Cloudera Manager 7.11.3

Cloudera Navigator Key HSM

Date published: 2020-11-30
Date modified: 2024-02-23

https://docs.cloudera.com/

Legal Notice

Contents

Cloudera Navigator Key HSM Overview

Cloudera Navigator Key HSM allows Cloudera Navigator Key Trustee Server to seamlessly integrate with a hardware security module (HSM). Key HSM enables Key Trustee Server to use an HSM as a root of trust for cryptographic keys, taking advantage of Key Trustee Server's policy-based key and security asset management capabilities while at the same time satisfying existing, internal security requirements regarding treatment of cryptographic materials.

For instructions on installing Key HSM, see Installing Cloudera Navigator Key HSM.

Initializing Navigator Key HSM

Key HSM is initialized using a series of CLI commands and prompts. The setup information you enter is dependent on which type of HSM you are using with Navigator Key HSM.

Before you begin

Before initializing Navigator Key HSM, verify that the HSM is properly configured and accessible from the Key HSM host, and that the HSM client libraries are installed on the Key HSM host:

- SafeNet Luna

  Install the SafeNet Luna client. No additional configuration is needed.
- SafeNet KeySecure

  Extract the KeySecure client tarball in the Key HSM library directory (/usr/share/keytrustee-server-keyhsm/).
- Thales

  Install the Thales client service. Copy nCipherKM.jar, jcetools.jar, and rsaprivenc.jar from the installation media (usually located in opt/nfast/java/classes relative to the installation media mount point) to the Key HSM library directory (/usr/share/keytrustee-server-keyhsm/).
- AWS CloudHSM

  Install the AWS CloudHSM client. No additional configuration is needed.

See your HSM product documentation for more information on installing and configuring your HSM and client libraries.

Note: When using an HSM with Key Trustee Server and Navigator Encrypt, encrypting many block devices may exceed the capacity of the HSM. A key is created in the HSM for each encrypted block device, so be sure that your HSM can support your encryption requirements.

For Luna v7, the keyhsm user must be added to the hsmusers group with the following command:

```
sudo usermod -a -G hsmusers keyhsm
```

Procedure

1. To initialize Key HSM, use the service keyhsm setup command in conjunction with the name of the target HSM distribution:

   ```
   sudo service keyhsm setup [keysecure|thales|luna|cloudhsm]
   ```

   For all HSM distributions, you are prompted for the IP address and port number that Key HSM listens on.

   

   Important: If you have implemented Key Trustee Server high availability, initialize Key HSM on each Key Trustee Server.

   Cloudera recommends using the loopback address (127.0.0.1) for the listener IP address and 9090 as the port number:

   ```
   -- Configuring keyHsm General Setup --
   Cloudera Recommends to use 127.0.0.1 as the listener port for Key HSM
   Please enter Key HSM SSL listener IP address: [127.0.0.1]127.0.0.1
   Will attempt to setup listener on 127.0.0.1
   Please enter Key HSM SSL listener PORT number: 9090
   validate Port:                                    :[ Successful ]
   ```

2. If the setup utility successfully validates the listener IP address and port, you are prompted for additional information specific to your HSM. For HSM-specific instructions, continue to the HSM-Specific Setup for Cloudera Navigator Key HSM on page 5 section for your HSM.
3. The Key HSM keystore defaults to a strong, randomly-generated password. However, you can change the keystore password in the application.properties file:

   ```
   keyhsm.keystore.password.set=yes
   ```

   Next, run the service keyhsm setup command with the name of the HSM to which the keystore password applies. You will be prompted to enter the new keystore password, which must be a minimum of six characters in length:

   ```
   sudo service keyhsm setup [keysecure|thales|luna|cloudhsm]
   ```

4. After initial setup, the configuration is stored in the /usr/share/keytrustee-server-keyhsm/application.properties file, which contains human-readable configuration information for the Navigator Key HSM server.

   

   Important: The truststore file created during Key HSM initialization must be stored at /usr/share/keytrustee-server-keyhsm/. There is no way to change the default location.

   For additional details about keystores and truststores, see Understanding Keystores and Trustores.

