Linux Integration and Solutions Guide

FOR CTE V7.6.0

Installing CTE on Hadoop

This section describes how to protect an HDFS cluster with CTE. It contains the following topics:

- Overview
- Implementing CTE on HDFS
- Deploy Cloudera Hadoop on CTE
- HDFS Upgrade with CTE
- CTE Installation and Configuration
- Deleting Metadata in HDFS when Migrating Out of LDT

Overview

The Hadoop Distributed File System (HDFS) is a file system that supports large files and directory structures distributed across hundreds, or even thousands, of commodity DataNode hosts in a cluster. Previously, CTE could only protect directories and files on the *local file system* rather than the actual HDFS files and directories. Now, CTE can protect *HDFS* files and directories.

A CipherTrust Manager can:

- Define an encryption policy for HDFS files and directories in HDFS name space.
- Selectively encrypt HDFS folders with different keys providing multi-tenancy support.
- Define user-based I/O access control rules for HDFS files in HDFS name space.

At the heart of an HDFS cluster is the *NameNode* that provides the framework to support a traditional hierarchical file and directory organization. The NameNode is a master server that manages the HDFS name space and regulates access to files by clients.

HDFS files are split into one or more data blocks that are distributed across *DataNode* hosts in a cluster. The NameNode maintains the namespace tree and the mapping of data blocks to DataNodes. To deploy CTE, install CTE on all of the NameNode and DataNode hosts in a cluster.

Overview of CTE on HDFS

The following sections list the high-level steps for implementing CTE protection on your HDFS. The process requires that the HDFS Administrator and CipherTrust Manager to work in tandem to complete separate tasks.

You can keep the HDFS cluster alive a active if you enable HDFS data replication and activate the nodes individually. Following are the high-level steps:

HDFS Administrator

1. Compile a list of directories specified by `dfs.datanode.data.dir`. If these directories do not already exist in the NameNode local file system, create them.
2. Pass the directory list to the CipherTrust Manager.
3. Ask CipherTrust Manager to:
   a. Add the NameNode to the HDFS host/client group.
   b. Create a GuardPoint for the HDFS host/client group on each of these directories.

Administrator

1. Create an HDFS host/client group to contain the HDFS nodes.
2. Create a host/client group GuardPoint on each of the datanode directories obtained from the previous step.
3. Add the NameNode to the HDFS host/client group.

HDFS Administrator

1. For each DataNode, take the node offline and perform a data transformation.
2. Ask the Administrator to add the DataNode to the host group. (After the DataNode is added to the host group, it can be brought online.)
3. Repeat this process until all of the nodes have been encrypted and added to the HDFS Host Group.
4. Modify the host group policies to protect specific HDFS files and directories, as needed.

Implementing CTE on HDFS

This section describes how to implement CTE protection on your HDFS NameNode or DataNode. If you enable HDFS data replication, protecting one node at a time allows you to maintain the HDFS cluster. When all of the nodes are configured, you can create GuardPoints on specific HDFS files and directories.

These instructions require that the HDFS Administrator and Administrator work in tandem to complete separate tasks.

Note

The instructions below assume that each NameNode and DataNode exist on their own separate host. If you have a NameNode and DataNode on the same host, see Implementing CTE on HDFS on a Single Host.

CTE on HDFS Implementation Assumptions

- You have installed, configured and registered CTE on all of the NameNodes and DataNodes in the Hadoop cluster.
- The HDFS Administrator has knowledge and experience with HDFS and Ambari.
- The CipherTrust Manager Administrator has knowledge and experience with CTE.
- The two can work and communicate in tandem with each other.

Create an Encryption Zone in HDFS Name Space for AWS EMR

HDFS requires the following manual steps if you want to have an encryption zone in the HDFS name space for AWS EMR, (Elastic MapReduce).

1. Add the following properties to `hdfs-site.xml`.

   dfs.vte.ioctl.lib vorhdfs dfs.vte.rename.check true dfs.vte.ioctl.device <.sec folder name, up to the CTE installation location>

> **Note**
>
> The default .sec folder is `/opt/vormetric/DataSecurityExpert/agent/secfs/.sec`

2. Save the file.

3. Restart HDFS NameNode and DataNode services.

Using the Original Information from HDFS

Update or add the following properties to `hdfs-site.xml` if you want CTE to use the original user information from HDFS.

1. Add the following properties to `hdfs-site.xml`.

```
<property>
    <name>dfs.block.access.token.enable</name>
    <value>true</value>
</property>
<property>
    <name>dfs.client.read.shortcircuit</name>
    <value>false</value>
</property>
<property>
    <name>dfs.vte.user.push</name>
    <value>true</value>
</property>
```

2. Save the file.

3. Restart HDFS NameNode and DataNode services.

Create an HDFS Host Group and GuardPoint in CipherTrust Manager

After configuring the NameNodes, the next steps in activating CTE on HDFS is for the security administrator. To create a client group:

1. Open the **Transparent Encryption** application.
2. Click **Clients > Client Groups**.
3. Click **Create Client Group**. The Create Client Group dialog box displays.
4. Enter a unique **Name** for the client group.
5. Select the **Cluster Type**: HDFS
    a. **NON CLUSTER**: Creates a non-clustered client group.
    b. **HDFS**: Creates a clustered client group. An HDFS client group is required to apply GuardPoints on CTE clients in an HDFS cluster.
6. (Optional, displayed if a profile already exists) From the Client Profile drop-down list, select the desired client profile. The default profile is DefaultClientProfile.
7. (Optional) Provide Description to identify the client group. The maximum length can be 256 characters.
8. Click **Create**. The client group is created.
9. Add the NameNode obtained from the HDFS Admin to the HDFS Client Group.

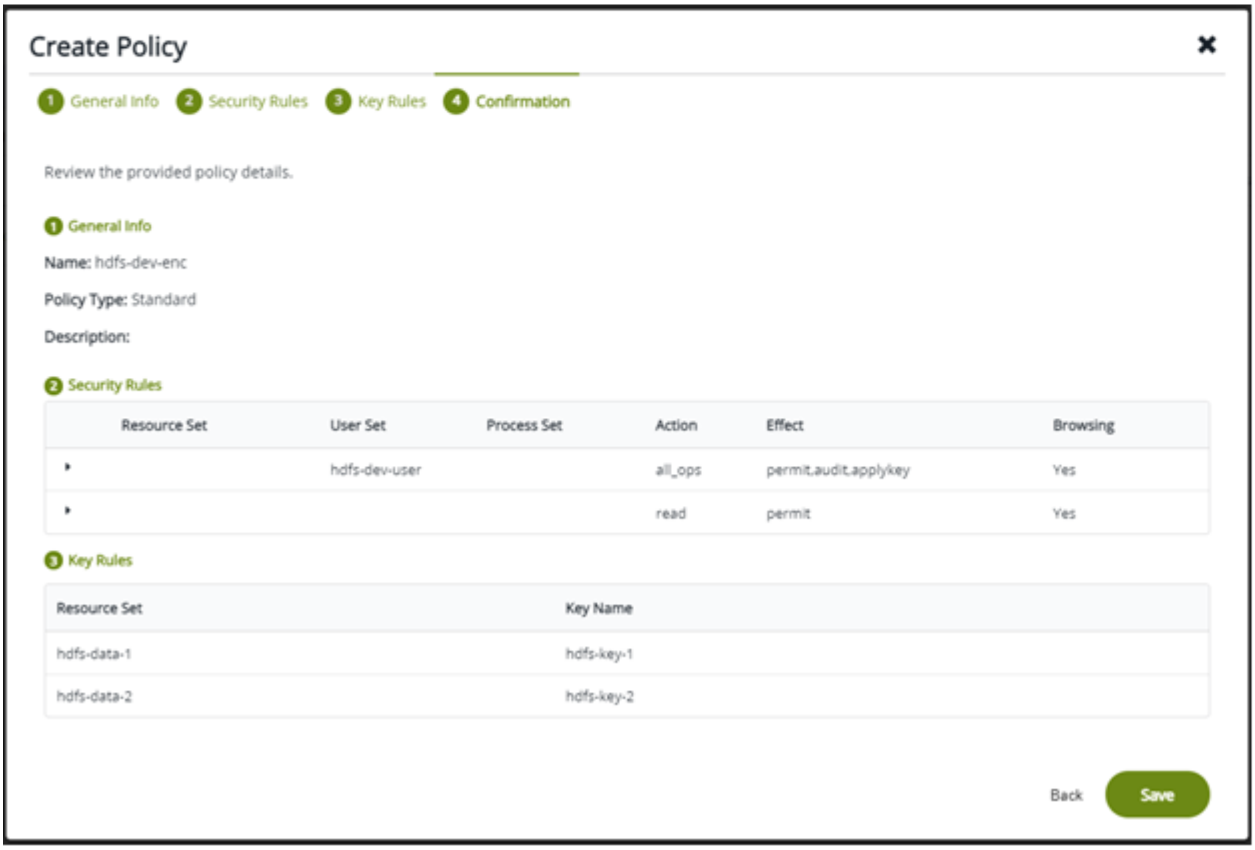
> **Note**
>
> Thales highly recommends using Auto Guard for HDFS. You can use manual guards, but this might result in data corruption if some nodes in a running cluster are guarded, while others are not.

Notes for HDFS Cluster Policies

The following screenshot depicts a valid HDFS policy. You can build a similar policy for your system:

The policy uses the following rules:

- For the user set hdfs-dev-user, the action is all_ops, the effect is Audit, Apply Key, Permit
- For other users the action is READ, the effect is Permit
- For the resource set hdfs-data-1, the key is hdfs-key-1
- For the resource set hdfs-data-2, the key is hdfs-key-2



Review the provided policy details.

1 General Info

Name: hdfs-dev-enc

Policy Type: Standard

Description:

2 Security Rules

	Resource Set	User Set	Process Set	Action	Effect	Browsing
▸		hdfs-dev-user		all_ops	permit,audit,applykey	Yes
▸				read	permit	Yes

3 Key Rules

Resource Set	Key Name
hdfs-data-1	hdfs-key-1
hdfs-data-2	hdfs-key-2

Additional Settings for HDFS (Linux Clients)

Depending on the Hadoop security authentication mode, additional settings are needed for CTE clients in an HDFS cluster. Add the following settings as appropriate.

```
!!! note

    `/usr/jdk64/jdk1.8.0_40/bin/java` is the Java executable used to
launch the HDFS services. Change the Java jdk path to reflect your end-
user environment.
```

- Sample setup when `hadoop.security.authentication` mode is simple.

```
|
authenticator+arg=+class=org.apache.hadoop.hdfs.server.namenode.N
ameNode|/usr/jdk64/jdk1.8.0_40/bin/java
               |
authenticator+arg=+class=org.apache.hadoop.hdfs.server.datanode.D
ataNode|/usr/jdk64/jdk1.8.0_40/bin/java
```

- Sample setup when `hadoop.security.authentication` mode is Kerberos.

```
|
authenticator+arg=+class=org.apache.hadoop.hdfs.server.namenode.N
ameNode|/usr/jdk64/jdk1.8.0_40/bin/java
               |
authenticator+arg=+class=org.apache.hadoop.hdfs.server.datanode.S
ecureDataNodeStarter|/usr/lib/bigtop-utils/jsvc
```

Adding a New DataNode to a CTE-protected HDFS

Use the following procedure to add a new DataNode to a CTE-protected HDFS. If not followed, HDFS encrypted files could be exposed in cleartext.

Note

If you already have CTE installed on the cluster nodes before Ambari installs the Hadoop software, see Install CTE on the Cluster Nodes before Ambari Installs Hadoop.

1. Install the HDFS client on the host. This option is available in Ambari when adding a new DataNode to the cluster.

2. Add the new node to the CipherTrust Manager database and make sure that host/ client settings of the new node is the same as existing nodes in the cluster. See the *CTE Installation for Hadoop chapter in the CTE Installation and Configuration Guide*.

3. Install CTE on the new node, register to CipherTrust Manager, and run `config-hadoop.sh` to prepare the libraries. See the *Configuring Hadoop to use CTE* section in the *CTE Installation and Configuration Guide*.

4. Make sure that the data directories (specified in `dfs.datanode.data.dir` property) exist on the new node. They must have the same permission and ownership as the other existing nodes in the cluster. If necessary, create them.

5. Add the host/client to the HDFS Host Group that is guarding the cluster. This is important: Do not rely on the DataNode to create the data directories as the data replication can occur before the GuardPoints are in effect.

6. Add the DataNode service to the new node. Again this option is available through Ambari.

7. If using Kerberos, check that the keytab files are created correctly.

8. Start the DataNode service on the new node.

9. Execute some `hdfs dfs` shell commands to ensure that encryption/decryption of data works correctly.

Install CTE on the Cluster Nodes before Ambari Installs Hadoop

If CTE is already installed on the cluster nodes before Ambari installs the Hadoop software, Ambari can mistakenly pick up the `.sec` directory in configuration steps to store the HDFS data. Make sure the following properties do not contain the `.sec` directory:

- DataNode data directory
- NameNode data directory
- Secondary NameNode checkpoint directory
- Zookeeper directory
- `yarn.nodemanager.local-dirs`
- `yarn.nodemanager.log-dirs`
- `yarn.timeline-service.leveldb-timeline-store.path`
- `yarn.timeline-service.leveldb-state-store.path`

Note

This list is not exhaustive. Depending on the Hadoop ecosystem packages installed, there can be others.

