Management Select Service to view the HDFS encryption zone keys with service Ranger KMS KTS.

Key Management Last Response Time: 08/02/2024 03:09:41 PM

Select Service:

Search for your keys...

Key Name	Cipher	Version	Attributes	Length	Created Date	Action
testkey1	AES/CTR/NoPadding	1	key.act.name=testkey1	128	08/01/2024 01:15:46 PM	✓ ✗
testkey2	AES/CTR/NoPadding	1	key.act.name=testkey2	128	08/01/2024 01:16:00 PM	✓ ✗

[Add New Key](#)

3. If you have setup Navigator Encrypt, locate its keys:
 - a. SSH into the active KTS node.
 - b. Login to Postgres 14.2 database.
 - c. Update the keytrustee user in /etc/passwd before accessing the database by running the following command:

```
sed -i "/keytrustee:x:${ id -u keytrustee }):${ id -g keytrustee }):Keytrustee User:\var\lib\keytrustee:\sbin\nologin\c\keytrustee:x:${ id -u keytrustee }):${ id -g keytrustee }):Keytrustee User:\var\lib\keytrustee:\bin\bash" /etc/passwd
```



Note: The keytrustee user is created with nologin by default.

- d. Run the following commands to locate the keys:

```
sudo -u keytrustee LD_LIBRARY_PATH=/opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/14.2/lib /opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/14.2/bin/psql -p 11381 keytrustee
```

```
select handle from deposit;  
handle  
-----  
mykey1  
mykey2  
  
control  
  
control  
(6 rows)
```

Procedure

1. Take backup of Ranger KMS KTS and KTS services and migrate the keys from Ranger KMS KTS and KTS services.



Note: If needed, change the store password and key password before executing the backup command.

- The keystore file is protected using store password (default value: mystorepass) and key password (default value: mykeypass), that are configurable in Ranger KMS KTS configuration. Change the values as per your requirement.

Ranger KMS KTS Keystore File Storepass ranger_kms_kts_keystore_storepass ranger_kms_kts_keystore_storepass	Ranger KMS Server Default Group *****
Ranger KMS KTS Keystore File Keypass ranger_kms_kts_keystore_keypass ranger_kms_kts_keystore_keypass	Ranger KMS Server Default Group *****

- Record the key password and store password. You need them after migration while importing keys in Ranger KMS DB.
- a) Go to Cloudera Manager.
 - b) Go to the cluster where Ranger KMS KTS is installed.
 - c) Click Actions Backup Keys from Keytrustee Server .

The Backup Keys from Keytrustee Server command is a cluster level command.

The screenshot shows the Cloudera Manager interface for Cluster 1. A context menu is open over the cluster list, with the option 'Backup Keys from Keytrustee Server' highlighted. A tooltip below the menu states: 'Takes backup of keys from Ranger KMS KTS & Keytrustee Server.' The background shows several performance charts: Cluster CPU, Cluster Disk IO, and HDFS IO.

- The backup of the KTS directory is created at `/var/lib/keytrustee/` on the active KTS node.
- The Backup Keys from Keytrustee Server command generates the CSV file required to import Navigator Encrypt keys in Ranger KMS DB after migration. The `deposits.csv` file is created at `/var/lib/keytrustee/.keytrustee`.
- The service GPG keys backup and the `migratedKeyStore.jceks` keystore file are created at `/var/lib/kms-keytrustee` on the Ranger KMS KTS node.

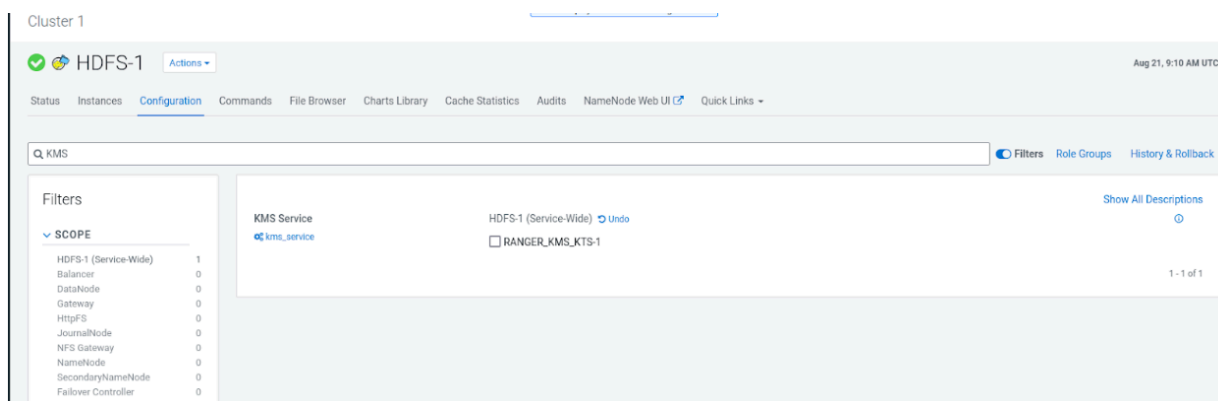
2. Verify that the deposits.csv file is generated at /var/lib/keytrustee/.keytrustee/deposits.csv on the active KTS node, by running the following command:

```
ls -ltr /var/lib/keytrustee/.keytrustee/deposits.csv
```

