CDP Private Cloud Base 7.0.3

Cloudera Navigator Key Trustee Server

Date published: 2020-11-30
Date modified: 2020-11-30

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

Cloudera Navigator Key Trustee Server Overview

An overview of Navigator Key Trustee Server and its architecture.

Key Trustee Server Overview

Cloudera Navigator Key Trustee Server is an enterprise-grade virtual safe-deposit box that stores and manages cryptographic keys and other security artifacts. With Navigator Key Trustee Server, encryption keys are separated from the encrypted data, ensuring that sensitive data is still protected if unauthorized users gain access to the storage media.

Key Trustee Server protects these keys and other critical security objects from unauthorized access while enabling compliance with strict data security regulations. For added security, Key Trustee Server can integrate with a hardware security module (HSM).

In conjunction with the Ranger KMS, Navigator Key Trustee Server can serve as a backing key store for HDFS transparent encryption, providing enhanced security and scalability over the file-based Java KeyStore used by the default Hadoop Key Management Server.

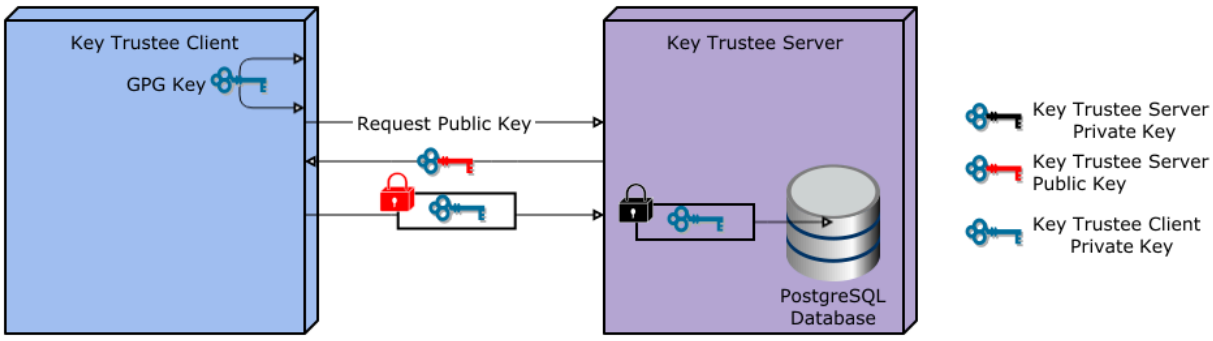
Cloudera Navigator Encrypt also uses Key Trustee Server for key storage and management.

Key Trustee Server Architecture

Key Trustee Server is a secure object store. Clients register with Key Trustee Server, and are then able to store and retrieve objects with Key Trustee Server. The most common use case for Key Trustee Server is storing encryption keys to simplify key management and enable compliance with various data security regulations, but Key Trustee Server is agnostic about the actual objects being stored.

All interactions with Key Trustee Server occur over a TLS-encrypted HTTPS connection.

Key Trustee Server does not generate encryption keys for clients. Clients generate encryption keys, encrypt them with their private key, and send them over a TLS-encrypted connection to the Key Trustee Server. When a client needs to decrypt data, it retrieves the appropriate encryption key from Key Trustee Server and caches it locally to improve performance. This process is demonstrated in the following diagram:



The most common Key Trustee Server clients are Navigator Encrypt and Key Trustee KMS.

When a Key Trustee client registers with Key Trustee Server, it generates a unique fingerprint. All client interactions with the Key Trustee Server are authenticated with this fingerprint. You must ensure that the file containing this fingerprint is secured with appropriate Linux file permissions. The file containing the fingerprint is /etc/navencrypt/keytrustee/ztrustee.conf for Navigator Encrypt clients, and /var/lib/kms-keytrustee/keytrustee/.keytrustee/keytrustee.conf for Key Trustee KMS.

Many clients can use the same Key Trustee Server to manage security objects. For example, you can have several Navigator Encrypt clients using a Key Trustee Server, and also use the same Key Trustee Server as the backing store for Key Trustee KMS (used in HDFS encryption).

Key Trustee Server System Requirements

Recommended Hardware and Supported Distributions

Key Trustee Server must be installed on a dedicated server or virtual machine (VM) that is not used for any other purpose. The backing PostgreSQL database must be installed on the same host as the Key Trustee Server, and must not be shared with any other services. For high availability, the active and passive Key Trustee Servers must not share physical resources.

The recommended minimum hardware specifications are as follows:

- Processor: 1 GHz 64-bit quad core
- Memory: 8 GB RAM
- Storage: 20 GB on moderate- to high-performance disk drives

Cloudera Manager Requirements

Installing and managing Key Trustee Server using Cloudera Manager requires Cloudera Manager 7.0.0 and higher. Key Trustee Server does not require Cloudera Navigator Audit Server or Metadata Server.

umask Requirements

Key Trustee Server installation requires the default umask of 0022.

Network Requirements

For new Key Trustee Server installations (5.4.0 and higher) and migrated upgrades, Key Trustee Server requires the following TCP ports to be opened for inbound traffic:

- 11371

  Clients connect to this port over HTTPS.
- 11381 (PostgreSQL)

  The passive Key Trustee Server connects to this port for database replication.

For upgrades that are not migrated to the CherryPy web server, the pre-upgrade port settings are preserved:

- 80

  Clients connect to this port over HTTP to obtain the Key Trustee Server public key.
- 443 (HTTPS)

  Clients connect to this port over HTTPS.
- 5432 (PostgreSQL)

  The passive Key Trustee Server connects to this port for database replication.

TLS Certificate Requirements

To ensure secure network traffic, Cloudera recommends obtaining Transport Layer Security (TLS) certificates specific to the hostname of your Key Trustee Server. To obtain the certificate, generate a Certificate Signing Request (CSR) for the fully qualified domain name (FQDN) of the Key Trustee Server host. The CSR must be signed by a trusted Certificate Authority (CA). After the certificate has been verified and signed by the CA, the Key Trustee Server TLS configuration requires:

- The CA-signed certificate
- The private key used to generate the original CSR
- The intermediate certificate/chain file (provided by the CA)

Cloudera recommends not using self-signed certificates. If you use self-signed certificates, you must use the --skip-ssl-check parameter when registering Navigator Encrypt with the Key Trustee Server. This skips TLS hostname validation, which safeguards against certain network-level attacks.

Cloudera Navigator Key Trustee Server

Cloudera Navigator Key Trustee Server is an enterprise-grade cryptographic key storage and management system that can be used by cluster components.

Important: Cloudera recommends that each cluster use its own KTS instance. Although sharing a single KTS across clusters is technically possible, it is neither approved nor supported for security reasons—specifically, the increased security risks associated with single point of failure for encryption keys used by multiple clusters.

After installing Key Trustee KMS, follow the steps in this guide to manage the system.