Configure NameNodes

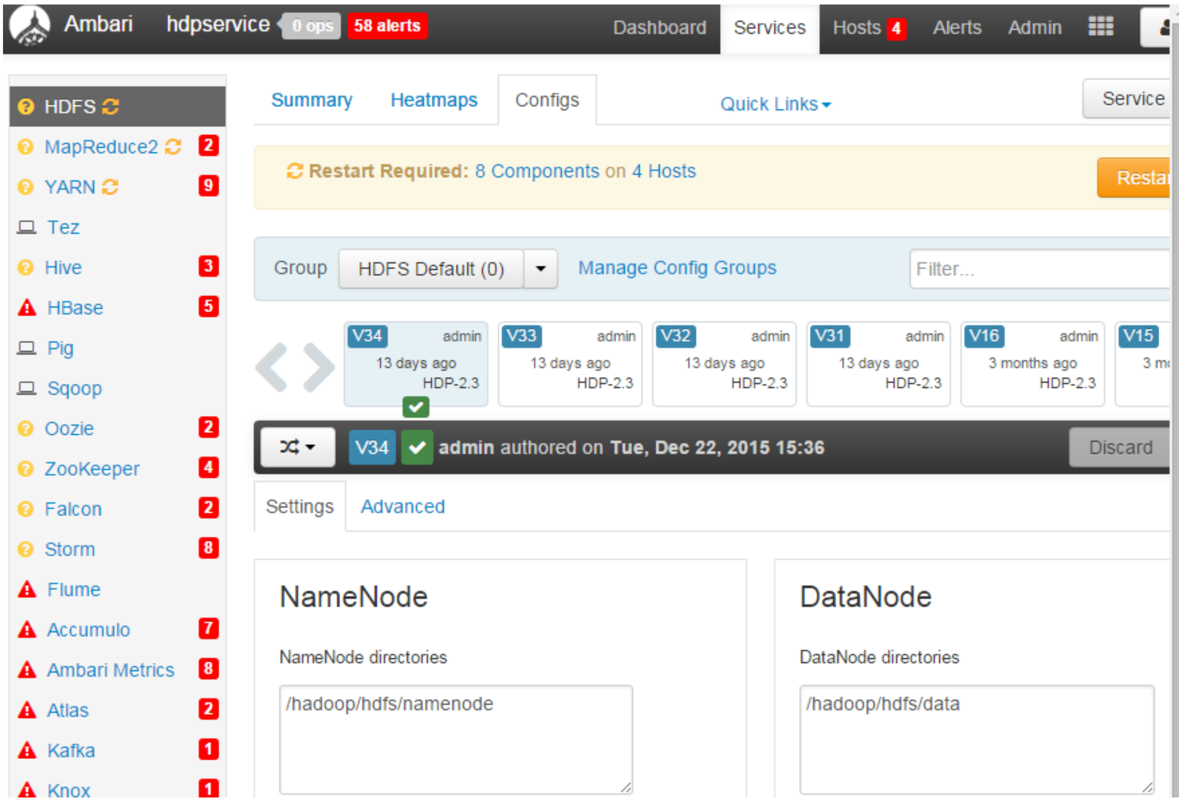
For both HDFS and Cloudera, you must configure the namenodes.

Configure the Ambari Hadoop NameNodes

The first step to implementing CTE on HDFS is for the **HDFS Administrator** to compile a list of the DataNode HDFS local file system directories, and create them on the NameNode local file systems. After this, the Administrator must add the NameNodes to an HDFS Host Group:

1. Compile a list of directories specified by `dfs.datanode.data.dir`. Obtain this from `hdfs-site.xml` or using Ambari go to:

   HDFS > Configs > Settings > DataNode > DataNode directories



NameNode

NameNode directories: /hadoop/hdfs/namenode

DataNode

DataNode directories: /hadoop/hdfs/data

2. If these directories do not already exist in the NameNode local file system, create them on each NameNode in your Hadoop cluster.

3. Pass the following information to the Administrator:

    - The directory list and instructions to create a GuardPoint for the HDFS Host Group on each of these directories.

    - Instructions to add the NameNodes IP addresses or host names to the HDFS Host Group.

Create and Configure the Cloudera Hadoop Namenodes and Datanodes

Create the role goups CTE on Cloudera Manager:

1. Create a new role group for the Thales-namenode.

    a. **Group Name**: thales-namenode

    b. **Role Type**: NameNode

    c. **Copy from**: NameNode Default Group

2. Move the name nodes to the group Thales-namenode.

3. Create a new role group for the Thales-datanode.

    a. **Group Name**: thales-datanode

    b. **Role Type**: DataNode

    c. **Copy from**: DataNode Default Group

4. Move the data nodes to the group thales-datanode.

Configure the Thales-namenode group

1. In the **NameNode Environment Advanced Configuration Snippet (Safety Valve)** dialog, in the thales-namenode field:

```
HADOOP_OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-
agent.jar=voragent"
```

2. In the **NameNode Advanced Configuration Snippet (Safety Valve)** for hdfs-site.xml, add the following values:

```
<property><name>dfs.vte.ioctl.device</name><value>/opt/vormetric/
DataSecurityExpert/agent/secfs/ .sec</value></property>
<property><name>dfs.vte.ioctl.lib</name><value>vorhdfs</value></
property>
<property><name>dfs.vte.rename.check</name><value>true</value></
property>
<property><name>dfs.block.access.token.enable</name><value>true</
value></property>
```

Configure the Thales-datanode group

1. In the **DataNode Environment Advanced Configuration Snippet (Safety Valve)** dialog, in the thales-namenode field:

```
HADOOP_OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-
agent.jar=voragent"
```

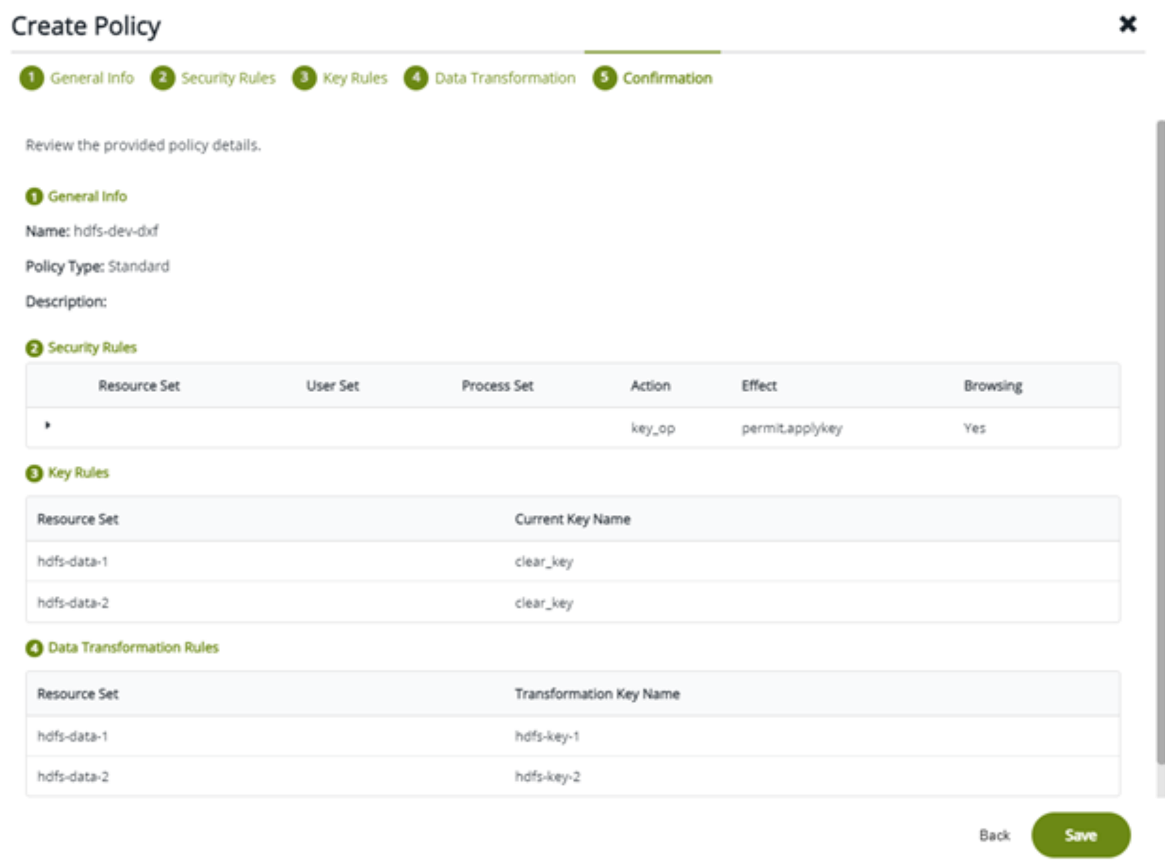
2. In the **DataNode Advanced Configuration Snippet (Safety Valve)** for hdfs-site.xml, add the following values:

```
<property><name>dfs.vte.ioctl.device</name><value>/opt/vormetric/
DataSecurityExpert/agent/secfs/.sec</value></property>
<property><name>dfs.vte.ioctl.lib</name><value>vorhdfs</value></
property>
<property><name>dfs.vte.user.push</name><value>true</value></
property>
<property><name>dfs.block.access.token.enable</name><value>true</
value></property>
```

Take a DataNode Offline and Perform Data Transformation

The next step in activating CTE on HDFS is to switch a DataNode to offline and transform (encrypt) its sensitive data. Once the data is transformed, the HDFS Admin can add the DataNode to the HDFS Host/Client Group. Then they can switch the DataNode back to online. Most of these procedures are completed by the **HDFS Administrator** although one is done by the Administrator.

1. **HDFS Administrator**: Switch a DataNode to offline.
2. **HDFS Administrator**: Encrypt the files in the directories specified by `dfs.datanode.data.dir` (see Configure NameNodes).
3. **Administrator**: Create encryption keys and a data transformation policy to transform the data.
   ** CipherTrust Manager**



Review the provided policy details.

1 General Info

Name: hdfs-dev-dxf

Policy Type: Standard

Description:

2 Security Rules

	Resource Set	User Set	Process Set	Action	Effect	Browsing
▸				key_op	permit.applykey	Yes

3 Key Rules

Resource Set	Current Key Name
hdfs-data-1	clear_key
hdfs-data-2	clear_key

4 Data Transformation Rules

Resource Set	Transformation Key Name
hdfs-data-1	hdfs-key-1
hdfs-data-2	hdfs-key-2

4. **Administrator**: After encrypting the data in those directories, add the DataNode host to the HDFS client group.

5. **HDFS Admin**: After the DataNode is added to the HDFS client group, activate the DataNode online.

6. Repeat this procedure for all of the DataNodes in your HDFS cluster.

For more information, see the CipherTrust Manager documentation.

Implementing CTE on HDFS on a Single Host

It is possible, though not recommended, that an HDFS NameNode and DataNode exist as separate processes on the same host. If this is your deployment, use the following CTE deployment guidelines:

1. Configure the HDFS NameNodes (see Configure NameNodes):

   > **Note**
   >
   > The directories specified by `dfs.datanode.data.dir` already exist on the local file system so you do not have to create them.

2. Pass the following information to the Administrator:

   - The `dfs.datanode.data.dir` directory list and instructions to create a GuardPoint for the HDFS Host Group on each of these directories.

   - Instructions to add the NameNodes IP addresses, or host names, to the HDFS Host Group.

3. Create an HDFS host/client group and host/client group GuardPoint:

   a. Administrator must create an HDFS Host Group to contain the HDFS nodes.

   b. The Administrator must create a GuardPoint for the Host Group on each of the directories specified by `dfs.datanode.data.dir`

4. Take the DataNode offline and perform a data transformation.

5. Add the NameNode/DataNode host/client to the host/client group.

Deploy Cloudera Hadoop on CTE

Deploying Cloudera Hadoop (CDP) on CTE consists of multiple steps that are documented in various locations. The following is the order that you should follow when installing CDP on CTE and links to the relevant documentation.

1. Deploy CDP 7 (Hadoop 3.1.1) cluster on RHEL 7.8 or RHEL 8.5, see the CDP Private Cloud Base Installation Guide

2. Configure Hadoop configuration for CTE, see Configure NameNodes

3. Cipher Trust Manager configuration, see CipherTrust Manager Documentation

4. CTE Installation and registration, see CipherTrust Transparent Encryption (CTE) Documentation Set

5. LDT, see CTE Live Data Transformation with CipherTrust Manager

HDFS Upgrade with CTE

To upgrade Hadoop, configure CTE to integrate with the new HDFS instance.

Upgrading one node at a time

Once CTE is installed and configured on the node:

1. Make sure that HDFS services are shut down on the node.

2. Upgrade **Hadoop**.

3. Type:

   ```
   /opt/vormetric/DataSecurityExpert/agent/secfs/hadoop/bin/config-hadoop.sh -i -y
   ```

4. Start HDFS services on the node.

Upgrade CTE with CTE-LDT in an HDFS Cluster

If you are using CTE-LDT with HDFS cluster, follow these steps when upgrading CTE, in order to maintain your CTE-LDT GuardPoints.

1. Suspend rekey on all data nodes.
2. Shutdown your namenodes/datanodes.
3. Upgrade CTE in the namenode first.

> **Note**
>
> Always upgrade namenodes before datanodes.