3. Verify that the migratedKeystore.jceks file is generated at /var/lib/kms-keytrustee/keytrustee on the Ranger KMS KTS node, by running the following command:

```
ls -lrt /var/lib/kms-keytrustee/keytrustee/migratedKeyStore.jceks
```

4. Stop the HDFS and Ranger KMS KTS services.
5. Remove service dependency of Ranger KMS KTS from the HDFS service.



6. Since Ranger KMS and Ranger KMS KTS services have similar configurations, take backup of the configurations that you modify as per your requirements. You can copy the configurations somewhere.

Here is a list of common configurations which should be migrated to Ranger KMS:

- RANGER_KMS_KTS_service_env_safety_valve
- ranger_kms_max_heap_size
- ranger_kms_kts_client_java_opts
- conf/dbks-site.xml_role_safety_valve
- conf/kms-site.xml_role_safety_valve
- conf/ranger-kms-site.xml_role_safety_valve
- RANGER_KMS_SERVER_KTS_role_env_safety_valve
- Max_log_size
- Process_username
- Process_groupname
- Kerberos_princ_name
- Ranger_accesslog_rotate_max_days
- Zookeeper_service
- Hadoop_kms_authentication_signer_secret_provider_zookeeper_path
- Hadoop_kms_authentication_signer_secret_provider
- Hadoop_kms_authentication_signer_secret_provider_zookeeper_auth_type



Note: The above list is not conclusive.

7. Delete Ranger KMS KTS service from the Cloudera Manager UI.
8. Stop the KTS service from the Cloudera Manager UI, even if it is installed in another cluster.
9. Delete KTS service from the Cloudera Manager UI.



Note: If the KTS service is installed in a separate cluster, you can ignore this step. If The KTS service is on the same cluster, you must delete it.

10. Install the Ranger KMS service from the Cloudera Manager UI and follow the steps as per the wizard.

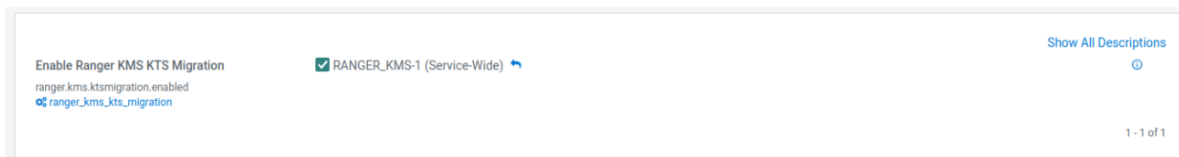
For more info, see related link for [Configuring a database for Ranger or Ranger KMS](#).



Note: Install Ranger KMS DB service on the same cluster from where you have deleted Ranger KMS KTS service. Cloudera also recommends using the same hosts where Ranger KMS KTS service was installed.

11. Enable the Enable Ranger KMS KTS Migration flag and complete the wizard.

- a) Go to Cloudera Manager Ranger KMS Configuration .
- b) Select Enable Ranger KMS KTS Migration.



Note: You must keep this property enabled even after migration. Disabling this property impacts the encryption zone key retrieval.

12. Start the Ranger KMS DB service.

Confirm that the service is successfully started and functional.

13. To confirm successful start of Ranger KMS service, perform the following steps:

- a) Login to Ranger UI with the user having access to Ranger KMS.
- b) Go to Service Manager Edit KMS service repository .
- c) Click Test Connection at the bottom of the page.

This establishes a connection to the Ranger KMS database. Confirm if the connection is successful.

- d) Got to Audits Plugins .
- e) Confirm if the plugin policies are successfully synced and have a 200 Http Response Code.

14. Login to the Ranger KMS Server host and copy the deposits.csv and migratedKeyStore.jceks files to /var/lib/kms-keytrustee/keytrustee/.