Backing up Key Trustee Server and clients

In case of failure, you should regularly back up Key Trustee Server databases and configuration files. You must also back up client configuration files and keys for Key Trustee Server clients, such as Key Trustee KMS and Navigator Encrypt clients.

Key Trustee Server high availability applies to read operations only. If either Key Trustee Server fails, the client automatically retries fetching keys from the functioning server. New write operations (for example, creating new encryption keys) are not allowed unless both Key Trustee Servers are operational.

Cloudera strongly recommends regularly backing up Key Trustee Server databases and configuration files. Because these backups contain encryption keys and encrypted deposits, you must ensure that your backup repository is as secure as the Key Trustee Server.

You must also back up client configuration files and keys for Key Trustee Server clients, such as Key Trustee KMS and Navigator Encrypt clients.

Note: In an HA configuration, the backup need only be performed on one of the hosts for Key Trustee Server and the Key Trustee KMS. For Key Trustee Server, run the backup on the active server. For Key Trustee KMS, you can run the backup on any instance.

Back up Key Trustee Server and Key Trustee KMS using Cloudera Manager

Cloudera Manager versions 5.8 and higher, when used with Key Trustee Server and Key Trustee KMS versions 5.7 and higher, allow for backups of the KT Server and KT KMS configurations.

The actions executed in this procedure are equivalent to running the ktbackup.sh script on the node in question (see Back up Key Trustee Server and Key Trustee KMS using the ktbackup.sh script on page 7 for additional details.

In addition, when using the HDFS Encryption Wizard in Cloudera Manager 5.8 or higher to install and configure Key Trustee Server and Key Trustee KMS versions 5.7 and higher, a cron job is automatically set up to back up the Key Trustee Server on an ongoing basis. See Initializing Standalone Key Trustee Server on page 16 for more detail.

To back up the KT Server or KT KMS service configuration using Cloudera Manager:

1. Select the KT Server or KMS service configuration that you wish to back up.
2. For a KT Server backup, select Create Backup on Active Server (or Create Backup on Passive Server) from the Actions menu. For a KMS backup, select Create Backup.

A successfully completed backup of the KT Server is indicated by the message “Command Create Backup on Active Server finished successfully on service keytrustee_server”.

Back up Key Trustee Server and Key Trustee KMS using the ktbackup.sh script

Key Trustee Server releases 5.7 and higher include a script, ktbackup.sh, to simplify and automate backing up Key Trustee Server. Key Trustee KMS releases 5.7 and higher include the same script for backing up Key Trustee KMS.

When run on a Key Trustee Server host, the script creates a tarball containing the Key Trustee Server private GPG keys and the PostgreSQL database. When run on a Key Trustee KMS host, the script creates a tarball containing the Key Trustee KMS private GPG keys and configuration file.

To preserve the security of the backup, you must specify a GPG recipient. Because this recipient is the only entity that can decrypt the backup, the recipient must be someone authorized to access the Key Trustee Server database, such as a key administrator.

Creating and Importing a GPG Key for Encrypting and Decrypting Backups

If the key administrator responsible for backing up and restoring Key Trustee Server and Key Trustee KMS does not already have a GPG key pair, they can create one using the gpg --gen-key command. The following example demonstrates this procedure:

Note: By default, gpg --gen-key fails at the password prompt if you have logged in to your user account with the su command. You must log in to the SSH session with the user account for which you want to generate the GPG key pair.

```
[john.doe@backup-host ~]$ gpg --gen-key
gpg (GnuPG) 2.0.14; Copyright (C) 2009 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)
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)
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: John Doe
Email address: john.doe@example.com
Comment: Key Trustee Backup
You selected this USER-ID:
```

```
    "John Doe (Key Trustee Backup) <john.doe@example.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.
can't connect to `/home/john.doe/.gnupg/S.gpg-agent': No such file or direc
tory
gpg-agent[10638]: directory `/home/john.doe/.gnupg/private-keys-v1.d' creat
ed
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: /home/john.doe/.gnupg/trustdb.gpg: trustdb created
gpg: key 0936CB67 marked as ultimately trusted
public and secret key created and signed.
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   2048R/0936CB67 2016-02-10
      Key fingerprint = CE57 FDED 3AFE E67D 2041  9EBF E64B 7D00 0936 CB67
uid                  John Doe (Key Trustee Backup) <john.doe@example.com>
sub   2048R/52A6FC5C 2016-02-10
```

After the GPG key pair is generated, you can export the public key:

```
[john.doe@backup-host ~]$ gpg --armor --output /path/to/johndoe.pub --export
 'John Doe'
```

Copy the public key (johndoe.pub in this example) to the Key Trustee Server or Key Trustee KMS host, and import it into the service account keyring (keytrustee for Key Trustee Server and kms for Key Trustee KMS):

- On the Key Trustee Server host:

  ```
  sudo -u keytrustee gpg --import /path/to/johndoe.pub
  ```

- On the Key Trustee KMS host:

  ```
  sudo -u kms gpg --import /path/to/johndoe.pub
  ```

Running the ktbackup.sh Script

You must run ktbackup.sh as the service account. The location of the script depends on the service and installation method. See the following table for the script location and default service account for package- and parcel-based installations for Key Trustee Server and Key Trustee KMS.

Table 1: Backup Script Locations

Service	Service Account	Parcel-Based Installation	Package-Based Installation
Key Trustee Server	keytrustee	/opt/cloudera/parcels/KEYTRUSTEE_SERVER/bin/ktbackup.sh	/usr/bin/ktbackup.sh
Key Trustee KMS	kms	/opt/cloudera/parcels/KEYTRUSTEE/bin/ktbackup.sh	/usr/share/keytrustee-keyprovider/bin/ktbackup.sh

The following table lists the command options for ktbackup.sh.

Table 2: Command Options for ktbackup.sh

Command Option	Description
-c, --confdir=CONFDIR	Specifies the Key Trustee configuration directory. Defaults to /var/lib /keytrustee/.keytrustee for parcel-based Key Trustee Server. For Key Trustee KMS and package-based Key Trustee Server, you must specify this option.
--database-port=PORT	Specifies the Key Trustee Server database port. Defaults to 11381 for parcel-based installations. For package-based Key Trustee Server installations, you must specify this option.
--gpg-recipient=GPG_RECIPIENT	Specifies the GPG recipient. The backup is encrypted with the public key of the specified recipient. The GPG recipient public key must be imported into the service account keyring before running the script. See Creating and Importing a GPG Key for Encrypting and Decrypting Backups on page 7 for more information.
--cleartext	Outputs an unencrypted tarball. To preserve the security of the cryptographic keys, do not use this option in production environments.
--output=DIR	Specifies the output directory for the tarball. Defaults to /var/lib/key trustee for parcel-based Key Trustee Server. For Key Trustee KMS and package-based Key Trustee Server, you must specify this option.
--roll=*n*	Deletes backups older than the last *n* backups from the directory specified by the --output parameter. For example, if you have 10 backups, specifying --roll=10 creates a new backup (11 backups total) and then delete the oldest backup. Specifying --roll=1 creates a new backup and then deletes all other backups.  **Note:** This option works for Key Trustee Server only.
-q, --quiet	Suppresses console log messages and, if successful, returns only the backup tarball file path. This is useful for automating backups.
--verbose	Outputs additional log messages to the console for debugging.