1. After CTE upgrade succeeds, type:

   ```
   config-hadoop.sh -i -y
   ```

2. On the Ambari admin console, start the namenode.
3. Verify that the CipherTrust java process successfully launched in the namenode. (You should not see an error message.) Type:

   ```
   ps -ef | grep java | grep vormetric
   ```

4. Check the CipherTrust Manager status. It should show LDT rekeyed status.
5. Check the namenode status. It should display the GuardPoint status and match the state before upgrade. Type:

   ```
    secfsd -status guard
   GuardPoint         Policy           Type   ConfigState   Status
   Reason
   ----------         ------           ----   -----------   ------
   ------
   ```

```
/hadoop/hdfs/data  LDT_HDFS_Sanity  local  guarded     guarded
N/A
```

6. Repeat the above steps for all of the datanodes in the HDFS cluster.

Rolling Upgrades

Hortonworks Data Platform has introduced rolling upgrades to automate the Hadoop upgrade process (`http://bit.ly/2pQrFo3`). The upgrade process is controlled by the Upgrade Pack (`http://bit.ly/2rkutvF`) that is predefined and certified by Hortonworks.

To integrate CTE with the upgrade, you need to temporarily change the Ambari scripts before performing the rolling upgrades and then restore the scripts after the upgrades.

1. On Ambari server machine, type:

```
 cd /var/lib/ambari-server/resources/common-services/HDFS/
2.1.0.2.0/package/scripts
```

2. Copy the utils.py file, type:

```
 cp utils.py utils.py.org
```

3. Using a text editor, add the following commands to `utils.py`:

```
 if action == "start":
if name == "namenode" or name == "datanode":
Execute(format("secfs/hadoop/bin/c onfig-hadoop.sh
-i -h {hadoop_bin}/../ -j <java home> -p hdp -d",
not_if=service_is_up,
user=params.root_user)
# For Redhat 6.x, uncomment the following command
# Execute(format("/etc/init.d/secfs secfsd restart"),
not_if=service_is_up, user=params.root_user)
# For Redhat 7.x, uncomment the following command
# Execute(format("/secfs restart"), not_if=service_is_up,
user=params.root_user)
before
```

```
Execute(daemon_cmd, not_if=service_is_up,
environment=hadoop_env_exports
The Java home of your HDFS instance should be used to replace
<java home>:
if action == "start":   if name == "namenode" or name ==
"datanode":
Execute(format("secfs/hadoop/bin/config-hadoop.sh -i -h
{hadoop_bin}
/../ -j <java home> -p hdp -d"), not_if=service_is_up,
user=params.root_user)
# For Redhat 6.x, uncomment the following command
# Execute(format("/etc/init.d/secfs secfsd restart"),
not_if=service_is_up, user=params.root_user)
# For Redhat 7.x, uncomment the following command
# Execute(format("/secfs restart"), not_if=service_is_up,
user=params.root_user)
Execute(daemon_cmd,
    not_if=service_is_up,
    environment=hadoop_env_exports)
```

4. Type:

```
ambari-server restart
```

5. Perform rolling upgrades.

6. During the upgrade process, many of the intermediate service status checks can fail. Skip over them by clicking on **Proceed to Upgrade**.

7. Click **Finalize** to complete the upgrade. If the active NameNode fails to activate due to the incompatible HDFS layout version, manually start the NameNode with '`-upgrade`' option to correct the layout version file.

```
/var/lib/ambari-server/ambari-sudo.sh su hdfs -l -s /bin/bash -c
'ulimit -c unlimited ; /usr/hdp/current/hadoop-client/sbin/hadoop-
daemon.sh --config /usr/hdp/current/hadoop-client/conf start
namenode -upgrade'
```

8. If there are excessive under-replicated blocks, run the following command to isolate them and manually start the replication:

   ```
    su - <$hdfs_user>
   # hdfs fsck / | grep 'Under replicated' | awk -F':' '{print $1}'
   >> /tmp/under_replicated_files
   # for hdfsfile in `cat /tmp/under_replicated_files`; do echo
   "Fixing $hdfsfile  :" ;  hadoop fs -setrep 3 $hdfsfile; done
   ```

9. Restart the HDFS services. Wait for the replication to complete and the NameNodes to exit safe mode.

10. When Hbase is restarted after upgrades, it tries to rename from: `/apps/hbase/data/.tmp/data/hbase/namespace` to: `/apps/hbase/data/data/hbase/namespace`, which may cause key conflict if the GuardPoint is set incorrectly (for example, `/apps/hbase/data/data` is guarded, but not `/apps/hbase/data/.tmp`). This results in Hbase shutting down.

    Before re-starting Hbase, make sure that the GuardPoint policies on the Hbase files are set correctly to cover all Hbase-related files. A broader GuardPoint (`/apps/hbase/data` instead of just `/apps/hbase/data/data` and other folders) could fix this issue.

11. Check cluster upgrade by verifying the `hadoop version`.

12. Run a few map reduce jobs and Hbase commands to make sure that the entire Hadoop stack is working properly.

13. Rename `utils.py.org` to `utils.py`

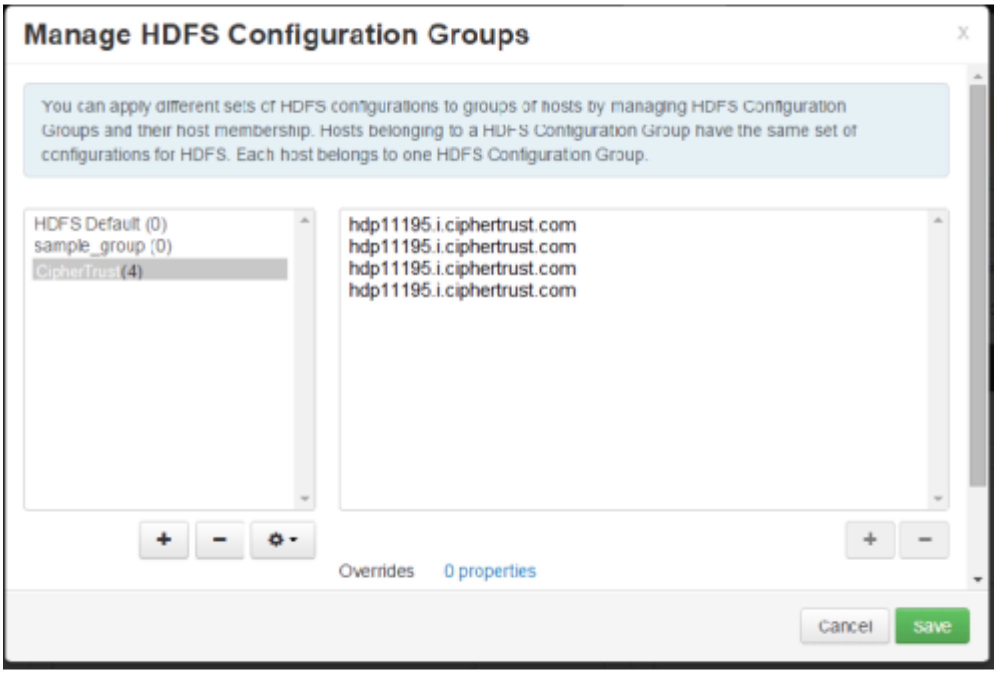
Configure the Hadoop Cluster for CTE

Configure the Hadoop cluster to use CTE before installing and configuring CTE on the nodes. Use Ambari to perform this configuration.

Create a CipherTrust Configuration Group

The CipherTrust Configuration Group will eventually contain all of the hosts in your Hadoop cluster. At first you create an empty group. Later you populate it with the hosts on which you will install and configure agents.

1. On Ambari, go to **HDFS > Configs > Manage Config Groups**.
2. Add a new configuration group: *CipherTrust*.
3. Make the group, *CipherTrust*, the current group.



Manage HDFS Configuration Groups

You can apply different sets of HDFS configurations to groups of hosts by managing HDFS Configuration Groups and their host membership. Hosts belonging to a HDFS Configuration Group have the same set of configurations for HDFS. Each host belongs to one HDFS Configuration Group.

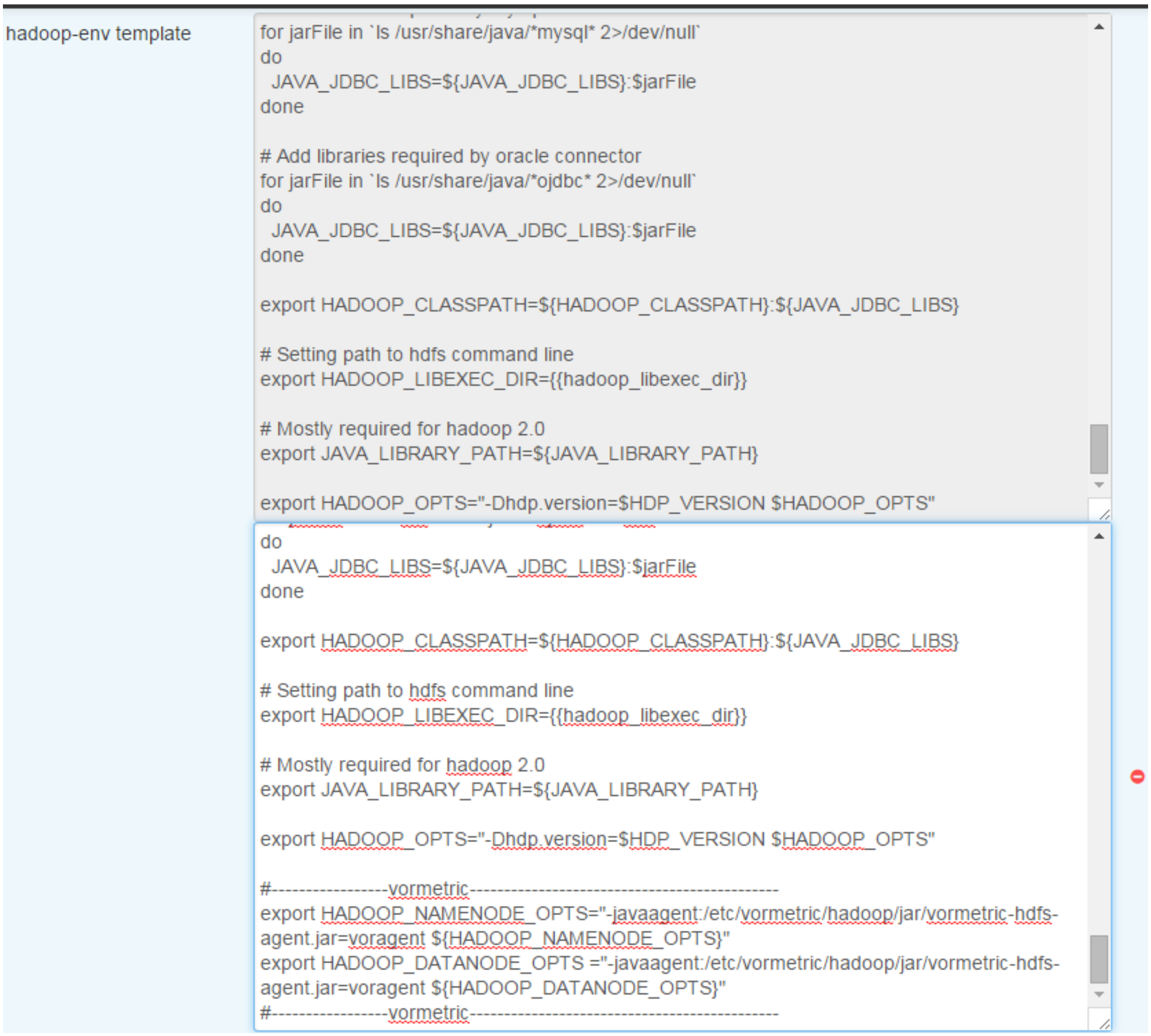
HDFS Default (0)
sample_group (0)
CipherTrust (4)

hdp11195.i.ciphertrust.com
hdp11195.i.ciphertrust.com
hdp11195.i.ciphertrust.com
hdp11195.i.ciphertrust.com

Overrides 0 properties

Update the Hadoop-env Template with CTE Settings

1. Go to **HDFS > Configs > Advanced > Advanced hadoop-env > hadoop-env template**.
2. Copy and paste the original `hadoop-env` templates into the Thales template and add the following two export lines to specify that the CTE Java agent is instrumented into NameNode and DataNode.



```
for jarFile in `ls /usr/share/java/*mysql* 2>/dev/null`
do
  JAVA_JDBC_LIBS=${JAVA_JDBC_LIBS}:$jarFile
done

# Add libraries required by oracle connector
for jarFile in `ls /usr/share/java/*ojdbc* 2>/dev/null`
do
  JAVA_JDBC_LIBS=${JAVA_JDBC_LIBS}:$jarFile
done

export HADOOP_CLASSPATH=${HADOOP_CLASSPATH}:${JAVA_JDBC_LIBS}

# Setting path to hdfs command line
export HADOOP_LIBEXEC_DIR={{hadoop_libexec_dir}}

# Mostly required for hadoop 2.0
export JAVA_LIBRARY_PATH=${JAVA_LIBRARY_PATH}

export HADOOP_OPTS="-Dhdp.version=$HDP_VERSION $HADOOP_OPTS"
```

```
do
  JAVA_JDBC_LIBS=${JAVA_JDBC_LIBS}:$jarFile
done

export HADOOP_CLASSPATH=${HADOOP_CLASSPATH}:${JAVA_JDBC_LIBS}

# Setting path to hdfs command line
export HADOOP_LIBEXEC_DIR={{hadoop_libexec_dir}}

# Mostly required for hadoop 2.0
export JAVA_LIBRARY_PATH=${JAVA_LIBRARY_PATH}

export HADOOP_OPTS="-Dhdp.version=$HDP_VERSION $HADOOP_OPTS"

#----------------vormetric------------------------------------------
export HADOOP_NAMENODE_OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-agent.jar=voragent ${HADOOP_NAMENODE_OPTS}"
export HADOOP_DATANODE_OPTS ="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-agent.jar=voragent ${HADOOP_DATANODE_OPTS}"
#----------------vormetric------------------------------------------
```

```
export HADOOP_NAMENODE_OPTS="-javaagent:/etc/vormetric/hadoop/jar/
vormetric-hdfs-agent.jar=voragent ${HADOOP_NAMENODE_OPTS}"
export HADOOP_DATANODE_OPTS="-javaagent:/etc/vormetric/hadoop/jar/
vormetric-hdfs-agent.jar=voragent ${HADOOP_DATANODE_OPTS}"
```

Modify the HDFS IOCTL

1. Go to **HDFS > Configs > Advanced > Custom hdfs-site**.

2. In the **dfs.vte.ioctl.lib** field, type: `vorhdfs`

3. In the **dfs.vte.ioctl.device** field, type:

```
/opt/vormetric/DataSecurityExpert/agent/secfs/.sec
```

Change the HDFS File Rename Check

1. Go to **HDFS > Configs > Advanced > Custom hdfs-site**.
2. Set **dfs.vte.rename.check** to **true**

User Information Push

You only need this configuration if you want to use the original user information with HDFS operation for IO access check.