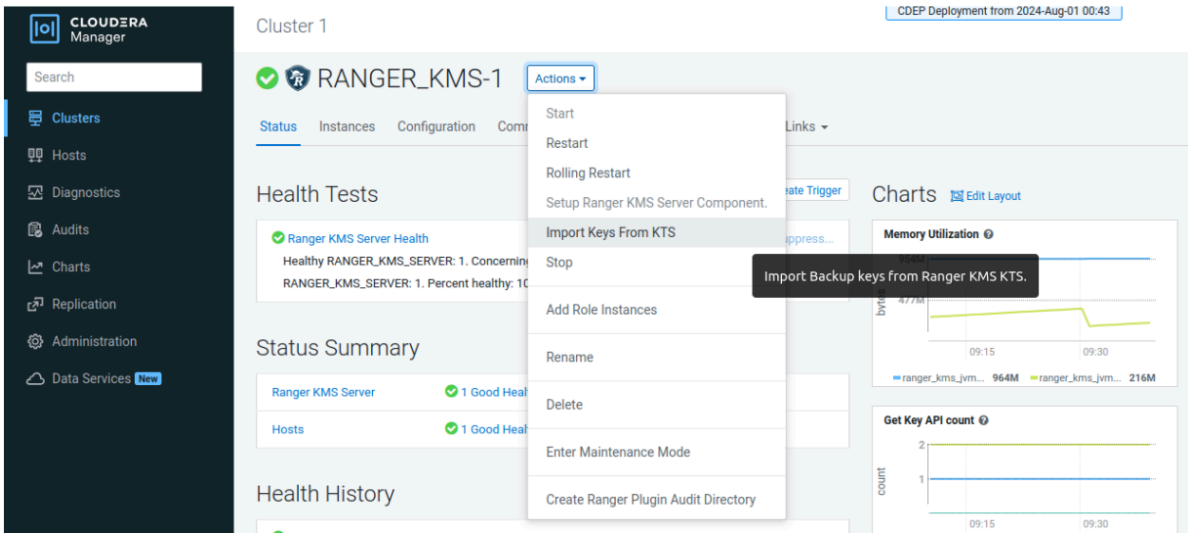
```
scp root@<ranger-kms-hostname>:/var/lib/kms-keytrustee/keytrustee/migratedKeyStore.jceks /var/lib/kms-keytrustee/keytrustee/

scp root@<ranger-kms-hostname>:/var/lib/keytrustee/.keytrustee/deposits.csv /var/lib/kms-keytrustee/keytrustee/

ls -lrt /var/lib/kms-keytrustee/keytrustee/deposits.csv
ls -lrt /var/lib/kms-keytrustee/keytrustee/migratedKeyStore.jceks
```

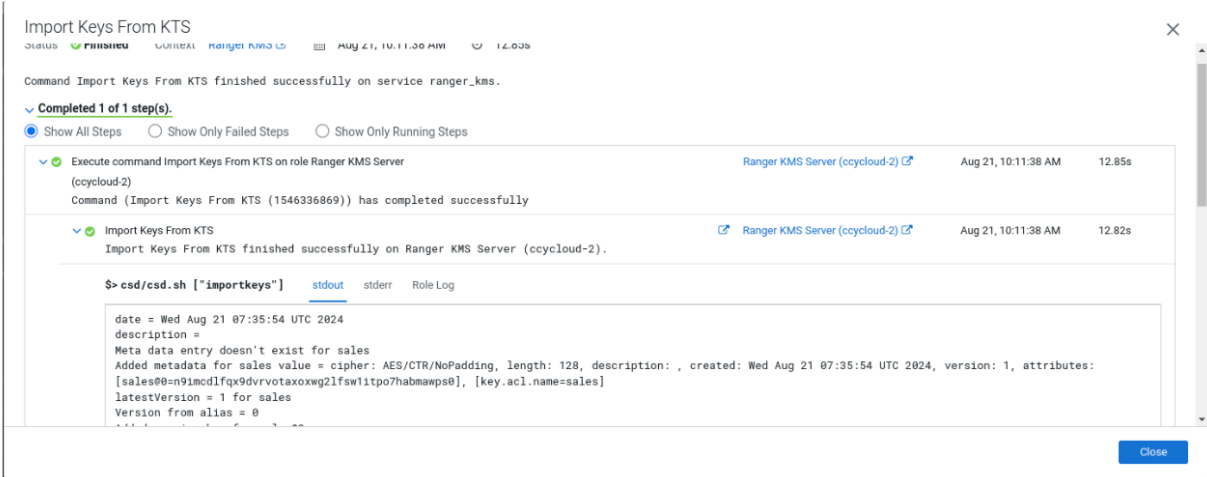

15. Run Import Keys from KTS command from Ranger KMS DB.

- a) Go to Cloudera Manager Ranger KMS .
- b) Select Actions Import Keys From KTS .