Important: Running the ktbackup.sh script on the KMS server creates a backed up copy of the GPG private keys currently being used by the KMS. If this backup is not performed, and the GPG private keys are not saved, then the system cannot be fully restored in the event of a catastrophic failure, and access to both keys and data may be lost. It is therefore imperative that you perform a manual backup after the KMS starts successfully for the first time, and on all KMS instances whenever a new Key Trustee High Availability instance is added. Running this backup ensures that the most recent backup contains a copy of the GPG private keys currently in use.

The following examples demonstrate the command usage for different scenarios:

- To back up a parcel-based Key Trustee Server, specifying the GPG recipient by name:

```
$ sudo -u keytrustee /opt/cloudera/parcels/KEYTRUSTEE_SERVER/bin/ktbacku
p.sh --gpg-recipient='John Doe'
```

- To back up a parcel-based Key Trustee KMS, specifying the GPG recipient by email:

```
$ sudo -u kms /opt/cloudera/parcels/KEYTRUSTEE/bin/ktbackup.sh -c /var/l
ib/kms-keytrustee/keytrustee/.keytrustee --output=/var/lib/kms-keytrustee
 --gpg-recipient=john.doe@example.com
```

- To back up a package-based Key Trustee Server with the database running on a non-default port (12345 in this example):

```
$ sudo -u keytrustee ktbackup.sh --database-port=12345 --gpg-recipient=j
ohn.doe@example.com
```

- To back up a package-based Key Trustee KMS, specifying the GPG recipient by email:

```
$ sudo -u kms /usr/share/keytrustee-keyprovider/bin/ktbackup.sh -c /var/
lib/kms-keytrustee/keytrustee/.keytrustee --output=/var/lib/kms-keytrustee
 --gpg-recipient=john.doe@example.com
```

Automating Backups Using cron

You can schedule automatic backups of Key Trustee Server using the cron scheduling utility.

Create a crontab entry using the following commands:

1. Edit the crontab by running the following command:

```
sudo -u keytrustee crontab -e
```

2. Add the following entry to run the backup script every 30 minutes. This example is for a parcel-based installation of Key Trustee Server. See the Backup Script Locations table for the package-based script location.

```
*/30 * * * * /opt/cloudera/parcels/KEYTRUSTEE_SERVER/bin/ktbackup.sh --g
pg-recipient='John Doe' --quiet --output=/tmp/backups --roll=10
```

Run man 5 crontab to see the crontab man page for details on using cron to schedule backups at different intervals.

Back up Key Trustee Server manually

Use this procedure for both parcel-based and package-based installations.

The following procedure references the default database port and location; if you modified these settings during installation, replace the database and port with your values.

1. Back up the Key Trustee Server database:
   - For Key Trustee Server 3.8:

     ```
     su - postgres
     pg_dump -c -p 5432 keytrustee | zip --encrypt keytrustee-db.zip -
     ```

   - For Key Trustee Server 5.4 and higher:

     ```
     su - keytrustee
     pg_dump -c -p 11381 keytrustee | zip --encrypt keytrustee-db.zip -
     ```

   The --encrypt option prompts you to create a password used to encrypt the zip file. This password is required to decrypt the file.

   For parcel-based installations, you must set environment variables after switching to the keytrustee user:

   ```
   su - keytrustee
   export PATH=$PATH:/opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/post
   gres/9.3/bin
   export LD_LIBRARY_PATH=/opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/
   postgres/9.3/lib
   pg_dump -c -p 11381 keytrustee | zip --encrypt keytrustee-db.zip -
   ```

2. Back up the Key Trustee Server configuration directory (/var/lib/keytrustee/.keytrustee):

   ```
   zip -r --encrypt keytrustee-conf.zip /var/lib/keytrustee/.keytrustee
   ```

   The --encrypt option prompts you to create a password used to encrypt the zip file. This password is required to decrypt the file.

3. Move the backup files (*keytrustee-db.zip* and *keytrustee-conf.zip*) to a secure location.

Back up Key Trustee Server clients

Cryptographic keys stored in Key Trustee Server are encrypted by clients before they are sent to Key Trustee Server. The primary clients for Key Trustee Server are Key Trustee KMS and Navigator Encrypt. Cloudera strongly recommends backing up regularly the configuration files and GPG keys for Key Trustee Server clients. See Back up Key Trustee Server and Key Trustee KMS using the ktbackup.sh script on page 7 for instructions on backing up Key Trustee KMS using the provided backup script.

Warning: Failure to back up these files can result in irretrievable data loss. For example, encryption zone keys used for HDFS Transparent Encryption are encrypted by the KMS before being stored in Key Trustee Server. A catastrophic failure of the KMS with no backup causes all HDFS data stored in encryption zones to become permanently irretrievable.

To prevent permanent data loss, regularly back up the following directories on each client that stores objects in Key Trustee Server:

Table 3: Key Trustee Server Client Configuration Directories

Key Trustee Server Client	Directories to Back Up
Key Trustee KMS	/var/lib/kms-keytrustee
Navigator Encrypt	/etc/navencrypt

Restoring Key Trustee Server

Restore Key Trustee Server in parcel-based installations

Note: These instructions apply to Key Trustee Servers deployed using parcels. For package-based deployments, skip to the Restore Key Trustee Server in package-based installations on page 12 section.

The following procedures assume the default database port and location; if you modified these settings during installation, replace the database and port with your custom values.

If the Key Trustee Server host has failed completely, remove the host from the cluster and add a new host using Cloudera Manager:

1. Remove the failed host from the cluster.
2. Add a new host with the same hostname and IP address as the failed host to the cluster.

   Important: Make sure that the replacement host uses the same operating system version as the failed host.

3. Install Key Trustee Server on the new host. Make sure to install the same Key Trustee Server version as the failed host.

After you have provisioned a new host and installed Key Trustee Server (or if you are restoring the database or configuration on the original host), restore the database and configuration directory. If your backups were created using the ktbackup.sh script, skip to Restore Key Trustee Server and Key Trustee KMS from ktbackup.sh backups on page 13. If you need to restore the Active Key Trustee Server from the Passive Key Trustee Server, skip to Restore Active Key Trustee Server from Passive Key Trustee Server on page 15.