HSM-Specific Setup for Cloudera Navigator Key HSM

Additional HSM-specific setup information for Key HSM.

SafeNet KeySecure

Note: KeySecure was previously named DataSecure, but the Key HSM configuration process is the same for both.

Prerequisites

Before setting up SafeNet KeySecure, be sure to:

- Set the protocol to NAE-XML

- Set Allow Key and Policy Configuration Operations to enabled
- Set Password Authentication to required
- Disable Client Certificate Authentication
- Set KeySecure Crypto Operations to activated

For additional details about SafeNet KeySecure settings, see the SafeNet KeySecure product documentation.

After entering the Key HSM listener IP address and port, the HSM setup for SafeNet KeySecure prompts for login credentials, the IP address of the KeySecure HSM, and the port number:

```
-- Ingrian HSM Credential Configuration --
Please enter HSM login USERNAME: keyhsm
Please enter HSM login PASSWORD: ******

Please enter HSM IP Address or Hostname: 172.19.1.135
Please enter HSM Port number: 9020
```

If the connection is successful, the following message is displayed:

```
Valid address:                                    :[ Successful ]
```

The KeySecure setup utility then prompts you whether to use SSL:

```
Use SSL? [Y/n] Y
```

If you choose to use SSL, Key HSM attempts to resolve the server certificate, and prompts you to trust the certificate:

```
  [0]         Version: 3
         SerialNumber: 0
             IssuerDN: C=US,ST=TX,L=Austin,O=ACME,OU=Dev,
CN=172.19.1.135,E=webadmin@example.com
           Start Date: Thu Jan 29 09:55:57 EST 2015
           Final Date: Sat Jan 30 09:55:57 EST 2016
            SubjectDN: C=US,ST=TX,L=Austin,O=ACME,OU=Dev,
CN=172.19.1.135,E=webadmin@example.com
           Public Key: RSA Public Key
            modulus: abe4a8dcef92e145984309bd466b33b35562c7f875
                     1d1c406b1140e0584890272090424eb347647ba04b
                     34757cacc79652791427d0d8580a652c106bd26945
                     384b30b8107f8e15d2deba8a4e868bf17bb0207383
                     7cffef0ef16d5b5da5cfb4d3625c0affbda6320daf
                     7c6b6d8adfcb563960fcd1207c059300feb6513408
                     79dd2d929a5b986517531be93f113c8db780c92ddf
                     30f5c8bf2b0bea60359b67be306c520358cc0c3fc3
                     65500d8abeeac99e53cc2b369b2031174e72e6fca1
                     f9a4639e09240ed6d4a73073885868e814839b09d5
                     6aa98a5a1e230b46cdb4818321f546ac15567c5968
                     33be47ef156a73e537fd09605482790714f4a276e5
                     f126f935
    public exponent: 10001

  Signature Algorithm: SHA256WithRSAEncryption
            Signature: 235168c68567b27a30b14ab443388039ff12357f
                       99ba439c6214e4529120d6ccb4a9b95ab25f81b4
                       7deb9354608df45525184e75e80eb0948eae3e15
                       c25c1d58c4f86cb9616dc5c68dfe35f718a0b6b5
                       56f520317eb5b96b30cd9d027a0e42f60de6dd24
                       5598d1fcea262b405266f484143a74274922884e
                       362192c4f6417643da2df6dd1a538d6d5921e78e
                       20a14e29ca1bb82b57c02000fa4907bd9f3c890a
                       bdae380c0b4dc68710deeaef41576c0f767879a7
                       90f30a4b64a6afb3a1ace0f3ced17ae142ee6f18
```

```
                        5eff64e8b710606b28563dd99e8367a0d3cbab33
                        2e59c03cadce3a5f4e0aaa9d9165e96d062018f3
                        6a7e8e3075c40a95d61ebc8db43d77e7
        Extensions:
                        critical(false) BasicConstraints: isCa(true)
                        critical(false) NetscapeCertType: 0xc0
Trust this server? [y/N] Y
Trusted server:                                    :[ Successful ]
```