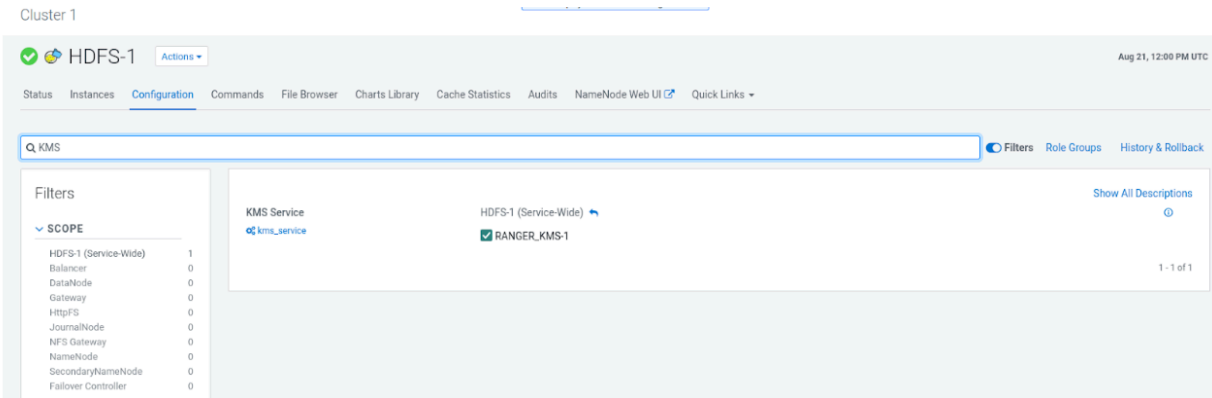


16. Confirm the successful execution of Import Keys from KTS command.

A message saying Keys from {keystore_file_path} has been successfully imported into RangerDB appears.



17. Ensure service dependency of Ranger KMS is selected from HDFS service.



18. Start HDFS service.

19. Verify the successful migration of keys from the Ranger UI.

- Login to Ranger UI with key admin credentials.
- Go to Key Management Select Service , to view the keys.

Key Management
Last Response Time
08/02/2024 03:11:13 PM

Select Service: cm_kms X ▾

Add New Key

Key Name	Cipher	Version	Attributes	Length	Created Date	Action
test1	AES/CTR/NoPadding	1	key.acd.name==test1 test1[@@→aggvjcrgkqfcpdm06dkesrmlwbq4dpe1...	128	08/01/2024 04:25:27 PM	<div>🔍 ⛔</div>
testm2	AES/CTR/NoPadding	1	key.acd.name==testm2 testm2[@@→ixisccg2qjcp6f67bo2sej999A5d36...	128	08/01/2024 04:25:25 PM	<div>🔍 ⛔</div>

- c) For Navigator Encrypt keys, login to Ranger KMS DB and check for Ranger KMS tables.

The Navigator Encrypt keys are visible in Ranger KMS DB.

```
mysql -u root -p
use rangerkms;
show tables;
```

Tables_in_rangerkms
navencrypt_deposit
ranger_keystore
ranger_masterkey

```
3 rows in set (0.00 sec)
```

What to do next

After successful verification, proceed to upgrade the Cloudera version from Cloudera Manager.



Note: Post migration confirmations:

- Once migration is completed successfully, navigate back to the upgrade wizard and click on Run All Checks Again. If the migration steps are performed correctly now the validation errors should be resolved and you will be able to proceed for the upgrade.
- A confirmation message appears to get user confirmation for a successful migration of KTS keys. Please select the checkbox to proceed.

▼! Other Tasks

- ☐ Only for customers migrating from Keytrustee Server & Ranger KMS KTS service to Ranger KMS in the database configuration (without an HSM). Ensure you have completed the key migration using the [migration document](#).
- ☐ I have completed the migration of keys from Ranger KMS KTS & Keytrustee Server to Ranger KMS.

- If you decide not to use encryption and do not want to migrate the keys to Ranger KMS DB, you can uninstall the blocking services and provide your confirmation through the check box.

▼! Other Tasks

- ☐ For customers using encryption at rest with Keytrustee Server or Ranger KMS, You must complete the migration of keys stored in Keytrustee Server to Ranger KMS in the database configuration (without an HSM) before proceeding. Please complete the migration using the [migration document](#). If this scenario is not applicable in your case, please check the box to proceed.
- ☐ I do not use encryption at rest or have decided to disable encryption at rest.

Navigator Encrypt key migration with HSM

Learn how to migrate Navigator Encrypt keys from Ranger KMS KTS to Ranger KMS, if you are using an HSM.

Before you begin

- You must have set up a 7.1.9 SP1 cluster with Navigator Encrypt registered to Ranger KMS + KTS.
- You must have set up Luna HSM for KTS and Key HSM. For details on how to set up Luna HSM on active and passive servers, see [Setting up Luna 7 HSM for KTS and Key HSM](#).