If your backups were created manually using the pg_dump command, do the following:

1. Copy or move the backup files (*keytrustee-db.zip* and *keytrustee-conf.zip*) to the Key Trustee Server host.

2. Start the PostgreSQL server:

```
sudo ktadmin db --start --pg-rootdir /var/lib/keytrustee/db --background
```

3. Restore the Key Trustee Server database:

```
su - keytrustee
export PATH=$PATH:/opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/post
gres/9.3/bin
export LD_LIBRARY_PATH=/opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/
postgres/9.3/lib
unzip -p /path/to/keytrustee-db.zip | psql -p 11381 -d keytrustee
```

If the zip file is encrypted, you are prompted for the password to decrypt the file.

4. Restore the Key Trustee Server configuration directory:

```
su - keytrustee
cd /var/lib/keytrustee
unzip /path/to/keytrustee-conf.zip
```

If the zip file is encrypted, you are prompted for the password to decrypt the file.

5. Stop the PostgreSQL server:

```
sudo ktadmin db --stop --pg-rootdir /var/lib/keytrustee/db
```

6. Start the Key Trustee Server service in Cloudera Manager (Key Trustee Server serviceActionsStart).
7. Restart the Key Trustee KMS service in Cloudera Manager (Key Trustee KMS serviceActionsRestart).
8. Remove the backup files (*keytrustee-db.zip* and *keytrustee-conf.zip*) from the Key Trustee Server host.

Restore Key Trustee Server in package-based installations

The following procedures assume the default database port and location; if you modified these settings during installation, replace the database and port with your custom values.

If the Key Trustee Server host has failed completely, provision a new host with the same hostname and IP address as the failed host, and re-install Key Trustee Server.

Important: Make sure to install the same operating system and Key Trustee Server versions as the failed host.

After you have provisioned a new host and installed Key Trustee Server (or if you are restoring the database or configuration on the original host), restore the database and configuration directory. If your backups were created using the ktbackup.sh script, skip to Restore Key Trustee Server and Key Trustee KMS from ktbackup.sh backups on page 13. If you need to restore the Active Key Trustee Server from the Passive Key Trustee Server, skip to Restore Active Key Trustee Server from Passive Key Trustee Server on page 15.

If your backups were created manually using the pg_dump command, do the following:

1. Copy or move the backup files (*keytrustee-db.zip* and *keytrustee-conf.zip*) to the Key Trustee Server host.
2. Change the file ownership on the backup files to keytrustee:keytrustee:

```
sudo chown keytrustee:keytrustee /path/to/keytrustee*.zip
```

3. Restore the Key Trustee Server database:

```
su - keytrustee
unzip -p /path/to/keytrustee-db.zip | psql -p 11381 -d keytrustee
```

If the zip file is encrypted, you are prompted for the password to decrypt the file.

4. Restore the Key Trustee Server configuration directory:

```
cd /var/lib/keytrustee
unzip /path/to/keytrustee-conf.zip
```

   If the zip file is encrypted, you are prompted for the password to decrypt the file.
5. Start the Key Trustee Server service:
   - RHEL 6-compatible: $ sudo service keytrusteed start
   - RHEL 7-compatible: $ sudo systemctl start keytrusteed
6. Remove the backup files (*keytrustee-db.zip* and *keytrustee-conf.zip*) from the Key Trustee Server host.

Restore Key Trustee Server and Key Trustee KMS from ktbackup.sh backups

After installing Key Trustee Server or Key Trustee KMS on a new host after a failure, or if you need to restore accidentally deleted keys on the same host, use the following procedure to restore Key Trustee Server or Key Trustee KMS from backups generated by the ktbackup.sh script.

1. Decrypt the backup tarball using the private key of the GPG recipient specified in the backup command by running the following command as the GPG recipient user account. The GPG recipient private key must be available on the Key Trustee Server or Key Trustee KMS host on which you run this command.

```
gpg -d -o /path/to/decrypted/backup.tar /path/to/encrypted/tarball
```

2. Verify the decrypted tarball using the tar tvf */path/to/decrypted/backup.tar* command. For example:

```
$ tar tvf kts_bak_kts01_example_com_2016-02-10_11-14-37.tar
drwx------ keytrustee/keytrustee 0 2016-02-09 16:43 var/lib/keytrustee/.
keytrustee/
-rw------- keytrustee/keytrustee 434 2016-02-09 16:43 var/lib/keytrustee
/.keytrustee/keytrustee.conf
-rw------- keytrustee/keytrustee 1280 2016-02-09 16:43 var/lib/keytrust
ee/.keytrustee/trustdb.gpg
-rw------- keytrustee/keytrustee 4845 2016-02-09 16:43 var/lib/keytrustee
/.keytrustee/secring.gpg
-rw------- keytrustee/keytrustee  600 2016-02-09 16:43 var/lib/keytrust
ee/.keytrustee/random_seed
drwx------ keytrustee/keytrustee    0 2016-02-09 16:40 var/lib/keytrustee
/.keytrustee/.ssl/
-rw------- keytrustee/keytrustee 1708 2016-02-09 16:40 var/lib/keytrustee
/.keytrustee/.ssl/ssl-cert-keytrustee-pk.pem
-rw------- keytrustee/keytrustee 1277 2016-02-09 16:40 var/lib/keytrust
ee/.keytrustee/.ssl/ssl-cert-keytrustee.pem
-rw------- keytrustee/keytrustee 2263 2016-02-09 16:43 var/lib/keytruste
e/.keytrustee/pubring.gpg
-rw-r--r-- keytrustee/keytrustee  457 2016-02-09 16:43 var/lib/keytrustee/
.keytrustee/logging.conf
-rw------- keytrustee/keytrustee 2263 2016-02-09 16:43 var/lib/keytrust
ee/.keytrustee/pubring.gpg~
-rw------- keytrustee/keytrustee  157 2016-02-09 16:40 var/lib/keytruste
e/.keytrustee/gpg.conf
-rw-r--r-- keytrustee/keytrustee 47752 2016-02-10 11:14 var/lib/keytruste
e/kts_bak_kts01_example_com_2016-02-10_11-14-37.sql
```

3. Restore the files to their original locations, using this command for both Key Trustee Server and Key Trustee KMS backups:

```
tar xvf /path/to/decrypted/backup.tar -C /
```

4. (Key Trustee Server Only) Drop and re-create the keytrustee PostgreSQL database, and restore the database from the backup.