CloudHSM

Before completing the CloudHSM setup, verify that the CloudHSM connection is properly configured by running either the cloudhsm_mgmt_util or key_mgmt_util tool and verifying that the tool connects to the HSM successfully. After validating this connection, you can exit the tool:

```
$ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.
cfg
Connecting to the server(s), it may take time
depending on the server(s) load, please wait...

Connecting to server '<hsm ip>': hostname '<hsm ip>', port 2225...
Connected to server '<hsm ip>': hostname '<hsm ip>', port 2225.
aws-cloudhsm>quit
```

After entering the Key HSM listener IP address and port identifier, the HSM setup for CloudHSM prompts you for: the crypto user name and crypto user password.

Note: You can configure different user types in the CloudHSM client (for example, the crypto user type manages key operations). Be sure to enter the user type crypto (CU) here.

```
-- Configuring CloudHSM --
Please enter the crypto user name: <username>
Please enter the crypto user password  (input suppressed):
Configuration saved in 'application.properties' file
Configuration stored in: 'application.properties'. (Note: You can also use
 keyhsm settings to quickly view your current configuration)
```

Thales HSM

By default, the Thales HSM client process listens on ports 9000 and 9001. The Cloudera Manager agent also listens on port 9000. To prevent a port conflict, you must change the Thales client ports. Cloudera recommends using ports 11500 and 11501.

To change the Thales client ports, run the following commands:

```
sudo /opt/nfast/bin/config-serverstartup --enable-tcp --enable-privileged-tc
p --port=11500 --privport=11501
sudo /opt/nfast/sbin/init.d-ncipher restart
```

To configure Key HSM to use the modified port, edit the /usr/share/keytrustee-server-keyhsm/start.sh file and add the -DNFAST_SERVER_PORT=11500 Java system property. For example:

```
java -classpath "*:/usr/safenet/lunaclient/jsp/lib/*:/opt/nfast/java/classes
/*" -Djava.library.path=/usr/safenet/lunaclient/jsp/lib/ -DNFAST_SERVER_PORT
=11500 com.cloudera.app.run.Program $@
```

Before completing the Thales HSM setup, run the nfkminfo command to verify that the Thales HSM is properly configured:

```
$ sudo /opt/nfast/bin/nfkminfo
  World generation  2
  state       0x17270000 Initialised Usable Recovery !PINRecovery !Existing
Client
              RTC  NVRAM FTO !AlwaysUseStrongPrimes SEEDebug
```

If state reports !Usable instead of Usable, configure the Thales HSM before continuing. See the Thales product documentation for instructions.

After entering the Key HSM listener IP address and port, the HSM setup for Thales prompts for the OCS card password:

```
Please enter the OCS Card Password (input suppressed):

Configuration saved in 'application.properties' file
Configuration stored in: 'application.properties'. (Note: You can also use
service keyHsm settings to quickly view your current configuration)
```

Luna HSM

Important: If you have implemented Key Trustee Server high availability, ensure that the Luna client on each Key Trustee Server is configured with access to the same partition. See the Luna product documentation for instructions on configuring the Luna client.

Before completing the Luna HSM setup, run the vtl verify command (usually located at /usr/safenet/lunaclient/bin/vtl) to verify that the Luna HSM is properly configured.

After entering the Key HSM listener IP address and port, the HSM setup for Luna prompts for the slot number and password:

```
-- Configuring SafeNet Luna HSM --
Please enter SafeNetHSM Slot Number: 1
Please enter SafeNet HSM password (input suppressed):
Configuration stored in: 'application.properties'. (Note: You can also use s
ervice keyHsm settings to quickly view your current configuration)
Configuration saved in 'application.properties' file
```

See the Luna product documentation for instructions on configuring your Luna HSM if you do not know what values to enter here.

Validating Key HSM Settings

After you finish setting up Navigator Key HSM, you can check the configuration settings and verify that Key HSM is properly connected to your HSM.

About this task

After the setup completes, the Key HSM configuration is stored in /usr/share/keytrustee-server-keyhsm/application.properties.