Procedure

1. Register Navigator Encrypt and start Navigator Encrypt service.

- a) Register the Navigator Encrypt with KTS by executing the following command:

```
sudo navencrypt register --server=https://<cluster-name>-4.vpc.cloudera.com:11371 --passive-server=https://<cluster-name>-5.vpc.cloudera.com:11371 --org=qa-test --auth=cloudera --skip-ssl-check # Add passphrase and register active and passive servers.
```



Note: After this is working, you can start Navigator Encrypt and start encrypting. You can encrypt two types of devices; a Linux loop device and a physical device. This topic describes how to use a loop device.

- b) Start Navigator Encrypt service by executing the following command:

```
sudo systemctl start navencrypt-mount
```

- c) View unused loop devices by executing the following command:

```
sudo losetup -f
```

You can then use the unused loop devices found by the above command.

- d) Create a new directory called `navencrypt_loop_devices` by executing the following command:

```
sudo mkdir /navencrypt_loop_devices
```

Datastore files for loop devices go here.

- e) Create `loop0`, which becomes one of the mount points, inside `nav mount`:

```
sudo mkdir -p /navencrypt_mount/loop0
```

- f) Create an input file called `diskimageloop0` of customized size for the unused loop device found in step c:

```
sudo dd if=/dev/zero of=/navencrypt_loop_devices/diskimageloop0 bs=100M count=10
```

The `dd` command creates a 500 GB file. You need to modify the `bs` and `count` values to generate the required file size.

- g) Use the available loop device with the `navencrypt-prepare -d` command to prepare the loop device mount point:

```
sudo navencrypt-prepare -d /navencrypt_loop_devices/diskimageloop0 /dev/loop0 /navencrypt_mount/loop0
```

- h) Execute the following command to get hostnames:

```
sudo yum install jq
```

- i) Run `ssh` using the keyboard-interactive password authentication mode:

```
sudo yum install sshpass
```

2. Install upgraded rpms for ctrustee, libkeytrustee, navencrypt-kernel-module, navencrypt from successful GBN with the following commands:

```
sudo yum install ctrustee-7.3.1.0-8_el9-8.x86_64.rpm<add latest rpm>
sudo yum install libkeytrustee-7.3.1.0-8_el9-8.x86_64.rpm
sudo yum install navencrypt-kernel-module-7.3.1.0-8_el9.x86_64.rpm
sudo yum install navencrypt-7.3.1.0-8_el9.x86_64.rpm
```

3. Ensure that the uuids.sh and generate_deposits.sh files are present in the /etc/navencrypt folder.
4. Ensure that ztrustee is present in the /usr/bin/ folder.
If not, try to install ctrustee.rpm again.
5. Generate the gpg key.

```
[root@nekts-mhsm-rhel88-1 navencrypt]# gpg --list-secret-keys
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
/root/.gnupg/pubring.kbx
-----
sec   rsa2048 2024-08-23 [SC]
      AA1F26481634C37682A063B4071E6471A0B8EE9D
uid           [ultimate] Snehalatha Venkatesh <snehalatha.venkatesh@cloudera.com>
ssb   rsa2048 2024-08-23 [E]
[root@nekts-mhsm-rhel88-1 navencrypt]# gpg --full-generate-key
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n>  = key expires in n days
    <n>w  = key expires in n weeks
    <n>m  = key expires in n months
    <n>y  = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: <Name>
Email address: <cloudera email address>
Comment:
You selected this USER-ID:
    "<Name><email address>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
```

```

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 7A234E85BC56BBD0 marked as ultimately trusted
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/CB52
70B06F39E4BB92BC80017A234E85BC56BBD0.rev'
public and secret key created and signed.
pub   rsa2048 2024-09-10 [SC]
        CB5270B06F39E4BB92BC80017A234E85BC56BBD0
uid                               Snehalatha Venkatesh <snehalatha.venkatesh@cloude
ra.com>
sub   rsa2048 2024-09-10 [E]

```

6. To hold the gpg key for further purpose, execute the following lines:

```

echo "gpg_passphrase" > /etc/navencrypt/passphrase.txt -> "gpg_passphrase"
is the passphrase given while generating gpg key
Example: echo "cloudera123" > /etc/navencrypt/passphrase.txt

chmod 600 /etc/navencrypt/passphrase.txt

```

7. Run the uuids.sh shell script and ensure that you get the success message and check for the retrieved_keys file in the /etc/navencrypt folder, as follows:

```

[root@nechts-keyhsm-1 navencrypt]# sh uuids.sh
Processing UUID: yMex10Q5Nb2g674GP7wOiTmevXL0GSA1XZMSGU5ZRSq
Successfully retrieved and saved key for UUID yMex10Q5Nb2g674GP7wOiTmevX
L0GSA1XZMSGU5ZRSq.