   - For parcel-based installations:

```
$ su - keytrustee
$ source /opt/cloudera/parcels/KEYTRUSTEE_SERVER/meta/keytrustee_env.sh
$ /opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/9.3/bin/p
sql -p 11381
psql (9.3.6)
Type "help" for help.
keytrustee=# \list
                                  List of databases
    Name    |   Owner    | Encoding |   Collate   |    Ctype    |
 Access privileges
------------+------------+----------+-------------+-------------+----
-----------------------
 keytrustee | keytrustee | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres   | keytrustee | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0  | keytrustee | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/key
trustee           +
            |            |          |             |             | keytr
ustee=CTc/keytrustee
 template1  | keytrustee | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/ke
ytrustee          +
            |            |          |             |             | keyt
rustee=CTc/keytrustee
(4 rows)

keytrustee=# \c postgres;
You are now connected to database "postgres" as user "keytrustee".
postgres=# drop database keytrustee;
DROP DATABASE
postgres=# create database keytrustee;
CREATE DATABASE
postgres=# \q
$ sudo -u keytrustee /opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG
_DB/opt/postgres/9.3/bin/psql -p 11381 -f /var/lib/keytrustee/
kts_bak_kts01_example_com_2016-02-10_11-14-37.sql
```

   - For package-based installations:

```
$ su - keytrustee
$ psql -p 11381
psql (9.3.6)
Type "help" for help.

keytrustee=# \list
                                  List of databases
    Name    |   Owner    | Encoding |   Collate   |    Ctype    |     Ac
cess privileges
------------+------------+----------+-------------+-------------+-------
--------------------
 keytrustee | keytrustee | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres   | keytrustee | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0  | keytrustee | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/ke
ytrustee          +
            |            |          |             |             | keyt
rustee=CTc/keytrustee
 template1  | keytrustee | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/k
eytrustee          +
            |            |          |             |             | key
trustee=CTc/keytrustee
```

```
(4 rows)
keytrustee=# \c postgres;
You are now connected to database "postgres" as user "keytrustee".
postgres=# drop database keytrustee;
DROP DATABASE
postgres=# create database keytrustee;
CREATE DATABASE
postgres=# \q
$ sudo -u keytrustee psql -p 11381 -f /var/lib/keytrustee/
kts_bak_kts01_example_com_2016-02-10_11-14-37.sql
```

5. Restart Key Trustee Server.
   - Using Cloudera Manager: Key Trustee Server serviceActionsRestart
   - Using the Command Line: Run the following command on the Key Trustee Server hosts:

     ```
     sudo service keytrusteed restart      #RHEL 6-compatible
     sudo systemctl restart keytrusteed    #RHEL 7-compatible
     ```
6. Restart the Key Trustee KMS service in Cloudera Manager (Key Trustee KMS serviceActionsRestart).

Restore Active Key Trustee Server from Passive Key Trustee Server

If the Active Key Trustee Server fails, and you do not have a backup, you can restore it from the Passive Key Trustee Server using the following procedure. You can also use this procedure if you need to restore keys that were successfully written to the Passive Key Trustee Server, but are not included in the most recent backup.

The following procedure assumes you have installed Key Trustee Server on the replacement host and (if you are using Cloudera Manager) added the Key Trustee Server service.

1. Copy the Key Trustee Server database from the Passive Key Trustee Server host to the new Active Key Trustee Server host. Run the following command on the Passive Key Trustee Server host:

   ```
   sudo rsync --exclude recovery.conf -a /var/lib/keytrustee/db roo
   t@kts01.example.com:/var/lib/keytrustee/
   ```

   Replace *kts01.example.com* with the hostname of the new Active Key Trustee Server.
2. Make sure that the recovery.conf file did not get copied to the Active Key Trustee Server (for example, if there was a typo in your rsync command). Run the following command on the Active Key Trustee Server host:

   ```
   sudo ls -l /var/lib/keytrustee/db/recovery.conf
   ```

   If the file exists on the Active Key Trustee Server host, delete it. Make sure you are on the Active Key Trustee Server host before deleting the file. Do not delete the recovery.conf file on the Passive Key Trustee Server host.
3. Copy the configuration directory from the Passive Key Trustee Server host to the new Active Key Trustee Server host. Run the following command on the Passive Key Trustee Server host:

   ```
   sudo rsync --exclude .ssl --exclude '*.pid' -a /var/lib/keytrustee/.keyt
   rustee root@kts01.example.com:/var/lib/keytrustee/
   ```

   Replace *kts01.example.com* with the hostname of the new Active Key Trustee Server.
4. Create the logs directory and make sure it is owned by the keytrustee user and group:

   ```
   sudo mkdir /var/lib/keytrustee/logs
   sudo chown keytrustee:keytrustee /var/lib/keytrustee/logs
   ```
5. (Cloudera Manager only) Generate the Key Trustee Server keyring: Key Trustee Server serviceActionsGenerate Key Trustee Server Keyring

6. Set up the database on the Active Key Trustee Server host.
   - Using Cloudera Manager: Key Trustee Server serviceActionsSet Up Key Trustee Server Database
   - Using the Command Line:

     ```
     sudo ktadmin --confdir /var/lib/keytrustee db --port 11381 --pg-rootdir
     /var/lib/keytrustee/db --bootstrap --slave kts02.example.com
     ```

     Replace *kts02.example.com* with the hostname of the Passive Key Trustee Server.
7. Start the database.
   - Using Cloudera Manager: Key Trustee Server serviceInstancesActive DatabaseActionsStart this Active Database
   - Using the Command Line: Run the following command on the Active Key Trustee Server host:

     ```
     sudo ktadmin --confdir /var/lib/keytrustee db --port 11381 --pg-rootdir
     /var/lib/keytrustee/db --bootstrap --slave kts02.example.com
     ```

     Replace *kts02.example.com* with the hostname of the Passive Key Trustee Server.
8. Enable synchronous replication.
   - Using Cloudera Manager: Key Trustee Server serviceActionsSetup Enable Synchronous Replication in HA mode
   - Using the Command Line: Run the following command on the Active Key Trustee Server host:

     ```
     sudo ktadmin --confdir /var/lib/keytrustee enable-synchronous-replic
     ation
     ```
9. Restart the Active Key Trustee Server.
   - Using Cloudera Manager: Key Trustee Server serviceActionsRestart
   - Using the Command Line: Run the following command on the Active Key Trustee Server host:

     ```
     sudo service keytrusteed restart      #RHEL 6-compatible
     sudo systemctl restart keytrusteed    #RHEL 7-compatible
     ```
10. Restart the Key Trustee KMS service in Cloudera Manager (Key Trustee KMS serviceActionsRestart).

Restoring Navigator Key Trustee Server and Key Trustee KMS

If a Key Trustee Server fails catastrophically, you must restore it from backup to a new host with the same hostname and IP address as the failed host. Cloudera does not support PostgreSQL promotion to convert a passive Key Trustee Server to an active Key Trustee Server.

When restoring the Key Trustee Server database from backup, keep in mind that any keys or deposits created after the backup are not restored. If you are using Key Trustee Server high availability, you can restore the Active Key Trustee Server from the Passive Key Trustee Server. This restores all keys that were successfully written to the Passive Key Trustee Server before the failure.