You can view these settings using the service keyhsm settings command:

```
$ sudo service keyhsm settings

  # keyHsm Server Configuration information:
  keyhsm.management.address : 172.19.1.2
```

```
keyhsm.server.port : 9090
keyhsm.management.port : 9899
keyhsm.service.port : 19791
keyhsm.hardware : ncipher
# Module OCS Password
thales.ocs_password :
  GIqhXDuZsj1Oet137Lb+f+tqkYvKYDm/8StefpNqZWwlB+LfSYlB4eHd
  endtYJio8qLjjbT+e7j2th5xf8O9t8FwfVguuyFW+6wdD
  uNGvse1LY/itCwqF0ScMlB1Mnz4010xqC6ylPW7l+0JjjkkqqM5gJJbl8lsQFFaIGVM/pY=
```

These settings can be manually configured by modifying the application.properties file, with the exception of any passwords. These are encrypted by design, and can only be changed by re-running the setup utility.

Verifying Key HSM Connectivity to HSM

Procedure

1. To verify Hardware Security Module (HSM) operations using Key HSM, run the following command on the Key Trustee Server host (which should also be the Key HSM host as described in Installing Cloudera Navigator Key HSM):

   ```
   curl -k https://keytrustee01.example.com:11371/test_hsm
   ```

   If Key HSM operations to the HSM are successful, the command returns output similar to the following:

   ```
   "Sample Key TEST_HELLO_DEPOSIT2016-06-03-072718 has been created"
   ```

   You must run this command from the Key Trustee Server host. If you run it from a different host, the command returns an HTTP 403 error code.
2. If the command returns an HTTP 405 error code, restart Key Trustee Server and try again.

Note: If you are using the test_hsm script to verify that the Key Hardware Security Module (Key HSM) has successfully integrated with the Key Trustee Server, or to verify that the Key HSM is connected to HSM, and the Key Trustee Server private key file is password-protected, then the verification may fail. This can occur even if the integration is successful or connected.

If this occurs, create a key through Hadoop for the test.

Managing the Navigator Key HSM Service

Key HSM includes a command line tool for managing basic server operations. You can also manage logging, audits, and keys.

Using the keyhsm service

You can use the keyhsm service for basic server operations:

```
$  sudo service keyhsm
keyHsm service usage:
    setup <hsm name> - set up a new connection to an HSM
    trust <path>     - add a trusted client certificate
    validate         - validate that Key HSM is properly configured
    settings         - display the current server configuration
    start            - start the Key HSM proxy server
    status           - show the current Key HSM server status
    stop|shutdown    - force Key HSM server to shut down
```

```
    reload           - reload the server (without shutdown)
```

The reload command causes the application to restart internal services without ending the process itself. If you want to stop and start the process, use the restart command.

Logging and Audits

The Navigator Key HSM logs contain all log and audit information, which by default are stored in the /var/log/keyhsm directory.

You can configure the maximum log size (in bytes) and maximum number of log files to retain by adding or editing the following entries in the /usr/share/keytrustee-server-keyhsm/conf/logback.xml file.

```
<appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender
">
        <file>/var/log/keyhsm/keyhsm.log</file>

        <encoder>
            <pattern>%date %level %logger: %msg%n</pattern>
        </encoder>

        <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRolli
ngPolicy">
            <fileNamePattern>/var/log/keyhsm/keyhsm.log.%i</fileNamePattern>
            <minIndex>1</minIndex>
            <maxIndex>10</maxIndex>
        </rollingPolicy>
        <triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTrigg
eringPolicy">
            <maxFileSize>10MB</maxFileSize>
        </triggeringPolicy>
    </appender>

    <root level="info">
        <appender-ref ref="FILE" />
    </root>
```

The default log level is info. The filename is /var/log/keyhsm/keyhsm.log, the max file size is 10MB, and the last 10 rolled log files will be retained.