```

8. Run the generate_deposits.sh file and get the success message, as follows, and check for deposits.csv and consolidated_deposits.csv files in the /etc/navencrypt folder.

```

[root@tgnechts-mhsm-rhel8-1 navencrypt]# sh generate_deposits.sh -> Enter
passphrase given for gpg key.
Key file content encrypted and signed successfully.
Added UUID 31D77qRTgZBe5HkgAdp78uONhJhOI1Jh0hdqpXrKLCB to /etc/navencrypt/
deposits.csv.
Deposits CSV file created at /etc/navencrypt/deposits.csv.
Enter Cloudera Manager hostname: tgnechts-mhsm-rhel8-1.vpc.cloudera.com
Enter Cloudera Manager port: 7183
Node list: tgnechts-mhsm-rhel8-1.vpc.cloudera.com,tgnechts-mhsm-rhel8-2.vpc
.cloudera.com,tgnechts-mhsm-rhel8-3.vpc.cloudera.com,tgnechts-mhsm-rhel8-4
.vpc.cloudera.com,tgnechts-mhsm-rhel8-5.vpc.cloudera.com
Ensuring SSH access for tgnechts-mhsm-rhel8-1.vpc.cloudera.com...
Generating public/private rsa key pair.
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:iL/yH/5sz98T4pFZYmzQ4xYgegjlilElK5F5Fqu9EPpU root@tgnechts-mhsm-rhel
8-1.vpc.cloudera.com
The key's randomart image is:
+---[RSA 3072]---+
| .o ..+.. . .o |
| o o = + o .. + |
| o = . + . + o |
| . o .... B . |
| o ..E. S + = |
| + .. = . |
| = . . . o . |
| o o o o.. . o |
| . oo.ooo.o.. o |
+---[SHA256]-----+

```

```

# tgnkts-mhsm-rhel8-1.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
# tgnkts-mhsm-rhel8-1.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
# tgnkts-mhsm-rhel8-1.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
root@tgnkts-mhsm-rhel8-1.vpc.cloudera.com's password:
Ensuring SSH access for tgnkts-mhsm-rhel8-2.vpc.cloudera.com...
# tgnkts-mhsm-rhel8-2.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
# tgnkts-mhsm-rhel8-2.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
# tgnkts-mhsm-rhel8-2.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
root@tgnkts-mhsm-rhel8-2.vpc.cloudera.com's password:
Ensuring SSH access for tgnkts-mhsm-rhel8-3.vpc.cloudera.com...
# tgnkts-mhsm-rhel8-3.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
# tgnkts-mhsm-rhel8-3.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
# tgnkts-mhsm-rhel8-3.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
root@tgnkts-mhsm-rhel8-3.vpc.cloudera.com's password:
Ensuring SSH access for tgnkts-mhsm-rhel8-4.vpc.cloudera.com...
# tgnkts-mhsm-rhel8-4.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
# tgnkts-mhsm-rhel8-4.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
# tgnkts-mhsm-rhel8-4.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
root@tgnkts-mhsm-rhel8-4.vpc.cloudera.com's password:
Ensuring SSH access for tgnkts-mhsm-rhel8-5.vpc.cloudera.com...
# tgnkts-mhsm-rhel8-5.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
# tgnkts-mhsm-rhel8-5.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
# tgnkts-mhsm-rhel8-5.vpc.cloudera.com:22 SSH-2.0-OpenSSH_8.0
root@tgnkts-mhsm-rhel8-5.vpc.cloudera.com's password:
Copying deposits.csv from tgnkts-mhsm-rhel8-1.vpc.cloudera.com...
deposits.csv
                                                                    100%
  773   903.3KB/s   00:00
Copying deposits.csv from tgnkts-mhsm-rhel8-2.vpc.cloudera.com...
deposits.csv
                                                                    100%   77
    3   903.3KB/s   00:00
Copying deposits.csv from tgnkts-mhsm-rhel8-3.vpc.cloudera.com...
deposits.csv
                                                                    100%   77
    3   903.3KB/s   00:00
Copying deposits.csv from tgnkts-mhsm-rhel8-4.vpc.cloudera.com...
deposits.csv
                                                                    100%   77
    3   903.3KB/s   00:00
Copying deposits.csv from tgnkts-mhsm-rhel8-5.vpc.cloudera.com...
deposits.csv
                                                                    100%   77
    3   903.3KB/s   00:00
Consolidated CSV file created at /etc/navencrypt/consolidated_deposits.
csv.