The procedure to restore Key Trustee Server is different for parcel-based than for package-based installations. For more information about parcels, see “Overview of Parcels”.

Initializing Standalone Key Trustee Server

Initializing Standalone Key Trustee Server Using Cloudera Manager

Important: If you are using SSH software other than OpenSSH, the initialization fails. To prevent this, pre-create the SSH key before continuing:

```
sudo -u keytrustee ssh-keygen -t rsa -f /var/lib/keytrustee/.ssh/id_rsa
```

For new installations, use the Set up HDFS Data At Rest Encryption wizard. When prompted, deselect the Enable High Availability option to proceed in standalone mode.

To set up Key Trustee Server manually, add the Key Trustee Server service to your cluster. When customizing role assignments, assign only the Active Key Trustee Server and Active Database roles.

Important: You must assign the Key Trustee Server and Database roles to the same host. Key Trustee Server does not support running the database on a different host.

For parcel-based Key Trustee Server releases 5.8 and higher, Cloudera Manager automatically backs up Key Trustee Server (using the ktbackup.sh script) after adding the Key Trustee Server service. It also schedules automatic backups using cron. For package-based installations, you must manually back up Key Trustee Server and configure a cron job.

Cloudera Manager configures cron to run the backup script hourly. The latest 10 backups are retained in /var/lib/keytrustee in cleartext. For information about using the backup script and configuring the cron job (including how to encrypt backups), see Back up Key Trustee Server and Key Trustee KMS using the ktbackup.sh script on page 7.

Specifying TLS/SSL Minimum Allowed Version and Ciphers

Depending on your cluster configuration and the security practices in your organization, you might need to restrict the allowed versions of TLS/SSL used by Key Trustee Server. Older TLS/SSL versions might have vulnerabilities or lack certain features.

Specify one of the following values using the Minimum TLS Support configuration setting:

- tlsv1: Allow any TLS version of 1.0 or higher. This setting is the default when TLS/SSL is enabled.
- tlsv1.1: Allow any TLS version of 1.1 or higher.
- tlsv1.2: Allow any TLS version of 1.2 or higher.

Note: The pyOpenSSL version on the Key Trustee Server cluster should be updated to 16.2 before changing the TLS version to 1.2. If pyOpenSSL is not updated, then the following error appears when the Key Trustee Server service attempts to restart:

```
keytrustee-server: Error in setting the protocol to the value TLSv1.2.
This usually means there is no support for the entered value.
Python Error: 'module' object has no attribute 'OP_NO_TLSv1_1"
```

Along with specifying the version, you can also specify the allowed set of TLS ciphers using the Supported Cipher Configuration for SSL configuration setting. The argument to this option is a list of keywords, separated by colons, commas, or spaces, and optionally including other notation.

```
AES256:CAMELLIA256-SHA
```

By default, the cipher list is empty, and Key Trustee Server uses the default cipher list for the underlying platform. See the output of `man ciphers` for the full set of keywords and notation allowed in the argument string.

Configuring a Mail Transfer Agent for Key Trustee Server

The Key Trustee Server requires a mail transfer agent (MTA) to send email. Cloudera recommends Postfix, but you can use any MTA that meets your needs.

To configure Postfix for local delivery, run the following commands:

```
export KEYTRUSTEE_SERVER_PK="/var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-k
eytrustee-pk.pem"
export KEYTRUSTEE_SERVER_CERT="/var/lib/keytrustee/.keytrustee/.ssl/ssl-ce
rt-keytrustee.pem"
export KEYTRUSTEE_SERVER_CA="/var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-
keytrustee-ca.pem"
export KEYTRUSTEE_SERVER_HOSTNAME="$(hostname -f)"  # or adjust as required
postconf -e 'mailbox_command ='
postconf -e 'smtpd_sasl_local_domain ='
postconf -e 'smtpd_sasl_auth_enable = yes'
postconf -e 'smtpd_sasl_security_options = noanonymous'
postconf -e 'broken_sasl_auth_clients = yes'
postconf -e 'smtpd_recipient_restrictions = permit_sasl_authenticated,permi
t_mynetworks,reject_unauth_destination'
postconf -e 'inet_interfaces = localhost'
postconf -e 'smtp_tls_security_level = may'
postconf -e 'smtpd_tls_security_level = may'
```

Start the Postfix service and ensure that it starts at boot:

```
service postfix restart
sudo chkconfig --level 235 postfix on
```

For information on installing Postfix or configuring a relay host, see the Postfix documentation.

Verifying Cloudera Navigator Key Trustee Server Operations

Verify that the installation was successful by running the following command on all Key Trustee Servers.

```
curl -k https://keytrustee.example.com:11371/?a=fingerprint
```

Replace keytrustee.example.com with the fully qualified domain name (FQDN) of each Key Trustee Server you are validating. This command outputs a fingerprint similar to the following:

```
4096R/4EDC46882386C827E20DEEA2D850ACA33BEDB0D1
```

If high availability is enabled, the output should be identical for all Key Trustee Servers.

Managing Key Trustee Server Organizations

Organizations allow you to configure Key Trustee for use in a multi-tenant environment. Using the keytrustee-orgtool utility, you can create organizations and administrators for multiple organizations. Organization administrators can then approve or deny the registration of clients, depending on the registration method.

The keytrustee-orgtool Utility

keytrustee-orgtool is a command-line utility for administering organizations. The keytrustee-orgtool command must be run as the root user.

The following table explains the various keytrustee-orgtool commands and parameters. Run keytrustee-orgtool --help to view this information at the command line.

Table 4: Usage for keytrustee-orgtool

Operation	Usage	Description
Add	keytrustee-orgtool add [-h] –n *name* –c *contacts*	Adds a new organization and administrators for the organization.
List	keytrustee-orgtool list	Lists current organizations, including the authorization secret, all administrators, the organization creation date, and the organization expiration date.
Disable client	keytrustee-orgtool disable-client [-h] –-fingerprint *fingerprint*	Disables a client that has already been activated by the organization administrator.
Enable client	keytrustee-orgtool enable-client [-h] –-fingerprint *fingerprint*	Enables a client that has requested activation but has not yet been approved by the organization administrator.
Set authorization Code	keytrustee-orgtool set-auth [-h] –n *name* –s *secret*	Sets the authorization code to a new string, or to blank to allow automatic approvals without the code.

Create Organizations

Each new Key Trustee tenant needs its own organization. You can create new organizations using the keytrustee-orgtool add command. For example, to create a new organization for the Disaster Recovery group and add two administrators, Finn and Jake:

```
keytrustee-orgtool add -n disaster-recov -c finn@example.com,jake@exampl
e.com
```

When adding organizations:

- Do not use spaces or special characters in the organization name. Use hyphens or underscores instead.
- Do not use spaces between email addresses (when adding multiple administrators to an organization). Use a comma to separate email addresses, without any space (as shown in the example above).