Warning:

Modifying the settings for items such as the log file size, log level, and number of rolled logs should not cause any issues. However, if more extensive changes are made to the logback.xml file (for example, changing the policy classes or log message format) and these changes introduce malformed XML or incorrect logback settings, then the Key HSM service may return an error and fail to start. Check the validity of the logback.xml file by running the command keyhsm after any updates. If there are errors in the formatting of logback.xml, they will appear in the command line:

```
$ keyhsm
...
06:55:20,208 |-ERROR in ch.qos.logback.core.joran.spi.Interpreter@11:85
 - no applicable action for [rollingPolicy], current ElementPath  is
 [[configuration][appender][rollingPolicy]]
06:55:20,208 |-ERROR in ch.qos.logback.core.joran.spi.Interpreter@12
:30 - no applicable action for [fileNamePattern], current ElementPath
 is [[configuration][appender][rollingPolicy][fileNamePattern]]
```

Address any errors before restarting Key HSM to pick up the logging changes. A copy of the default logback.xml file is provided at logback.xml.bkup. If there is an error in the updates, you can use this file to recover the logging configuration.

Key Naming Convention

To ensure you can manage keys (for example, delete a key), you must understand the naming convention for keys. Keys adhere to the following naming convention: handle name-uuid-date, which means if you know the key name and date, you can make modifications to it.

The following example shows the key nomenclature in the context of a key list query on Luna HSM:

```
[root@user 64]# cmu list
Please enter password for token in slot 1 : **********
handle=220
label=key1-3T17-YYdn-2015-07-23
handle=806
label=key2-CMYZ-8Sym-2015-07-23
handle=108
label=key3-qo62-XQfx-2015-07-23
handle=908
label=key2-CMYZ-8Sym-2015-07-23--cert0
handle=614
label=key3-qo62-RWz0-2015-07-23--cert0
handle=825
label=key1-3T11-YYdz-2015-07-23--cert0
```

Integrating Key HSM with Key Trustee Server

Using a hardware security module with Navigator Key Trustee Server requires Key HSM. This service functions as a driver to support interactions between Navigator Key Trustee Server and the hardware security module, and it must be installed on the same host system as Key Trustee Server.

Before you begin

- Prepare Existing Keys for Migration

  In this procedure, you are prompted to migrate any existing keys from the Key Trustee Server to the HSM. Successful migration depends on the existing keys conforming to the following constraints:

  

  Warning: Migration fails if any existing keys do not adhere to these constraints.

  - Key names can begin with alpha-numeric characters only
  - Key names can include only these special characters:
    - Hyphen -
    - Period .
    - Underscore _

  To prepare for migration, check your key names and do the following if any of them are non-conforming:

  - Decrypt any data using the non-conforming key.
  - Create a new key, named as desribed above.
  - Re-encrypt the data using the new key.

  

  Important: Keys are not available during migration, so you should perform these tasks during a maintenance window.

- Both Key HSM and Key Trustee Server must be set up and running. See Installing Cloudera Navigator Key HSM for details.

Procedure

1. Establish Trust from Key HSM to Key Trustee Server.

   This step assumes that Key Trustee Server has a certificate for TLS (wire) encryption as detailed in Managing Key Trustee Server Certificates. Key HSM service must explicitly trust the Key Trustee Server certificate (presented during TLS handshake). To establish this trust, run the following command:

   ```
   sudo keyhsm trust /path/to/key_trustee_server/cert
   ```

   The */path/to/key_trustee_server/cert* in this command (and in the commands below) depends on whether the Key Trustee Server uses the default certificate (created by default during install), or uses a custom certificate (obtained from a commercial or internal CA). The two alternate paths are shown in the table below. The custom path is a common example but may differ from that shown.

Default	Custom
/var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keytrustee.pem	/etc/pki/cloudera/certs/*cert-file.crt*
/var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keytrustee-pk.pem	/etc/pki/cloudera/private/*private-key.key*

   Note: The system requires TLS and Kerberos authentication throughout the system for security reasons. Connections attempted over SSL (1 through 3) and connections from untrusted clients are immediately terminated to prevent POODLE (Padding Oracle On Downgraded Legacy Encyryption) exploits. See the Cloudera Security Bulletin for more information.