Note

It has a performance cost.

1. Go to **HDFS > Configs > Advanced > Advanced hdfs-site** and uncheck **HDFS Short-circuit read**.
2. Set **dfs.block.access.token.enable** to **true**.
3. Go to **HDFS > Configs > Advanced > Custom hdfs-site** and set **dfs.vte.user.push** to **true**.

Uninstalling CTE for the Hadoop Cluster

This section explains how to remove CTE and restore the environment back to the non-CTE cluster environment. Thales recommends uninstalling the agent from HDFS nodes one by one, starting from the DataNode. Uninstall the agent from the NameNode last.

1. Shut down one DataNode.
2. Perform the normal agent uninstall procedure.
3. Go to Ambari, remove the DataNode host from the CipherTrust Configuration Group.
4. Start the DataNode.

5. Delete the CipherTrust Configuration Group from Ambari when agent is uninstalled from all DataNodes.

6. Repeat these steps for each DataNode.

7. Repeat these steps for each NameNode.

CTE Installation and Configuration

After configuring the Hadoop cluster for CTE:

1. Install and register CTE on the HDFS nodes.

    - You can do this to all the nodes at once, but the HDFS is unavailable during CTE installation and configuration.

    - You can also do this one node at a time. If you install and register CTE nodes one at a time, you must start from NameNode, then DataNode, and always keep NameNode service up once NameNodes are configured.

2. In either case, add the FQDN of the node to the CipherTrust Configuration Group, then proceed with agent installation and configuration. See Installing and Configuring CTE on an HDFS Node.

    - Modify the Host Group. See Modifying host settings for HDFS hosts on the CipherTrust Manager.

    - Configure CTE by running `config-hadoop.sh` on the HDFS node. See Configuring Hadoop to Use CTE.

    - Review the SecFS configuration variables that support the HDFS name cache. HDFS Name Cache.

Installing and Configuring CTE on an HDFS Node

1. Using Ambari, add the FQDN of the node to the CipherTrust Configuration Group. See Create a CipherTrust Configuration Group.

2. Install, configure, and register CTE.

3. Modify the host settings for each node. See Modifying host settings for HDFS hosts on the CipherTrust Manager.

Modifying host settings for HDFS hosts on the CipherTrust Manager

The Hadoop service can start as root and then downgrade to an unprivileged user. If the unprivileged user is not authenticated by password, CTE flags the user as fake. CTE cannot allow a user to access a resource protected by a user rule when the user is faked, even if the user matches the permit rule. Because of this, modify the CipherTrust Manager host/client setting as follows:

- On Ambari, go to **HDFS > Configs > Advanced > Advanced core-site** and find out if **hadoop.security.authentication** mode is set to **simple** (no authentication) or **Kerberos**.

Modifying host settings for alternate versions of Java

Sometimes, you may have more than one version of Java installed on your systems. This can create issues while using Hadoop, which uses Java to launch different services.

CipherTrust Transparent Encryption expects the following version of Java to be installed on a system:

```
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.252.b09-2.el7_8.x86_64/jre/bin/
java
```

If you have a different version of java installed on any of your systems, then you must perform a simple modification and add authenticators to the Host settings section in CipherTrust Manager for that version of Java as well. See the following section for more information.

Simple Modification

To use a **simple** modification, ask the Administrator to add the following lines to the host/client group:

```
`/usr/jdk64/jdk1.8.0_40/bin/java` is the Java executable used to launch
the HDFS services. Change the Java jdk path to reflect your end-user
environment.

|
authenticator+arg=+class=org.apache.hadoop.hdfs.server.namenode.NameNo
de|
/usr/jdk64/jdk1.8.0_40/bin/java
|
authenticator+arg=+class=org.apache.hadoop.hdfs.server.datanode.DataNo
de|
/usr/jdk64/jdk1.8.0_40/bin/java
```

The entire host/client settings will look like this:

```
|authenticator|/usr/sbin/sshd
|authenticator|/usr/sbin/in.rlogind
|authenticator|/bin/login/
|authenticator|/usr/bin/gdm-binary
|authenticator|/usr/bin/kdm
|authenticator|/usr/sbin/vsftpd

|
authenticator+arg=+class=org.apache.hadoop.hdfs.server.namenode.NameNo
de|usr/jdk64/jdk1.8.0_40/bin/java
|
authenticator+arg=+class=org.apache.hadoop.hdfs.server.datanode.DataNo
de|/usr/jdk64/jdk1.8.0_40/bin/java
```

Modifying Host Group for HDFS NameNodes HA on CipherTrust Manager

To enable high availability (HA) for your HDFS NameNodes, ask the Administrator to add the following lines to the host/client group.

```
`/usr/jdk64/jdk1.8.0_40/bin/java` is the Java executable used to launch the HDFS services. Change the Java `jdk` path to reflect your end-user environment.

|authenticator+arg+class=org.apache.hadoop.hdfs.qjournal.server.JournalNode|/usr/jdk64/jdk1.8.0_40/bin/java
|authenticator+arg+class=org.apache.hadoop.yarn.server.applicationhistoryservice.ApplicationHistoryServer|/usr/jdk64/jdk1.8.0_40/bin/java
|trust+arg=+class=org.apache.hadoop.hdfs.tools.DFSZKFailoverController|/usr/jdk64/jdk1.8.0_40/bin/java
```

The entire host/client group for HA (in this example, with Kerberos) will look like this:

```
|authenticator|/usr/sbin/sshd
|authenticator|/usr/sbin/in.rlogind
|authenticator|/bin/login/
|authenticator|/usr/bin/gdm-binary
|authenticator|/usr/bin/kdm
|authenticator|/usr/sbin/vsftpd

|authenticator+arg+class=org.apache.hadoop.hdfs.server.namenode.NameNode|/usr/jdk64/jdk1.8.0_40/bin/java
|authenticator+arg=+class=org.apache.hadoop.hdfs.server.datanode.SecureDataNodeStarter|/usr/lib/bigtop-utils/jsvc
|trust+arg+class=org.apache.hadoop.hdfs.qjournal.server.JournalNode|/usr/jdk64/jdk1.8.0_40/bin/java
|trust+arg=+class=org.apache.hadoop.yarn.server.applicationhistoryservice.ApplicationHistoryServer| \
/usr/jdk64/jdk1.8.0_40/bin/java
|trust+arg=+class=org.apache.hadoop.hdfs.tools.DFSZKFailoverController|/usr/jdk64/jdk1.8.0_40/bin/java
```

Configuring Hadoop to Use CTE

1. On the HDFS node, type:

```
/opt/vormetric/DataSecurityExpert/agent/secfs/hadoop/bin/config -
hadoop.sh –i.
```

HDFS prompts you for the following information:

- **Hadoop product name**: (i.e. *hdp*)
- **Hadoop product version**: (i.e. *2.6.0.2.2.0.0-2041*)
- **Path to JAVA_HOME used by Hadoop**: (i.e. */usr/jdk64/jdk1.8.0_40*)

> **Note**
>
> Alternatively, you can use the automated installation option: */opt/vormetric/DataSecurityExpert/agent/secfs/hadoop/bin/config-hadoop.sh -i -p hdp -v 2.6.0.2.2.0.0-2041 -j /usr/jdk64/jdk1.8.0_40* 2.

2. Verify the configuration the using the `-s` option:

```
/opt/vormetric/DataSecurityExpert/agent/secfs/hadoop/ bin/config-h
adoop.sh -s

Vormetric-Hadoop Configuration Status
PRODUCT_NAME=hdp- PRODUCT_VERSION=3.0.0.0-2557 HADOOP_HOME=/usr/h
dp/current/hadoop-client/sbin/../- HADOOP_VERSION=2.7.1 HADOOP_PR
ODUCT_VERSION=2.7.1.2.3.0.0-2557- HADOOP_VERSION_MAJOR=2.7 LIBVOR
HDFS_SO=/usr/hdp/current/hadoopclient/sbin/..//lib/native/libvorh
dfs.so LIBHDFS_SO=/etc/vormetric/hadoop/lib/libhdfs.so VORMETRIC_
HADOOP=/etc/vormetric/hadoop-
#----------vormetric--------------------------------------------
export HADOOP_NAMENODE_OPTS="-javaagent:/etc/vormetric/hadoop/jar/
vormetric-hdfs-agent.jar=voragent $
{HADOOP_NAMENODE_OPTS}"v6 . . . . . 62 export
HADOOP_DATANODE_OPTS="-javaagent:/etc/vormetric/hadoop/jar/
vormetric-hdfs-agent.jar=voragent ${HADOOP_DATANODE_OPTS}"
#-------------vormetric----------------------------------------- /
etc/vormetric/hadoop/lib/libhdfs.so ...ok
/usr/hdp/current/hadoop-client/sbin/..//lib/native/libvorhdfs.so .
..ok /etc/vormetric/hadoop/gen-vor-hadoop-env.sh ...ok
/etc/vormetric/hadoop/vor-hadoop.env ...ok
```

```
Looks ok.
Vormetric Transparent Encryption Agent 6.0.3 Installation and Con
figuration Guide v6 . . . . .62 export HADOOP_DATANODE_OPTS="-
javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-
agent.jar=voragent ${HADOOP_DATANODE_OPTS}"
#--------------vormetric ------------------------------------- /
etc/vormetric/hadoop/lib/libhdfs.so ...ok /usr/hdp/current/hadoop
client/sbin/..//lib/native/libvorhdfs.so ...ok /etc/vormetric/had
oop/gen-vor-hadoop-env.sh ...ok /etc/vormetric/hadoop/vor-hadoop.e
nv ...ok
Looks ok
```

Verify secfsd is Running with Hadoop Environment

Use a text editor to view the `/etc/init/secfsd - upstart.conf` file. The file should contain env entries, type:

```
cat /etc/init/secfsd - upstart.conf
```

HDFS Name Cache

Obtaining the HDFS file name from the NameNode is network intensive, so the map from HDFS block file name to HDFS file name is cached in a hash table. The following secfs configuration variables are used to support the hash cache. They are provided in case you need to tune the memory management of the name cache for better performance.

- **hdfs_cache_entry**: Default is 1,024.000, which could cover up to 125TB HDFS data because the default HDFS block size is up to 128MB (128MB * 1024000 = 125TB).
- **hdfs_cache_bucket**: Default is 10240.
- **hdfs_cache_timeout**: Default to 30 minutes.
- **hdfs_cache_interval**: Default wake up interval for a worker thread to update the cache entry whose timeout has expired is 10 seconds.

On Linux, you can configure each secfs configuration variable using the voradmin commands:

For example, to configure the variable, type:

```
voradmin secfs config hdfs_cache_interval 30
```

Note

Because HDFS rename is a metadata operation inside the HDFS NameNode and does not call into the local file system, the hash cache might contain expired data. The HDFS NameNode is coded to prevent renamed data from crossing a key boundary to prevent data corruption. However, other access checks based on the HDFS file name may give incorrect information if the name is expired. Understand this risk before using this feature.

Enabling CTE on HDFS

To enable CTE on your HDFS:

1. Restart CTE agent on the node.

    - For Redhat 6.x, type:

      ```
      /etc/init.d/secfs restart
      ```

    - For Redhat 7.x or 8.x, type:

      ```
      /etc/vormetric/secfs restart
      ```

2. Restart the Hadoop Services in the cluster.

You can now create GuardPoints to protect your entire HDFS.

Deleting Metadata in HDFS when Migrating Out of LDT

In an HDFS deployment, if you migrate from an LDT to a non-LDT environment, the administrator must delete the LDT mdstore file.