```

9. After the deposits.csv file is created, login to Cloudera Manager UI.

10. Check for node(instance) in Ranger KMS KTS.

- a) Go to Cloudera Manager Ranger KMS KTS service.
- b) Go to Instances.
- c) Check the node(instance).

11. Stop HDFS and Ranger KMS KTS services.

12. Delete Ranger KMS KTS service from Cloudera Manager UI.

13. Add Ranger KMS service from Cloudera Manager UI.

To add database details, ssh to Ranger KMS node as shown in step 7 and execute the following commands, where <fqdn> is the node(instance) name:

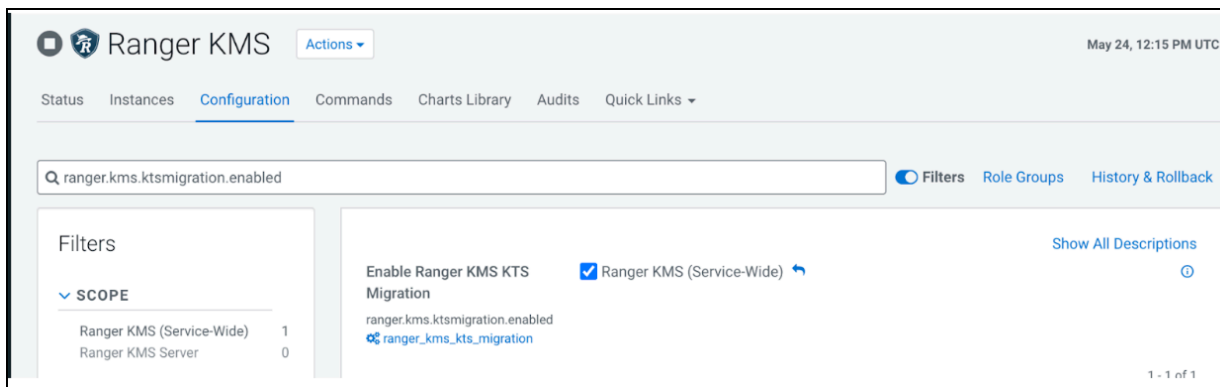
```

mysql -u root -p # Login to database with password
CREATE DATABASE rangerkms;

```

```
CREATE USER 'rangerkms'@'%' IDENTIFIED BY 'cloudera';
CREATE USER 'rangerkms'@'localhost' IDENTIFIED BY 'cloudera';
CREATE USER 'rangerkms'@'<fqdn>' IDENTIFIED BY 'cloudera';
GRANT ALL PRIVILEGES ON rangerkms.* TO 'rangerkms'@'%';
GRANT ALL PRIVILEGES ON rangerkms.* TO 'rangerkms'@'<fqdn>';
FLUSH PRIVILEGES;
```

14. Enable the migration flag under configuration for Ranger KMS and complete the wizard.



15. Configure the Key password (default value : mykeypass) and Store password (default value : mystorepass) which are used or entered before migration within Ranger KMS KTS service.

16. Copy the deposits.csv file created in step 5 to the KMS node.

```
[root@nekts-mig-hsm-3 ~]# scp root@nekts-mig-hsm-1.vpc.cloudera.com:/etc/
navencrypt/consolidated_deposits.csv /var/lib/kms-keytrustee/deposits.csv
root@nekts-mig-hsm-1.vpc.cloudera.com's password:
deposits.csv
100% 664 1.9MB/s
00:00
[root@nekts-mig-hsm-3 ~]# ls /var/lib/kms-keytrustee/
deposits.csv keytrustee kms_bak_nekts-mig-hsm-3_vpc_cloudera_com_2024-08-
01_01-25-04.tar kt_bak_nekts-mig-hsm-3_vpc_cloudera_com_2024-08-01-01-
25-04.log
```

17. Start the Ranger KMS service.

18. Create a new file called importPayload.json:

```
vi /tmp/importPayload.json
```

19. Add the following line to the new file called importPayload.json:

```
{
  "path": "/var/lib/kms-keytrustee/deposits.csv"
}
```

20. Run the following API to import Navigator Encrypt keys from the /tmp/deposits.csv file to Ranger KMS DB.

```
[root@nekts-mig-hsm-3 ~]# kinit systest
Password for systest@VPC.CLOUDERA.COM:
[root@nekts-mig-hsm-3 ~]# curl -ivk --negotiate -u : -H "Content-Type:
application/json" -X POST "https://nekts-mig-hsm-3.vpc.cloudera.com:9494/
kms/v2/deposit/import" -d @/tmp/importPayload.json
```

What to do next

Upgrade Cloudera version to 7.3.1 or above.

Updating Navigator Encrypt

You must update Navigator Encrypt to version 7.1.9 in order for it to work with Ranger KMS.