Each contact email address added when creating the organization receives a notification email, as detailed below.

Once an organization exists, use the keytrustee-orgtool add command to add new administrators to the organization. For example, to add an administrator to the disaster-recov organization:

```
keytrustee-orgtool add -n disaster-recov -c marceline@example.com
```

Note: You cannot remove contacts from an organization with the keytrustee-orgtool utility.

List Organizations

After creating an organization, verify its existence with the keytrustee-orgtool list command. This command lists details for all existing organizations. The following is the entry for the disaster-recov organization created in the example:

```
"disaster-recov": {
          "auth_secret": "/qFiICsyYqMLhdTznNY3Nw==",
```

```
            "contacts": [
             "finn@example.com",
             "jake@example.com"
        ],
        "creation": "2013-12-02T09:55:21",
        "expiration": "9999-12-31T15:59:59",
        "key_info": null,
        "name": "disaster-recov",
        "state": 0,
        "uuid": "xY3Z8xCwMuKZMiTYJa0mZOdhMVdxhyCUOc6vSNc9I8X"
    }
```

Change the Authorization Code

When an organization is created, an authorization code is automatically generated. When you run the keytrustee-orgtool list command, the code is displayed in the auth_secret field. To register with a Key Trustee Server, the client must have the authorization code along with the organization name. To set a new auth_secret, run the following command:

```
keytrustee-orgtool set-auth -n disaster-recov -s ThisISAs3cr3t!
```

Run the keytrustee-orgtool list command again, and confirm the updated auth_secret field:

```
"disaster-recov": {
          "auth_secret": "ThisISAs3cr3t!",
          "contacts": [
            "finn@example.com",
            "jake@example.com"
        ],
        "creation": "2013-12-02T09:55:21",
        "expiration": "9999-12-31T15:59:59",
        "key_info": null,
        "name": "disaster-recov",
        "state": 0,
        "uuid": "xY3Z8xCwMuKZMiTYJa0mZOdhMVdxhyCUOc6vSNc9I8X"
    }
```

If you do not want to use an authorization code, set the auth_secret field to an empty string:

```
keytrustee-orgtool set-auth -n disaster-recov -s ""
```

Cloudera recommends requiring an authorization code.

Notification Email and GPG Keys

Whenever an administrator is added to an organization, the Key Trustee Server sends an automated email message (subject: “KeyTrustee Contact Registration”) to the newly added administrator:

```

Hello, this is an automated message from your Cloudera keytrustee Server.

Welcome to Cloudera keytrustee! You have been listed as an administrator con
tact
for keytrustee services at your organization [test-org]. As an administrato
r,
you may be contacted to authorize the activation of new keytrustee clients.

We recommend that you register a GPG public key for secure administration of
your clients. To do so, visit the link below and follow the instructions.

https://keytrustee01.example.com:11371/?q=CnRV6u0nbm7zB07BQEpXCXsN0QJFBz684u
C0lcHMoWL
```

```
This link will expire in 12 hours, at Thu Sep  3 00:08:25 2015 UTC.
```

Cloudera recommends that an organization's administrators:

- Register the GPG public key by following the link contained in the notification email. Registering the GPG public key is optional, but if you choose to register your public key:
  - Complete the process within 12 hours, before the link expires.
  - Upload the entire key, including the BEGIN and END strings, as shown here:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.1 (GNU/Linux)
Comment: For info see http://www.gnupg.org

mQGiBDkHP3URBACkWGsYh43pkXU9wj/X1G67K8/DSrl85r7dNtHNfLL/ewil10k2
q8saWJn26QZPsDVqdUJMOdHfJ6kQTAt9NzQbgcVrxLYNfgeBsvkHF/POtnYcZRgL
tZ6syBBWs8JB4xt5V09iJSGAMPUQE8Jpdn2aRXPApdoDw179LM8Rq6r+gwCg5ZZa
. . .
-----END PGP PUBLIC KEY BLOCK-----
```

- Import the Key Trustee Server's public GPG key to verify that the server is the sender.

The organization's administrators are notified by email when new clients are registered to the Key Trustee Server using the mail transfer agent (as discussed in Configuring a Mail Transfer Agent for Key Trustee Server on page 18). However, if the server does not have access to email, you can use a local system mail address, such as user name@hostname, where hostname is the system hostname and username is a valid system user. If you use a local system email address, be sure to regularly monitor the email box.

Managing Key Trustee Server Certificates

Transport Layer Security (TLS) certificates are used to secure communication with Key Trustee Server. By default, Key Trustee Server generates self-signed certificates when it is first initialized. Cloudera strongly recommends using certificates signed by a trusted Certificate Authority (CA).

Generating a New Certificate

1. Generate a new certificate signing request (CSR):

   ```
   openssl req -new -key keytrustee_private_key.pem -out new.csr
   ```

   Replace *keytrustee_private_key.pem* with the filename of the private key. You can reuse the existing private key or generate a new private key in accordance with your company policies. For existing auto-generated self-signed certificates, the private key file is located at /var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keytrustee-pk.pem.
2. Generate a new certificate from the CSR:
   - For a CA-signed certificate, submit the CSR to the CA, and they will provide a signed certificate.
   - To generate a new self-signed certificate, run the following command:

   ```
   $ openssl x509 -req -days 365 -in new.csr -signke
   y keytrustee_private_key.pem \
   -out new_keytrustee_certificate.pem
   ```

Replacing Key Trustee Server Certificates

Use the following procedure if you need to replace an existing certificate for the Key Trustee Server. For example, you can use this procedure to replace the auto-generated self-signed certificate with a CA-signed certificate, or to replace an expired certificate.

Note: Key Trustee Server supports password-protected private keys, but not password-protected certificates.

1. After Generating a New Certificate on page 21, back up the original certificate and key files:

```
sudo cp -r /var/lib/keytrustee/.keytrustee/.ssl /var/lib/keytrustee/.key
trustee/.ssl.bak
```

2. (CA-Signed Certificates Only) Provide either the root or intermediate CA certificate:

Important: If you have separate root CA and intermediate CA certificate files, then you must concatenate them into a single file.