In the following example:

`/hadoop/hdfs` is the mount point

`/hadoop/hdfs/data` is the GuardPoint

To manage the migration:

1. In the CipherTrust Manager, click **Client > Client Groups**.
2. Click < *host/client group name*>. The Edit host/client group - window opens.
3. Click **GuardPoints**.
4. Select the appropriate HDFS directory with an LDT GuardPoint, and click **Unguard**.
5. Using the Ambari admin console, shutdown all NameNode/DataNode one by one. Ensure that no HDFS GuardPoints are busy.
6. Ensure that no GuardPoints are configured on any HDFS node in the cluster, type:

   ```
   secfsd -status guard
   No GuardPoints configured
   ```

7. On the node running secfs, type:

   ```
   voradmin ldt attr delete <guard path>
   # voradmin ldt attr delete /hadoop/hdfs/data
   LDT metadata has been removed from all files in GuardPoints /
   hadoop/hdfs/data
   ```

8. On the system, verify that the metadata store has been removed from the secfs mount points, type:

   ```
    voradmin ldt rmstore <mount_point>
   # voradmin ldt rmstore /hadoop/hdfs
   Enter YES if /hadoop/hdfs does not include any GuardPoints
   associated with an LDT policy ->YES
   MDS file /hadoop/hdfs/__vorm_mds__  has been removed.
   ```

9. Verify that the metadata store has been removed from the secfs mount points, type:

```
 ls -altr <mount_point>
# ls -altr /hadoop/hdfs
You should not see `/hadoop/hdfs/__vorm_mds__ ` listed.
```

10. Repeat the above steps for each node in the HDFS cluster.

Using CTE with Oracle

This section describes how to install and configure CTE on Oracle RAC ASM, how to use an ASM Filter Driver Linux. It contains the following topics:

- Oracle RAC ASM and ASMLib
- Oracle RAC ASMLib Multi-Disk Online Method
- About Oracle RAC ASM Raw Devices
- Oracle RAC ASM Multi-Disk Online Method
- Oracle RAC ASM Multi-Disk Offline Method (Backup/Restore)
- Surviving the Reboot and Failover Testing
- Using CTE with Oracle ASM Filter Driver
- Basic Troubleshooting Techniques
- Using Oracle with CipherTrust Transparent Encryption UserSpace

Oracle RAC ASM and ASMLib

This section describes how to install and configure CipherTrust Transparent Encryption on an Oracle RAC ASM and ASMLib.

Using CipherTrust Transparent Encryption with an Oracle RAC ASM

You can apply CipherTrust Transparent Encryption when the Oracle DB is active or inactive. If you choose to use it while the Oracle DB is active, it eliminates any downtime. You can apply CipherTrust Transparent Encryption during low volume traffic time frames. If you choose to use this option, then use the **rebalance** function of ASM. This allows you to:

1. Migrate data off of a disk so that it can be dropped/removed from a **Diskgroup**.

2. Apply CipherTrust Transparent Encryption protection.

3. Add the disk back into the diskgroup.

Caution

You must drop the related ASM diskgroup first, before dropping the disk. If, for example, you drop an ASM disk before dropping an ASM diskgroup, and then add it back to the diskgroup without cleanly wiping the disk, the ASM diskname will be corrupted. To avoid this problem, clear out the disk before you add it back into the diskgroup. `Example: dd if=/dev/zero of=/dev/secvm/dev/mapper/asmdg-asmlv002 bs=32k`

ASMLib

ASMLib is an optional support library for the Automatic Storage Management feature of the Oracle Database. If the customer is using ASMLib, then management is performed through an Oracle ASM command line. Using this can be simpler than the setup required for standard ASM. The commands and details of the procedure differ as well.

Important ASM Commands and Concepts

Rebalancing Disks

When you drop/remove a disk from the diskgroup, it is important to apply the proper value for the power setting for rebalance and to use the `WAIT` command.

Example ASM Command:

```
SQL> ALTER DISKGROUP <DiskGroupName> DROP DISK <diskName> REBALANCE
POWER 8 WAIT;
```

- The rebalance command moves the data off of the disk that you are removing from the diskgroup, distributing the data across the remaining DISKS.

- The power setting is a number from 1 to 11. It determines how much processing power is dedicated to the rebalance, versus normal operations. Unless the encrypting occurs during heavy traffic volume, the minimum value you should use is 6. Otherwise, consult the customer's DBA for the proper setting. An appropriate value to start with is 8.

Mapping Raw Devices

You can map raw devices for this configuration using:

- **Multipath I/O**

  This is typically evident when the path for the mapped devices is set to: `/dev/mapper/<device-name>`.

- **Raw devices**

  Some customers use raw devices to map a name like raw3 to a specific device name. You usually find this mapping in a file called: `/etc/sysconfig/rawdevices`.

> **Note**
>
> It is important to understand how the device names are used and if they are the same across all of the RAC nodes.

- **EMC PowerPath**

  If using EMC PowerPath then the device names are similar to the following: `/dev/emcpowerXXXX`.

When browsing the CipherTrust Manager through the local host, you cannot find Power Path devices. You must manually input the paths. The guarded disk names are prepended with: `/dev/secvm`.

Checking Rebalance Status

The `wait` command is very important when ASM performs a rebalance. When you specify `wait`, the command prompt does not display until all of the data is rebalanced and migrated off of the disk. If you do not specify wait, the command prompt returns immediately, and you must issue the following ASM command to check the status of the rebalance:

```
SQL> select * from v$asm_operation;
```

This command returns information about the:

- State
- Current power level
- Current amount rebalanced
- Estimated work until completion
- Rate
- Estimated minutes
- Any error codes

Note

It is highly recommended that you always specify the wait command when performing a **Drop Disk** with Rebalance. If it is not specified, ASM may prematurely release the disk, thereby allowing CTE to place a GuardPoint on the disk before the rebalance completes. This action may corrupt the data.

Oracle cautions against this issue:

Caution

The `ALTER DISKGROUP...DROP DISK` statement returns before the drop and rebalance operations complete. Do not reuse, remove, or disconnect the dropped disk until the `HEADER_STATUS` column in the `V$ASM_DISK` view for this disk changes to `FORMER`. You can query the `V$ASM_OPERATION` view to determine the amount of time remaining for the drop/rebalance operation to complete. For more information, refer to the Oracle Database SQL Language Reference and the Oracle Database Reference.

Determining Best Method for Encrypting Disks

A diskgroup can contain one or multiple disks. You must determine if the diskgroup contains enough disks and free space for encryption. If the diskgroup contains only one disk, or multiple disks but not enough free space, then you must use the **Offline** (backup/restore) method for encryption.

If the diskgroup contains more than one, you can use the **Online** (rebalancing) method. During rebalancing, additional disks allow for migrating data from the original disk so that it can be encrypted, added back into the diskgroup, and then migrated back to the source disk. Therefore, if the customer does not want to permanently add extra disks, they can add disks temporarily, just for rebalancing.

In general, once you have completed the initial setup for the operating system with which you are working, for both ASM or ASMLib, the high-level process is the same for applying CTE protection to raw devices and using them.

Online Method (No Application / Database Downtime)

Typically, when using the online method, follow these steps:

1. Make an ASM disk available for protection by either removing a disk from an existing diskgroup, or allocating a new disk.
2. Apply CTE encryption to the disk.
3. Add each protected disk to the diskgroup.
4. Restart the nodes and the failover test.
5. Repeat the previous steps for each disk in the diskgroup.

Offline Method (Backup the DB)

Typically, when using the offline method, follow these steps:

1. Backup the database.
2. Make an ASM disk available for protection by either removing a disk from an existing diskgroup, or allocating a new disk.
3. Stop the Oracle database.
4. Delete the diskgroup.
5. Apply CTE encryption to the disk.
6. Recreate the diskgroup.
7. Add the protected disk to the diskgroup.

8. Restart the nodes and the failover test.

9. Repeat the previous steps for each disk in the diskgroup.

General Prerequisites

Setup

- Verify that you have a current backup of the database
- Install and register CTE agents on all RAC node Hosts
- Create a **Host Group** and add all RAC node hosts as members
- Create an encryption key for the Oracle RAC Database / Application
- Create an Oracle policy using the proper encryption key

Note

If the raw device mappings for the disk(s) are **not** identical across all nodes in the RAC, then you cannot use a Host Group for managing the GuardPoint within the CipherTrust Manager. You **must** apply the GuardPoint to each Host individually. This is typically not optimal, as a Host Group is the most effective and consistent way to manage GuardPoints for Oracle RAC environments.

Altering ASM_DISKSTRING on ASM

ASM uses the `asm_diskstring` setting to identify the path where ASM will attempt to locate available disks to use. If you are using device names when adding the disk, you must modify the string to include the path to SecVM.

1. To retrieve the `ASM_DISKSTRING` setting, type:

```
SQL> SHOW PARAMETER ASM_DISKSTRING
```

2. To modify the setting, type:

```
SQL> ALTER SYSTEM SET ASM_DISKSTRING='/dev/mapper/*', '/dev/secvm/
dev/mapper/*';
```

Where the path added is the path to SecVM.

ASMlib manages the binding, not ASM. ASMlib creates ASMlib devices on the SecVM devices and presents it to ASM. ASM automatically recognizes the new device. This creates the need to alter diskstrings for ASM. In addition, Oracle ASM sees a new device created using ASMlib and Raw, by default.

For example:

Using Oracle ASMlib to bind the device:

```
oracleasm createdisk <devicename> /dev/secvm/dev/<blockdev>
```

Using the raw command to bind the device:

```
raw /dev/raw/rawN /dev/secvm/dev/<blockdev>
```

Specific Prerequisites

Establishing a Starting Point

In many production environments, you may find that it has been a very long time since the RAC nodes have had the services restarted or have been completely rebooted. This can result in a lack of understanding of the actual state of the RAC cluster and its ability to survive a reboot on its own, prior to installing CTE.

Restarts can uncover issues in the RAC environment that are unrelated to CTE. To avoid issues after a CTE installation, Thales recommends that you restart each RAC node **AFTER** CTE is installed and **PRIOR** to establishing any GuardPoints. This may not be feasible in a single node configuration. However, by doing so, CTE is installed but inactive, and you can ensure that the platform is in a workable state prior to getting started.

The Importance of Device Mapping

It is important to use device naming and mapping in a multi-node RAC configuration. Verify the device names to ensure that the disks are mapped to the same disks on each RAC node before applying any GuardPoints. Thales recommends that RAC nodes use the same device names across all nodes. If they do not match, then problems can occur.

If the RAC nodes use the same device names, use a Host Group to create GuardPoints. If they do not match, do not use a Host Group to create GuardPoints. Set them up independently on each Host.

Important Note about Raw Devices on UNIX

In general, raw devices are created as either character or block mode devices. Any I/O performed on character devices is non-buffered, while I/O on block devices is buffered and performed in defined block sizes (that is, 4K bytes).

While the Oracle documentation for using ASM with raw devices indicates that you can use either character or block devices, **CTE REQUIRES a block device for guarding**.

Note

- Attempting to apply a GuardPoint on a character device that does not have a corresponding block device may result in a GuardPoint that never encrypts data. The status of the GuardPoint never shows as guarded.
- The WebUI does not support browsing for the character devices. You would need to manually paste the name into the WebUI.

Block Device Support

CipherTrust Transparent Encryption UserSpace block device support requires that the underlying kernel supports I/O direct mode for loop devices (`LOOP_SET_DIRECT_IO`) in order to properly flush and retrieve data from a disk. This support was not added in the kernel loop driver until **kernel 4.10.** Any distribution with a kernel level lower than that may improperly cache data and therefore, should not be used in a multi-node setup sharing block devices because stale data may be returned to the application.

Oracle RAC ASMLib Multi-Disk Online Method

The online method describes how to remove, protect and add disks to a diskgroup.

Assumptions

Using the Online Method assumes that there is enough free space in the diskgroup so that you can drop/remove, protect and then add the disk back into the diskgroup.

Note

During the initial investigation, you may want to ensure that you have the correct raw device name for each disk that you plan on protecting. Before making any changes to the ASM configuration, obtain the definitive device names for each disk by running the following from the command prompt:

`oracleasm querydisk -p <diskName>`

To add the disk to the diskgroup using the online method and make it ready for use:

1. Open a terminal session on both RAC Nodes.

2. On **RAC Node 1**, perform the following:

   a. On the ASM, type the following to remove a disk for the diskgroup.

   ```
   SQL> ALTER DISKGROUP <diskGroupName> DROP DISK <diskName> REB
   ALANCE POWER 11 WAIT;
   ```

   b. Delete the disk from ASM, type:

   ```
   oracleasm deletedisk <diskName>
   ```

   c. Verify that the disk is deleted from the ASM and therefore, it is not listed, type:

   ```
   oracleasm listdisks
   ```

Note

If you are planning to apply a GuardPoint to a raw device that is currently in an ASM diskgroup, you must remove and delete the disk from the diskgroup before you apply the GuardPoint. ASMLib will not see the guarded disk if you skip this step. When deleting the disk, make sure that the deletion completes before continuing.

About Oracle RAC ASM Raw Devices

When Not Using ASMLib

Before starting the CTE implementation, investigate how the customer is using raw devices for their ASM configuration.

Devices using Raw Bindings

Typically, a device that uses a raw binding looks like the following to ASM:

`/dev/raw/raw1`

If the device is mapped this way, you must locate where the mapping is performed. Typically, you can find this in the following configuration file:

`/etc/sysconfig/rawdevices`

The underlying binding could be to either a **standard device** name or a **multipath** I/O device name. Either way, you must find where the bind commands are run so that you can modify them for SecVM.