About this task

Learn how to update RHEL compatible Navigator Encrypt. For information on SLES and Ubuntu compatible Navigator Encrypt installation, refer to 'Installing Cloudera Navigator Encrypt'.

Procedure

1. SSH as root to the host where Navigator Encrypt is installed.
2. Untar the new zip package.

```
tar zxvf navigator-encrypt-7.1.9.0-64-redhat8.tar.gz --directory navencrypt-7.1.9.0-repo
```

3. Stop Navigator Encrypt.

```
systemctl stop navencrypt-mount
```

4. Make a copy of /etc/navencrypt/.

```
cp -rp /etc/navencrypt/ .
```

5. Create, and edit repo file etc/yum.repos.d/navencrypt-7.1.9.0.repo, by adding the following lines.

```
[navencrypt-7.1.9.0]
name=navencrypt-7.1.9.0
baseurl=file:///root/navencrypt-7.1.9.0-repo
gpgkey=file:///root/navencrypt-repo/nepub.asc
enabled=1
gpgcheck=1
```

6. Ensure that the repository is accepted, and three packages are present.

```
# yum repolist
# yum list available --disablerepo=* --enablerepo=navencrypt-7.1.9.0
```

7. Edit the /etc/navencrypt/keytrustee/ztrustee.conf file and make the following changes:

- Change all the URLs to point to Ranger KMS.
- Change "PROTOCOL" to "json-cleartext".
- Add "IS_KMS": true

This is an example of a ztrustee.conf with KTS urls and port :

```
[root@gsne-2 navencryptFiles]# cat /etc/navencrypt/keytrustee/ztrustee.conf
{
    "LOCAL_FINGERPRINT": "2048R/51E9DD52660E134E74ECBA8AF0E1ED9AC6AC3BC9",
    "REMOTES": {
        "kts1.cloudera.com": {
            "REMOTE_SERVER": "https://kts1.cloudera.com:11371"
        },
        "HKP_PORT": 11371,
        "HKP_SCHEME": "https",
        "DEFAULT": true,
    }
}
```



```

        "HKP_TIMEOUT": 60,
        "REMOTE_SERVERS": ["https://kts1.cloudera.com:1137
1", "https://kts2.cloudera.com:11371"],
        "SSL_INSECURE": true,
        "PROTOCOL": "json-encrypt",
    }
}

```

This is an example of `ztrustee.conf` with Ranger KMS urls and port :

```

[root@gsne-2 ~]# cat /etc/navencrypt/keytrustee/ztrustee.conf
{
    "LOCAL_FINGERPRINT": "2048R/51E9DD52660E134E74ECBA8AF0E1ED9AC6AC3
BC9",
    "REMOTES": {
        "kms1.cloudera.com": {
            "REMOTE_SERVER": "https://kms1.cloudera.com:9494",
            "HKP_PORT": 11371,
            "HKP_SCHEME": "https",
            "DEFAULT": true,
            "HKP_TIMEOUT": 60,
            "REMOTE_SERVERS": ["https://kms1.cloudera.com:9494
", "https://kms2.cloudera.com:9494"],
            "SSL_INSECURE": true,
            "PROTOCOL": "json-cleartext",
            "IS_KMS": true
        }
    }
}

```

8. Update to new versions of Navigator Encrypt.

```

yum update libkeytrustee
yum update navencrypt-kernel-module
yum update navencrypt

```

9. Start Navigator Encrypt.

```

systemctl start navencrypt-mount

```

10. Check the version and status of Navigator Encrypt.

```

navencrypt --version;
navencrypt status -m;
navencrypt key --verify --only-keytrustee

```

Results

The version of Navigator Encrypt is 7.1.9. The status shows "navencrypt module is running". After entering the master-passphrase, navencrypt outputs "VALID".

Related Information

[Installing Cloudera Navigator Encrypt](#)

Rollback of Ranger KMS DB to KTS

This procedure describes how to rollback from Ranger KMS DB to Key Trustee Server (KTS) in case of failure during or after key migration.

Before you begin



Note: Any keys created after migration from KTS to Ranger KMS will not be present after rollback.

Procedure

1. Stop Ranger KMS and HDFS from the Cloudera Manager UI.
2. Stop Key Trustee Server



Note: KTS was stopped after the keys were migrated, but not deleted.

3. Delete Ranger KMS service from the Cloudera Manager UI.
4. Add Ranger KMS KTS service from the Cloudera Manager UI.



Note: While adding Ranger KMS KTS, restore the GPG keys backup created during the migration at location `/var/lib/kms-keytrustee/keytrustee/.keytrustee/`. If the backup files were not deleted while removing Ranger KMS KTS during migration, it will be already present at required location

5. Start Ranger KMS KTS and HDFS from the Cloudera Manager UI.