```
sudo mv /path/to/rootca.pem /var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-
keytrustee-ca.pem
```

3. Make sure that the certificate and key files are owned by the keytrustee user and group, with file permissions set to 600:

```
sudo chown keytrustee:keytrustee /path/to/new/certificate.pem /path/to/
new/private_key.pem
sudo chmod 600 /path/to/new/certificate.pem /path/to/new/private_key.pem
```

4. Update the Key Trustee Server configuration with the location of the new certificate and key files:
   - Using Cloudera Manager:
     a. Go to the Key Trustee Server service.
     b. Click the Configuration tab.
     c. Select CategorySecurity.
     d. Edit the following properties to specify the location of the new certificate and key files. If the private keys are not password protected, leave the password fields empty.
        - Active Key Trustee Server TLS/SSL Server Private Key File (PEM Format)
        - Active Key Trustee Server TLS/SSL Server Certificate File (PEM Format)
        - Active Key Trustee Server TLS/SSL Private Key Password
        - Passive Key Trustee Server TLS/SSL Server Private Key File (PEM Format)
        - Passive Key Trustee Server TLS/SSL Server Certificate File (PEM Format)
        - Passive Key Trustee Server TLS/SSL Private Key Password
     e. Enter a Reason for change, and then click Save Changes to commit the changes.
   - Using the command line:
     a. Edit /var/lib/keytrustee/.keytrustee/keytrustee.conf on the active and passive Key Trustee Server hosts and modify the SSL_CERTIFICATE and SSL_PRIVATE_KEY parameters as follows:

```
    "SSL_CERTIFICATE": "/path/to/new/certificate.pem",
```

```
    "SSL_PRIVATE_KEY": "/path/to/new/private_key.pem"
```

If the private key is password protected, add the following entry:

```
    "SSL_PRIVATE_KEY_PASSWORD_SCRIPT": "/path/to/password_script
 [arguments]"
```

Replace */path/to/password_script [arguments]* with the path to a script (and any necessary command arguments) that returns the password for the private key file. If you do not want to create a script, you can use a simple command, such as echo -n *password*. For example:

```
    "SSL_PRIVATE_KEY_PASSWORD_SCRIPT": "/bin/echo -n password"
```

Keep in mind that this method can expose the private key password in plain text to anyone who can view the /var/lib/keytrustee/.keytrustee/keytrustee.conf file.

5. Restart Key Trustee Server:
   - Using Cloudera Manager: Restart the Key Trustee Server service (Key Trustee Server serviceActionsRestart).
   - Using the Command Line: Restart the Key Trustee Server daemon:
     - RHEL 6-compatible: $ sudo service keytrusteed restart
     - RHEL 7-compatible: $ sudo systemctl restart keytrusteed
6. If you are using the Key Trustee KMS service in Cloudera Manager for HDFS Transparent Encryption, update the Java KeyStore (JKS) used on the Key Trustee KMS host:
   a. Download the new certificate to the Key Trustee KMS host:

   ```
   $ echo -n | openssl s_client -connect keytrustee01.example.com:11371 \
   | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /tmp/
   keytrustee_certificate.pem
   ```

   b. Delete the existing keystore entry for *keytrustee01.example.com*:

   ```
   keytool -delete -alias key_trustee_alias_name -keystore /path/to/
   truststore -v
   ```

   c. Add the new keystore entry for *keytrustee01.example.com*:

   ```
   $ keytool -import -trustcacerts -alias keytrustee01.example.com \
   -file /tmp/keytrustee_certificate.pem -keystore /path/to/truststore
   ```

   d. Restart the Key Trustee KMS service in Cloudera Manager.
7. If you are using Key HSM, update the Key Trustee Server and Key HSM configuration:
   a. Run the keyhsm trust command, using the path to the new certificate:

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

   b. Run the ktadmin keyhsm command, using the --client-certfile and --client-keyfile options to specify the location of the new certificate file and private key:

   ```
   sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 --client-
   certfile /path/to/new/key_trustee_server/cert --client-keyfile /path/to/
   new/key_trustee_server/private_key
   ```

Setting Up Key Trustee Server High Availability

Key Trustee Server high availability applies to read operations only. If either Key Trustee Server fails, the KeyProvider automatically retries fetching keys from the functioning server. New write operations (for example, creating new encryption keys) are not allowed unless both Key Trustee Servers are operational.

If a Key Trustee Server fails, the following operations are impacted:

- HDFS Encryption
  - You cannot create new encryption keys for encryption zones.
  - You can write to and read from existing encryption zones, but you cannot create new zones.
- Cloudera Navigator Encrypt
  - You cannot register new Cloudera Navigator Encrypt clients.
  - You can continue reading and writing encrypted data, including creating new mount points, using existing clients.

Cloudera recommends monitoring both Key Trustee Servers. If a Key Trustee Server fails catastrophically, restore it from backup to a new host with the same hostname and IP address as the failed host. Cloudera does not support PostgreSQL promotion to convert a passive Key Trustee Server to an active Key Trustee Server.

Depending on your cluster configuration and the security practices in your organization, you might need to restrict the allowed versions of TLS/SSL used by Key Trustee Server.

Configuring Key Trustee Server High Availability Using Cloudera Manager

About this task

Procedure

1. Pick one:
   a) For new installations, use the Set up HDFS Data At Rest Encryption wizard and follow the instructions in “Enabling HDFS Encryption Using the Wizard”. When prompted, make sure that the Enable High Availability option is selected.
   b) If you already have a Key Trustee Server service, and want to enable high availability, see “Adding a Role Instance” for the Key Trustee Server service instead to add the Passive Key Trustee Server and Passive Database roles.

   Note: You must assign the Key Trustee Server and Database roles to the same host. Assign the Active Key Trustee Server and Active Database roles to one host, and the Passive Key Trustee Server and Passive Database roles to a separate host.

   After completing the Add Role Instances wizard, the Passive Key Trustee Server and Passive Database roles fail to start. Complete the following manual actions to start these roles:
2. Stop the Key Trustee Server service (Key Trustee Server service Actions Stop).
3. Run the Set Up Key Trustee Server Database command (Key Trustee Server service Actions Set Up Key Trustee Server Database).
4. Run the following command on the Active Key Trustee Server:

   ```
   sudo rsync -zcav --exclude .ssl /var/lib/keytrustee/.keytrustee
   root@keytrustee02.example.com:/var/lib/keytrustee/.
   ```

   Replace *keytrustee02.example.com* with the hostname of the Passive Key Trustee Server.

5. Run the following command on the Passive Key Trustee Server:

```
sudo ktadmin init
```

6. Start the Key Trustee Server service (Key Trustee Server service Actions Start).

Important: Starting or restarting the Key Trustee Server service attempts to start the Active Database and Passive Database roles. If the Active Database is not running when the Passive Database attempts to start, the Passive Database fails to start. If this occurs, manually restart the Passive Database role after confirming that the Active Database role is running.

7. Enable synchronous replication (Key Trustee Server service Actions Setup Enable Synchronous Replication in HA mode).
8. Restart the Key Trustee Server service (Key Trustee Server service Actions Restart).

Recovering a Key Trustee Server

If a Key Trustee Server fails, restore it from backup as soon as possible. If the Key Trustee Server hosts fails completely, make sure that you restore the Key Trustee Server to a new host with the same hostname and IP address as the failed host.

For more information, see “Backing up Key Trustee Server and clients” and “Restoring Navigator Key Trustee Server”.