Note

If raw bindings are in use, then typically no changes are needed for the `asm_diskstring`. Because the binding to the actual device is created through the `bind` command, locate where the binding occurs and change the binding to SecVM.

Multipath I/O Devices

Devices using multipath I/O are typically found with the name:

`/dev/mapper/mpath1`

Generally, when using multipath I/O, you create SecVM on the multipath device name.

Note

- **CTE**: If you use multipath I/O devices in the ASM configuration to add its disk, you must modify the `asm_diskstring` parameter to include the `/dev/secvm/dev/*` path.
- **CTE-U**: If you use multipath I/O devices in the ASM configuration to add its disk, you must modify the `asm_diskstring` parameter to include the `/dev/secvm/*` path.

Standard Devices

In many cases the ASM configuration may be using plain device names, like the following:

`/dev/sda1`

Note

- **CTE**: If you use standard device names in the ASM configuration to add a disk, you must modify the `ASM_DISKSTRING` parameter to include the `/dev/secvm/dev/*` path.
- **CTE-U**: If you use standard device names in the ASM configuration to add a disk, you must modify the `ASM_DISKSTRING` parameter to include the `/dev/secvm/*` path.

Consistent Naming of Devices across RAC Nodes

As previously stated, if the raw device mappings for the disk(s) are **NOT** identical across all nodes in the RAC, then you **CANNOT** use a Host Group and you **MUST** apply the GuardPoints to each Host individually. This is typically NOT optimal, as a Host Group is the most effective way to manage an Oracle RAC environment.

Oracle RAC ASM Multi-Disk Online Method

Performing encryption with the online rebalancing method requires sufficient free space to allow the drop of the largest ASM disk.

Checking for Space

In the Oracle system, use the following commands to check for available disk space:

1. Check total free space in the disk group:

```
 SELECT name, free_mb, total_mb, free_mb/total_mb*100 as percentage
FROM v$asm_diskgroup;

NAME                                  FREE_MB   TOTAL_MB PERCENTAGE
------------------------------ ---------- ---------- ----------
DATA                                        7       2109 .331910858
```

2. Check individual ASM disk size and usage:

```
 select a.name DiskGroup, b.disk_number Disk#, b.name DiskName,
b.total_mb, b.free_mb, b.path, b.header_status FROM v$asm_disk b,
v$asm_diskgroup a where a.group_number (+) =b.group_number order
by b.group_number, b.disk_number, b.name

DISKGROUP  DISK#  DISKNAME        TOTAL_MB    FREE_MB
PATH                          HEADER_STATUS
---------  -----  --------       ---------    -------
------------------------------  -------------
DATA         0     DATA_0000          1874       1273   /dev/
oracleasm/disks/DATA3      MEMBER
DATA         1     DATA_0001          1992        608   /dev/
oracleasm/disks/DATA4      MEMBER
DATA         3     DATA_0003           117          0   /dev/
oracleasm/disks/DATA2      MEMBER
```

```
            0     DATA_ENC_0000        109          28    /dev/
oracleasm/disks/DATA1_ENC  MEMBER
```

Adding a Disk to the Diskgroup

Using the Online Method assumes that there is enough free space in the diskgroup so that you can drop/remove a disk, protect it with CTE, and then add it back into the diskgroup.

To add the disk to the diskgroup:

1. Open a terminal session on both RAC Nodes.

2. On **RAC Node 1**, on the ASM, remove the disk from the disk group, type.

   ```
   ALTER DISKGROUP <diskGroupName> DROP DISK <diskName> REBALANCE
   POWER 11 WAIT;
   ```

3. On the CipherTrust Manager, in the Host Group, apply a GuardPoint to the Raw Device: `<rawDevice1Name>`.

4. From **RAC Node 1**, to display the status of the guarded disks, type:

   ```
    secfsd -status guard
   ```

5. On both **RAC Node 1 and 2** type:

   ```
    chown oracle:oinstall /dev/secvm/<rawDevice1Name>
   chmod 660 /dev/secvm/<rawDevice1Name>
   ```

6. From **RAC Node, on the ASM**, add the protected disk to the disk group:

   ```
   ALTER DISKGROUP <diskGroupName> ADD DISK /dev/secvm/
   <rawDevice1Name> NAME <disk1Name>;
   ```

   The disk is now added to the diskgroup and ready for use.

7. The system is now ready for a reboot and failover test. For details, see Surviving the Reboot and Failover Testing.

Troubleshooting

Occasionally, settings do not persist when the system is rebooted. To ensure they do persist, edit the `/etc/rc.local` file and add the following lines:

```
Echo “Changing Permission for secvm devices”
chown oracle:oinstall /dev/secvm/dev/<rawDevice1Name>
chmod 660 /dev/secvm/dev/<rawDevice1Name>
```

Oracle RAC ASM Multi-Disk Offline Method (Backup/ Restore)

Using the Offline Method assumes that there is not enough free space in the diskgroup.

1. Open a terminal session on both RAC Nodes.

2. On RAC Node 1, on the ASM, type the following to remove the disk group.

   ```
   DROP DISKGROUP <diskGroupName> FORCE INCLUDING CONTENTS;
   ```

   > **Note**
   >
   > Make sure that the disk is removed before guarding the raw devices.

3. On the CipherTrust Manager, in the Host Group, apply GuardPoints to the three raw devices:

   ```
   <rawDeviceName1>
   <rawDeviceName2>
   <rawDeviceName3>
   ```

4. On **RAC Node 1**, to display the status of the guarded disks, type:

   ```
   secfsd -status guard
   ```

5. On both **RAC Node 1** and **2**, type:

CTE

```
chown oracle:oinstall /dev/secvm/<rawDeviceName1>
chmod 660 /dev/secvm/<rawDeviceName1>
chown oracle:oinstall /dev/secvm/<rawDeviceName2>
chmod 660 /dev/secvm/<rawDeviceName2>
chown oracle:oinstall /dev/secvm/<rawDeviceName3>
chmod 660 /dev/secvm/<rawDeviceName3>
```

CTE-U

```
chown oracle:oinstall <rawDeviceName1>
chmod 660 <rawDeviceName1>
chown oracle:oinstall <rawDeviceName2>
chmod 660 <rawDeviceName2>
chown oracle:oinstall <rawDeviceName3>
chmod 660 /dev/<rawDeviceName3>
```

6. From **RAC Node 1**, **on the ASM**, add the protected disk to the disk group, type:

```
 ALTER DISKGROUP <diskGroupName> ADD DISK /dev/secvm/
<rawDeviceName1> NAME <diskName1>;
 ALTER DISKGROUP <diskGroupName> ADD DISK /dev/secvm/
<rawDeviceName2> NAME <diskName2>;
 ALTER DISKGROUP <diskGroupName> ADD DISK /dev/secvm/
<rawDeviceName3> NAME <diskName3>;
```

The disks are now added to the diskgroup and ready for use.

7. On **RAC Node 1**, restore the database.

8. The system is now ready for a reboot and failover test. Go to the section Surviving the Reboot and Failover Testing.

Troubleshooting

Occasionally, settings do not persist when the system is rebooted. To ensure they do persist, edit the `/etc/rc.local` file and add the following lines:

```
“Changing Permission for secvm devices”
chown oracle:oinstall /dev/secvm/dev/<rawDeviceName1>
chmod 660 /dev/secvm/dev/<rawDeviceName1>
chown oracle:oinstall /dev/secvm/dev/<rawDeviceName2>
chmod 660 /dev/secvm/dev/<rawDeviceName2>
chown oracle:oinstall /dev/secvm/dev/<rawDeviceName3>
chmod 660 /dev/secvm/dev/<rawDeviceName3>
```

Surviving the Reboot and Failover Testing

Preparing for Failover Testing with ASMLib

When using ASMLib with the `createdisk` command, there is no requirement to make additional changes in `rc.local` or other areas for mapping device names or to use `chmod` or `chown` for SecVM. This is because it is managed for you by the `createdisk` function and you can verify this by running the following command:

```
ls -l /dev/oracleasm/disks
```

CTE Load Order and Startup Scripts

The last change is to ensure that CTE starts before ASM starts in the startup scripts.

Failover Testing

Confirm that everything is functional:

- Ensure that the GuardPoints are all operational.
- Ensure that you receive valid results when you query the database.

- Verify that the load order ensures that CTE starts before ASM.

Once verified, you can start the failover testing for each RAC Node.

1. Reboot the RAC Node 1 and monitor the startup.
2. Once the restart is clean, reboot RAC Node 2 and monitor the startup.

Issues with Device Mapper and Invalid Guard Path

If CTE is unable to apply a GuardPoint on a raw device, the logs may generate an error similar to the following:

```
[SecFS, 0] EVENT: Failed to guard /dev/mapper/devicename (reason: Inva
lid Guard Path flags 0x2 gypped 0x4 status 0x11) - Will retry later
```

If you receive this error, use the setup command to check the status of the disks, type:

```
setup info <deviceName>
```

Before attempting to establish a GuardPoint, look closely at the open count value and ensure that it is 0 on all nodes.

Using CTE with Oracle ASM Filter Driver

This chapter describes how to configure CTE on Oracle Standalone and RAC ASM Filter Driver for Linux. It contains the following topics:

- How to enable CTE for Oracle ASMFD
- How to configure CTE and create a guarded Oracle ASMFD
- How to convert a baseline ASMFD disk group to a guarded ASMFD disk group
- How to uninstall/ upgrade CTE with an active ASMFD Setup

About Oracle ASM Filter Driver

Oracle ASM Filter Driver (Oracle ASMFD) is a kernel module that resides in the I/O path of the Oracle ASM disks. Oracle ASM uses the filter driver to validate write I/O requests to Oracle ASM disks.

The Oracle ASMFD simplifies the configuration and management of disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted.

The Oracle ASM Filter Driver rejects any I/O requests that are invalid. This action eliminates accidental overwrites of Oracle ASM disks that would cause corruption in the disks and files within the disk group.

Enable Oracle ASMFD for CTE

To enable Oracle ASMFD:

1. Log in as the root user and stop the Oracle Grid Infrastructure for a standalone server, type:

   ```
   $GRID_HOME/bin/crsctl stop has
   ```

2. Configure Disk Discovery for AFD. Modify the following files by adding the content of `/dev/secvm/dev/sd*` to the OS files in `/etc/oracleafd.conf` and `/etc/afd.conf` with aft_filtering enabled, type:

   ```
       cat /etc/afd.conf
       afd_diskstring='/dev/sd*,/dev/secvm/dev/sd*'
       afd_filtering=enable

       cat /etc/oracleafd.conf
       afd_diskstring='/dev/sd*,/dev/secvm/dev/sd*'
       afd_filtering=enable
   ```

   a. **To prevent the file from changing back to `afd_diskstring='/dev/sd*'` make it read only, type:**

      ```
      chmod 444 /etc/oracleafd.conf /etc/afd.conf
      ```

3. Restart ACFS before proceeding, type:

```
 acfsload stop
acfsload start
```

4. Restart the Oracle Grid Infrastructure, type:

```
$GRID_HOME/bin/crsctl start has
```

5. Verify that `/dev/secvm/dev/sd*` is part of the SQL output and the state is enabled as a grid, type:

```
SELECT SYS_CONTEXT('SYS_ASMFD_PROPERTIES', 'AFD_DISKSTRING') FROM
DUAL;
```

Expected Response

```
SYS_CONTEXT('SYS_ASMFD_PROPERTIES','AFD_DISKSTRING')

------------------------------------------------------------------
--------------
/dev/sd*,/dev/secvm/dev/sd*
```

Note

If using Oracle RAC (Real Application Clusters), repeat the previous steps for all of the nodes in the related RAC cluster.

Configure CTE and create a guarded Oracle ASMFD

To configure CTE and create a Guarded Oracle ASMFD:

1. Create a raw device disk, for example:

```
/dev/sdc
```

2. Partition the targeted raw device, for example:

   ```
   /dev/sdc1
   ```

3. Guard the partitioned raw device, that you just created, using CTE.

4. Create an ASMFD disk on the guarded partitioned raw device, type:

   ```
    $GRID_HOME/bin asmcmd afd_label '<disk_label>'
   '<guarded_raw_partition>'
   ```

   Example:

   ```
   $GRID_HOME/bin asmcmd afd_label 'DISK1' '/dev/secvm/dev/sdc1'
   ```

5. Verify that the new disk was created, type:

   ```
   $GRID_HOME/bin asmcmd afd_lsdsk
   ```

   The disk should display in the list of disks, similar to the following:

   ```
    Label   Filtering   Path
   ================================================================
   ==============
   DISK1   ENABLED   /dev/secvm/dev/sdc1
   ```

6. If using Oracle RAC (Real Application Clusters), repeat the previous steps for all of the nodes in the RAC cluster.

7. Now you can use the Oracle Grid ASM Configuration Assistant GUI to create the targeted guarded ASMFD disk group on the above created guarded ASMFD disk, DISK1. See the Oracle GRID documentation for more information.

Convert a baseline ASMFD disk group to a CTE guarded ASMFD disk group using the Offline method

We will use the following example scenario in this section:

Objective:

- Convert the baseline ASMFD disk group **DATA1**, to a CTE guarded ASMFD disk group.
- Current configured baseline **DATA1** disk group consists of ASMFD disk **DISK1** on a raw partitioned device `/dev/sdc1`.

To convert:

1. If the targeted raw disk partition to be guarded does not yet exist, create it. For example: `/dev/sdd1`.

   > **Note**
   >
   > The capacity of `/dev/sdd1` must be large enough to transfer all of the data from ASMFD disk DISK1 on `/dev/sdc1`.