2. Integrate Key HSM and Key Trustee Server.

   The following steps assume that both Key HSM and the Key Trustee Server are on the same host system, as detailed in Installing Cloudera Navigator Key HSM. These steps invoke commands on the Key HSM service and the Key Trustee Server, and they must be run on the host—they cannot be run remotely from another host.

   a. Ensure the Key HSM service is running:

      ```
      sudo service keyhsm start
      ```

   b. Establish trust from Key Trustee Server to Key HSM specifying the path to the private key and certificate (Key Trustee Server is a client to Key HSM). This example shows how to use the --client-certfile and --client-keyfile options to specify the path to non-default certificate and key:

      ```
      $ sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 \
      --client-certfile  /etc/pki/cloudera/certs/mycert.crt \
      --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
      ```

      For a password-protected Key Trustee Server private key, add the --passphrase argument to the command and enter the password when prompted:

      ```
      $ sudo ktadmin keyhsm --passphrase \
      --server https://keyhsm01.example.com:9090 \
      --client-certfile  /etc/pki/cloudera/certs/mycert.crt \
      --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
      ```

      Note: The preceding commands also create a certificate file for the Key HSM that is used by the Key Trustee Server. This certificate file is stored in /var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keyhsm.pem.

   c. Any keys that exist on the Key Trustee Server are automatically migrated when you run the ktadmin keyhsm command. To complete the migration, enter y or yes at the command prompt:

      ```
      Some deposits were found that will need to be moved to the HSM.
              Note that although this operation can be interrupted, once comp
      lete,
              items stored in the HSM must remain there!
      ```

```
Do you want to perform this migration now? [y/N]: y
Migrating hsm deposits...

Migration Complete!
```

d. Restart the Key Trustee Server:

- Using Cloudera Manager: Restart the Key Trustee Server service (Key Trustee Server service Actions Restart).
- Using the Command Line: Restart the Key Trustee Server daemon:
  - RHEL 6-compatible: $ sudo service keytrusteed restart
  - RHEL 7-compatible: $ sudo systemctl restart keytrusteed

e. Verify connectivity between the Key HSM service and the HSM:

```
curl -k https://keytrustee01.example.com:11371/test_hsm
```

Important: You must perform the connection verification between Key HSM and the HSM for all Key Trustee Server hosts.

Successful connection and test of operations returns output like the following:

```
"Sample Key TEST_HELLO_DEPOSIT2016-06-03-072718 has been created"
```

Note: If you are using the test_hsm script to verify that the Key Hardware Security Module (Key HSM) has successfully integrated with the Key Trustee Server, or to verify that the Key HSM is connected to HSM, and the Key Trustee Server private key file is password-protected, then the verification may fail. This can occur even if the integration is successful or connected.

If this occurs, then create a key through Hadoop for the test.

See Verifying Key HSM Connectivity to HSM for more information about the validation process.

Caution: There is no path for migrating keys from Key HSM back to the Key Trustee Server. Once Key Trustee Server is integrated with Key HSM, you cannot get the keys from HSM and remove Key HSM. This is because the underlying key metadata is changed on the HSM.

You can achieve this only by performing the following steps, which help you in removing all the keys and setting up the KMS from scratch:

1. Backup the data present in the encryption zone:
    a. Create a new non-encrypted directory in HDFS.
    b. Using `distcp`, copy the data from the encryption zone to the new HDFS directory.
2. Delete the keys created previously.
3. Disable encryption:
    a. Stop Key HSM.
    b. Stop the Key Trustee Server service.
    c. Take backup of the Key Trustee Server.
    d. Delete the Key Trustee Server service in Cloudera Manager (to completely erase its DB).
        1. Delete contents from /var/lib/keytrustee/db and /var/lib/keytrustee/.keytrustee directories.
        2. Keep the directory structures.
    e. Delete KMS service in Cloudera Manager.
        1. Delete contents from /var/lib/kms-keytrustee/keytrustee/.keytrustee directory.
        2. Keep the directory structure.
4. Reinstall Key Trustee Server.
5. Reinstall KMS.
6. Create new keys and encryption zone.
7. Restore the content of encryption.