2. Using CTE, guard the raw partitioned device.

3. Create an ASMFD disk on the guarded raw partition:

   ```
   $GRID_HOME/bin asmcmd afd_label'<disk_label>'
   '<guardedRawPartition>
   ```

   Example

   ```
   $GRID_HOME/bin asmcmd afd_label 'DISK2' '/dev/secvm/dev/sdd1'
   ```

4. Ensure that there are current backups of all of the database(s) residing on the ASMFD disk group DATA.

5. Shutdown all databases residing on ASMFD disk group DATA1.

6. In the ASM database, run the following SQL command with the rebalance option to add a guarded ASMFD disk DISK2 to ASMFD disk group DATA1, type:

```
ALTER DISKGROUP <ASMFD diskgroup name> ADD DISK 'AFD:<ASMFD disk
name>' REBALANCE POWER 11 WAIT;
```

Example

```
ALTER DISKGROUP DATA1 ADD DISK 'AFD:DISK2' REBALANCE POWER 11
WAIT;
```

7. Verify the results, type:

col PATH for a15 col DG_NAME for a15 col DG_STATE for a10 col FAILGROUP for a10 select dg.name dg_name, dg.state dg_state, dg.type, d.disk_number dsk_no,d.path, d.mount_status, d.FAILGROUP, d.state from v$asm_diskgroup dg, v$asm_disk d where dg.group_number=d.group_number and dg.name = 'DATA1'

System Response:

```
DG_NAME DG_STATE TYPE   DSK_NO  PATH      MOUNT_S   FAILGROUP
STATE
----------------------  -------- -----    ---------    ----------
-----------
DATA1   MOUNTED  EXTERN 0   AFD:DISK1 CACHED    DISK1       NORMAL
DATA1   MOUNTED  EXTERN 0   AFD:DISK2 CACHED    DISK2       NORMAL
```

8. Run the following SQL command, in the ASM database, to transfer all data from ASMFD disk DISK1 to ASMFD disk DISK 2 and then drop ASMFD disk DISK1 from ASMFD disk group DATA1, type:

ALTER DISKGROUP DROP DISK 'AFD: REBALANCE POWER WAIT;

Example:

```
ALTER DISKGROUP DATA1 DROP DISK 'AFD:DISK1' REBALANCE POWER 11 WA
IT;
```

Verify the results:

```
col PATH for a15 col DG_NAME for a15 col DG_STATE for a10 col FAILGROUP for a10 select dg.name dg_name, dg.state dg_state, dg.type, d.disk_number dsk_no,d.path, d.mount_status, d.FAILGROUP, d.state from v$asm_diskgroup dg, v$asm_disk d where dg.group_number=d.group_number where dg.name = 'DATA1'
```

System response

```
DG_NAME     DG_STATE   TYPE   DSK_NO  PATH        MOUNT_S
FAILGROUP  STATE
---------- ---------- ------ ------ ---------   -------
---------  -------
DATA1       MOUNTED    EXTERN    0    AFD:DISK2   CACHED
DISK2      NORMAL
```

At this point, all data in the ASMFD disk group DATA1 has been transferred from the baseline ASMFD disk DISK1, to the guarded ASMFD disk DISK2. At this time, the databases that are residing in ASMFD disk group DATA1 can now be restarted.

Convert a baseline ASMFD disk group to a CTE guarded ASMFD disk group using the Online method

We will use the following example scenario in this section:

- Number of RAC nodes: 2
- Name of Diskgroup: DATA1
- Number of disks: 3

  Name of baseline disks assigned to raw device:

  - DISK1: /dev/sdb1
  - DISK2: /dev/sdc1
  - DISK3: /dev/sdd1

ASMFD disk group **DATA1** was created from baseline ASMFD disks DISK1, DISK2, and DISK3.

```
DG_NAME         DG_STATE   TYPE       DSK_NO PATH            MOUNT_S
FAILGROUP  STATE
--------------- ---------- ------ ---------- --------------- -------
---------- --------
DATA1           MOUNTED    EXTERN          0 AFD:DISK1       CACHED
DISK1      NORMAL
DATA1           MOUNTED    EXTERN          0 AFD:DISK2       CACHED
DISK2      NORMAL
DATA1           MOUNTED    EXTERN          0 AFD:DISK3       CACHED
DISK3      NORMAL
```

Objective:

- Convert all of the database data on ASMFD disk group DATA1 from baseline to a CTE guarded ASMFD disk group.

Note

The combined capacity of `/dev/sdc1` on DISK2 and `/dev/sdd1` on DISK3 must be large enough to hold all of the data transferring from ASMFD disk DISK1 on `/dev/sdb1`.

To convert:

From RAC Node 1:

1. Connect locally, as the system administrator, to an Oracle ASM instance using OS authentication, type:

   ```
   sqlplus 'as sysasm'
   ```

2. At the ASM prompt, drop the ASMFD disk DATA1 from the disk group DISK1, type:

   ```
   ALTER DISKGROUP DATA1 DROP DISK DISK1 REBALANCE POWER 11 WAIT;
   ```

3. Verify that the ASMFD DISK2 is no longer part of the ASMFD disk group DATA1, type:

   sqlplus '/as sysasm' col PATH for a15 col DG_NAME for a15 col DG_STATE for a10 col FAILGROUP for a10 select dg.name dg_name, dg.state dg_state, dg.type, d.disk_number dsk_no,d.path, d.mount_status, d.FAILGROUP, d.state from v$asm_diskgroup dg, v$asm_disk d where dg.group_number=d.group_number and dg.name = 'DATA1';

   System Response

   ```
   DG_NAME         DG_STATE   TYPE       DSK_NO PATH
   MOUNT_S FAILGROUP  STATE
   --------------- ---------- ------ ---------- ---------------
   ------- ---------- --------
   DATA1           MOUNTED    EXTERN          0 AFD:DISK2
   CACHED  DISK2      NORMAL
   DATA1           MOUNTED    EXTERN          0 AFD:DISK3
   CACHED  DISK3      NORMAL
   ```

> **Note**
>
> Depending on the size of your data, DISK1 might take some time to no longer show in the above query.

4. From the OS command prompt, drop ASMFD disk DISK1:

   ```
   $GRID_HOME/bin asmcmd afd_unlabel DISK1
   ```

5. From the command prompt, verify DISK1 is no longer visible on both RAC Node 1 and 2.

   ```
   $GRID_HOME/bin asmcmd afd_lslbl
   ```

6. Using CTE, guard the DISK1 raw partitioned device `/dev/sdb1`.

7. From the command prompt, recreate ASMFD disk DISK1, with the guarded raw device `/dev/sdb1`.

CTE

```
$GRID_HOME/bin asmcmd afd_label 'DISK1' '/dev/secvm/dev/sdb1'
```

CTE-U

```
$GRID_HOME/bin asmcmd afd_label 'DISK1' '/dev/secvmsdb1'
```

8. From the command prompt, verify that DISK1 is now visible on both RAC Node 1 and 2 with the newly guarded path:

   ```
   $GRID_HOME/bin asmcmd afd_lslbl
   ```

9. From the ASM prompt, add the guarded ASMFD disk DISK1 back to the ASMFD disk group DATA1, type:

   ```
    sqlplus 'as sysasm'
    ALTER DISKGROUP DATA1 ADD DISK 'AFD:DISK1' REBALANCE POWER 11 WA
   IT;
   ```

10. Verify that ASMFD disk DISK1 is now part of ASMFD diskgroup DATA1, type:

    sqlplus '/as sysasm' col PATH for a15 col DG_NAME for a15 col DG_STATE for a10 col FAILGROUP for a10 select dg.name dg_name, dg.state dg_state, dg.type, d.disk_number dsk_no,d.path, d.mount_status, d.FAILGROUP, d.state from v$asm_diskgroup dg, v$asm_disk d where dg.group_number=d.group_number and dg.name = 'DATA1';

    System Response

    ```
    DG_NAME         DG_STATE   TYPE       DSK_NO PATH
    MOUNT_S FAILGROUP  STATE
    --------------- ---------- ------ ---------- ---------------
    ------- ---------- --------
    DATA1           MOUNTED    EXTERN          0 AFD:DISK1
    ```

```
CACHED  DISK1      NORMAL
DATA1           MOUNTED    EXTERN           0 AFD:DISK2
CACHED  DISK2      NORMAL
DATA1           MOUNTED    EXTERN           0 AFD:DISK3
CACHED  DISK3      NORMAL
```

Note

Depending on the size of your data, DISK1 might take some time to display in the above query.

11. Repeat the previous steps for Baseline DISK 2 and DISK 3

Uninstall/ Upgrade CTE with active ASMFD Setup

For an Active ASMFD Standalone Setup

1. Shutdown all databases running on guarded devices.

2. Shutdown the CRS and ACFS, type:

```
crsctl stop resource -all
crsctl stop has
afdload stop
```

3. Perform CTE maintenance, uninstall or upgrade.

4. Restart ACFS and CRS, type:

```
afdload start
crsctl start has
crsctl start resource -all
```

5. Restart related databases.

For an Active ASMFD RAC Setup

To perform CTE maintenance on all ASMFD RAC nodes in a cluster:

1. Shutdown all databases running on guarded devices.
2. Shutdown CRS, ACFS, and AFD on all nodes, type:

   ```
    $GRID_HOME/bin/crsctl stop cluster -all
   $GRID_HOME/bin/acfsload stop cluster -all
   $GRID_HOME/bin/afdload stop cluster –all
   ```

3. Perform CTE maintenance, uninstall, or upgrade on all targeted RAC nodes.
4. Restart AFD, ACFS, and CRS on all nodes, type:

   ```
    $GRID_HOME/bin/afdload start cluster -all
   $GRID_HOME/bin/acfsload start cluster -all
   $GRID_HOME/bin/crsctl start cluster –all
   ```

5. Restart related databases.

To perform CTE maintenance on a ASMFD RAC cluster in a rolling fashion:

1. Shutdown all instances running on guarded devices on the targeted RAC node only.
2. Shutdown CRS, ACFS, and AFD on the targeted RAC node only. Type:

   ```
    $GRID_HOME/bin/crsctl stop has
   $GRID_HOME/bin/acfsload stop
   $GRID_HOME/bin/afdload stop
   ```

3. Perform CTE maintenance, uninstall, or upgrade on the targeted RAC node only.

4. Restart AFD, ACFS, and CRS on a targeted RAC node only, type:

```
$GRID_HOME/bin/afdload start
$GRID_HOME/bin/acfsload start
$GRID_HOME/bin/crsctl start has
```

5. Restart related instances on the targeted RAC node.

Note

If AFD fails to stop, try the following OS command:

```
modprobe -r oracleafd
```

Using CTE LDT with Oracle

Oracle compatibility with LDT

Oracle is compatible with LDT with the following changes:

You must run Oracle with the following file system IO option, when mounted on NFS:

- DirectIO

Note

LDT is not supported on raw devices such as ASM.

Basic Troubleshooting Techniques

The following are some of the most common configuration issues that prevent the Oracle ASM configuration from working properly.

If you encountering errors similar to:

- ORA-15075: disk(s) are not visible cluster-wide
- ORA-15032: not all alterations performed

This could be the result of improper settings for the I/O layer, meaning that your disks are not properly configured.

Perform the following tasks to verify that the settings are correct:

1. On the CipherTrust Manager, in the Host Group that was created for the RAC cluster, verify that the host group for this configuration does **NOT** have the Cluster Group option set (this option is only for GPFS, which is not supported with CTE).

2. Ensure that the GuardPoints for the block devices are set at the Host Group level. This ensures that each node receives identical GuardPoints.

3. Verify that the GuardPoints are active on all nodes. When the GuardPoints are set, go to each node and verify that they are set and guarded, using the WebUI or the `secfsd -status guard1` command. If they do not guard correctly:

    - Verify that the udev rules are set correctly, as described in Modify the UDEV Rules
    - Make sure the device names are the same across all nodes.

4. From ASM, make sure that the `asm_diskstring` parameter is modified to include the CTE devices and that the proper pathname is used, see Altering ASM_DISKSTRING on ASM.

Verifying Database Encryption

Option 1

The best way to verify the state of the data, without impacting anything in the existing environment, is to use the Oracle kfed command. You can run this command against the native path of the existing GuardPoints and make sure it returns with valid header information. If it returns valid information with the GuardPoint in place, then this confirms that the data is properly encrypted. If it returns with invalid header information, then that indicates that the data is either clear, double encrypted, or not in the expected encrypted state. The syntax for running this command would look similar to the following but will vary based on your environment.

```
 /app/oracle/grid/product/19.0.0/grid/bin/kfed read /dev/rdisk/
<diskName>
```

If the location is properly encrypted, following is an example of the viewable output:

```
 /app/oracle/grid/product/19.0.0/grid/bin/kfed read /dev/rdisk/
<diskName>
```

System Response:

```
 kfbh.endian:                           1 ; 0x000: 0x01
kfbh.hard:                            242 ; 0x001: 0xf2
kfbh.type:                            124 ; 0x002: *** Unknown Enum ***
kfbh.datfmt:                           66 ; 0x003: 0x42
kfbh.block.blk:                1088904227 ; 0x004: blk=1088904227
kfbh.block.obj:                1558192170 ; 0x008: file=8234
kfbh.check:                    3321251423 ; 0x00c: 0xc5f6465f
kfbh.fcn.base:                  932956641 ; 0x010: 0x379bc9e1
kfbh.fcn.wrap:                 3040493590 ; 0x014: 0xb53a4016
kfbh.spare1:                   3806015223 ; 0x018: 0xe2db2ef7
kfbh.spare2:                   3794962182 ; 0x01c: 0xe2328706
60000000000D8000 01F27C42 40E75C23 5CE0202A C5F6465F
[..|B@.\#\. *..F_]
60000000000D8010 379BC9E1 B53A4016 E2DB2EF7 E2328706  [7....:@......
2..]
60000000000D8020 CA2F30AD 522B4D21 99292639 004EBB34  [./0.R+M!.)&9.N.
4]
60000000000D8030 A3896BE8 BD839D23 2204E19E 946C575C
[..k....#"....lW\]
60000000000D8040 4CE2218F 35E1B101 AF751A70 780E6D6E  [L.!.
5....u.px.mn]
60000000000D8050 5E7E6A38 C600ED5F 929047C4 DF372A8E  [^~j8..._..G..
7*.]
60000000000D8060 E103152D BA87CC03 11A7D963 9D72FCE1
[...-.......c.r..]
60000000000D8070 1EC6B48B 03EE869F 61D651F9 E7614957
[........a.Q..aIW]
60000000000D8080 810E0353 9C461F49 69569733 501D19EF  [...S.F.IiV.
```

```
3P...]
60000000000D8090 B268002B 4F9457B6 BDB04AC5 D3D07446  [.h.
+O.W...J...tF]
60000000000D80A0 FD9EE5E0 9B46CB66 30D10B22 F63AB77E
[.....F.f0..".:.~]
60000000000D80B0 6FF79075 4BBD1FAD 8F226188 7774300D
[o..uK...."a.wt0.]
60000000000D80C0 A809B6FB E1F1C80B B5760E68 9747D97D
[........v.h.G.}]
KFED-00322: Invalid content encountered during block traversal: [kfbtT
raverseBlock][Invalid OSM block type][][124]
```

Option 2

The second option to verify the state of the data is to use the dd command. This requires you to specify some blocks and write it out to a file. After this completes, read the file using the strings command. You can do this while the device is in use. In the example below some sectors are skipped and it only dumps 10000 counts.

For example:

```
dd if=/dev/mapper/asm_data2p1 of=/tmp/dd2.out skip=1047 count=10000
```

Option 3

The third option to verify the state of the data without impacting the existing environment is to use the strings command.

Note

The strings command cannot read a busy or large device.

You can run this command against the native path (`/dev/<deviceName>`) of the existing GuardPoints (`/dev/secvm/dev/<deviceName>`). The new path to SecVM would be similar to `/dev/secvm/dev/<deviceName>`. By executing the strings command against the native path `strings /dev/devicename | more`, this does not go through the SecVM device and therefore is not be decrypted. If it is encrypted the output should contain illegible text.

Using Oracle with CTE-U

When guarding an ASM disk with CipherTrust Transparent Encryption UserSpace, there can be a 10% drop in database performance. This occurs when the underlying device is in Oracle direct mode. This mode avoids double-caching, which consumes memory, and therefore impacts performance on some workloads.

This issue is affected by the `block_o_direct` parameter. If that parameter is set to `0`, the IO will be reported as complete, (meaning committed to disk) before it actually is. This is a data integrity risk for a database, because the normal mode is to mark as journal complete as soon as the data is safe on a disk. This incorrect reporting could result in a race that could lead to a journal being cleared before the data is on the disk. If a crash occurs during this window, the data would be lost.

In this scenario, data integrity and performance are at odds. You must choose which one to highlight.

1. To set CTE-U for **optimal data integrity**, type the following command:

   ```
   voradmin secfs config block_o_direct 1
   ```

2. To set CTE-U for **optimal performance**, type the following command:

   ```
   voradmin secfs config block_o_direct 0
   ```

Note

The commands listed in the steps above will change the `block_o_direct` parameter in CipherTrust Transparent Encryption UserSpace, in the `config.yaml` file located in: `/opt/vormetric/DataSecurityExpert/agent/secfs/.sec/conf/secfs_config.yaml`

Integrating and Configuring EDB

This chapter describes how to configure and integrate Enterprise DB Postgres Advanced Server (EDB) with CTE in Linux. It contains the following sections:

- Preparing to Create GuardPoints
- Integration with CTE

Overview

EDB provides database management software to get more functionality from PostgreSQL. EDB offers secure, scalable, advanced and enterprise-class PostgreSQL solutions. Advanced Server extends PostgreSQL with the security and performance features that enterprises need. It’s also compatible with Oracle databases.

Prerequisites

Refer to EDB documentation for information on how to setup and configure EnterpriseDB.

Preparing to Create GuardPoints

Before you can safely reboot your machine, create GuardPoints, and administer any of the services, you must perform the following steps:

1. Stop the EDB service.
2. Open the `/lib/systemd/system/edb-as-13.service` file.
3. Below the [Unit] section, add the following line: `Requires=secfs-fs-barrier.service`.
4. Save the file and exit.
5. Open the file: `/lib/systemd/system/secfs-fs-barrier.service`.
6. At the end of the [Before] clause, add `edb-as-13.service`.
7. Save the file and exit.
8. Type: `systemctl daemon-reload`.
9. Type: `systemctl start edb-as-13.service`.
10. Reboot the system.
11. Stop the EDB.
12. Restart SecFS.
13. Start the EDB service again.

Integration with CTE

The following describes how to integrate EDB with CTE using an offline dataxform policy:

1. Enable and start EDB.
2. Create a database with sample data.
3. Install the latest CTE build. Refer to the CTE Agent Linux Quick Start Guide for more information.
4. Register CTE with a CipherTrust Manager.
5. Stop the EDB service.
6. Guard the EDB data and log directories with a Dataxform policy.
   For example, create a policy that transforms from Clear key to AES256.
7. Perform a data transformation on both directories.
8. After Dataxform finishes, guard both directories with a policy using the correct key.
   For example, use AES256 on both directories.
9. After the database and log directories are guarded, start the EDB service again.

Integration with Pacemaker

This document covers the following information:

- Using CTE with Pacemaker
- Performing CTE maintenance with LDT GuardPoints and a Pacemaker Setup

Performing CTE maintenance with an LDT GuardPoint in a Pacemaker Setup

Disable the LDT GuardPoint with Pacemaker setup

1. Stop the SQL server, type:

   ```
   # sudo systemctl stop mssql-server
   ```

2. Delete the existing resource constraints from LDT and the colocation from the Pacemaker setup.

   Example

   A. If it exists, get the ID of the PCS constraint co-location. In the example below the, ID is: `co-location_set_msmsmt`

   ```
   # sudo pcs constraint show --full
   .
   .
   .
   Resource Sets:
   set mssql_fs mssql_gpfs mssql_gpldt sequential=false (id:co-location_set_msmsmt_set) setoptions score=INFINITY (id:co-location_set_msmsmt)
   ```

   B. Remove the PCS constraint co-location and PCS resource, type:

   ```
   # sudo pcs constraint remove co-location_set_msmsmt
   # sudo pcs resource delete mssql-ldtmgp
   ```

   Expected Result: Current LDT GuardPoint status should be now be: `unguarded`.

   Example

```
# secfsd -status guard
```

Response

```
GuardPoint          Policy                          Type
ConfigState  Status       Reason
----------          ------                          ----
-----------  ------       ------
/var/opt/mssql/LDT  LDT_clear_to_CS1_aes256_onhost  manual
unguarded    not guarded  Inactive
```

3. Disable the LDT GuardPoint from the Cluster Host Group in CipherTrust Manager.

   Expected Result: The LDT GuardPoint should no longer be displayed when `secfsd -status guard` is executed.

Perform CTE maintenance

Now you can perform CipherTrust Transparent Encryption maintenance tasks that require stopping secfs such as:

- Stop CipherTrust Transparent Encryption
- Upgrade CipherTrust Transparent Encryption
- Uninstall CipherTrust Transparent Encryption
- Create a new baseline database for LDT on the same LDT &{gp}

See CTE Agent for Linux Advanced Configuration for more information.

Note

Before creating a new baseline database for CTE-LDT: Ensure that your target LDT GuardPoint path does not have an `ldt xattr` value on it so that the directory can trigger the initial rekey at the start of guarding. If it does have an `ldt xattr` value from a previous setup, then use the command `voradmin ldt xattr delete <LDT GuardPoint>` to remove the old xattr value.

Re-Enable LDT GuardPoint with Pacemaker Setup

1. Re-enable the LDT GuardPoint from the Cluster Host Group in CipherTrust Manager. Wait until the GuardPoint is visible again and in the `unguarded` state before moving to the next step below.

   Example

   ```
   # secfsd -status guard
   ```

   Response

   ```
   GuardPoint          Policy                            Type
   ConfigState  Status       Reason
   ----------          ------                            ----
   -----------  ------       ------
   /var/opt/mssql/LDT  LDT_clear_to_CS1_aes256_onhost  manual
   unguarded    not guarded  Inactive
   ```

2. Add the resource `mssql-ldtmgp` back into the Pacemaker setting and verify that it has started:

   Example

   ```
   # sudo pcs resource create mssql-ldtmgp ocf:heartbeat:mgp mgpdir=/
   var/opt/mssql/LDT --group Apache_Grp-fs
   ```

   The LDT GuardPoint should now be automatically active again in Pacemaker.

   ```
   # secfsd -status guard
   ```

   Response

   ```
   GuardPoint          Policy                            Type
   ConfigState  Status   Reason
   ----------          ------                            ----
   -----------  ------   ------
   ```

```
/var/opt/mssql/LDT  LDT_clear_to_CS1_aes256_onhost  manual
guarded      guarded  N/A
```

3. Start SQL server, type:

```
# sudo systemctl start mssql-server
```

4. Add the LDT resource constraints and co-location back into the Pacemaker setting.

   Example

   A. Add failover resource constraints back:

```
    # sudo pcs constraint order start mssql_gpldt then start
mssql-ldtmgp

    # sudo pcs constraint order stop mssql-ldtmgp then stop
mssql_gpldt
```

   B. If constraint co-location was used, add it back:

```
    # sudo pcs constraint co-location set mssql_fs mssql_gpfs
mssql_gpldt sequential=false
```

5. Restart Pacemaker so that it can pick up the newly created resource configurations.

   A. Stop SQL server, type:

```
# sudo systemctl stop mssql-server
```

   B. Reboot Pacemaker, type:

```
# sudo systemctl restart pacemaker
```

   C. Re-start SQL server, type:

```
# sudo systemctl start mssql-server
```

Using CTE with Pacemaker

This chapter describes how to configure CTE with Pacemaker and MySQL on Red Hat 7 or Red Hat 8. It contains the following sections:

- Overview
- Considerations and Requirements
- Creating GuardPoints

Overview

Pacemaker is a high-availability cluster resource manager that runs on a set of hosts (a cluster of nodes) in order to preserve integrity and minimize downtime of desired services (resources). Every resource has a resource agent that abstracts the service it provides and present a consistent view to the cluster.

Thales provides a resource agent for CTE that allows CTE to guard nodes running MySQL databases over an NFS share in a Pacemaker environment.

Considerations and Requirements

- System Requirements:
    - Red Hat 7 or Red Hat 8. Other versions of Red Hat are not supported.
    - Pacemaker with Corosync and the MySQL database service configured. Make sure that the Pacemaker properties settings for features such as STONITH and quorum are correct based on your Pacemaker environment.

> **Note**
>
> Other database applications may be used instead of MySQL, but Thales has only tested the CTE resource agent with MySQL.

- The CTE resource agent supports manual GuardPoints on a device or folder. Automatic GuardPoints are not supported.
- The CTE resource agent supports GuardPoints created from Standard or Live Data Transformation policies. It does not support IDT-Capable GuardPoints or CTE-Efficient StorageGuardPoints.

- If you install Pacemaker after you have installed CTE, you must copy the CTE resource agent to the `ofc` directory.

  ```
  cp /opt/vormetric/DataSecurityExpert/agent/secfs/systemd/
  pacemaker_ra_mgp  //CTE-U
  cp /opt/vormetric/DataSecurityExpert/agent/secfs/.sec/bin/
  pacemaker_ra_mgp //CTE
  ```

- Upgrades to the CTE resource agent will require using Pacemaker to put each node into maintenance mode while the upgrade is performed.

Creating GuardPoints

1. Install the CTE Agent on each node in the Pacemaker cluster, and register that node in the same domain in your key manager. For details, see Getting Started with CTE for Linux.

2. In your key manager, do the following:

   - Create a manual GuardPoint for the MySQL data directory `/var/lib/mysql/` on all of the nodes in the cluster. You can use either a Standard or a Live Data Transformation policy to create the GuardPoint, but you must use the same policy on the GuardPoint for each node.

   - Create any other manual GuardPoints you want to use. Each GuardPoint must be created on all nodes in the cluster and each set of GuardPoints must use the same policy.

   For example, if you have three nodes in the cluster and you want to add a Standard GuardPoint for `/hr/shared/files` and a Live Data Transformation GuardPoint for `/dir/accounting/data`, on *each* of the three nodes in the cluster you would create:

   - A Standard or Live Data Transformation GuardPoint for `/var/lib/mysql/`.
   - A Standard GuardPoint for `/hr/shared/files`.
   - A Live Data Transformation GuardPoint for `/dir/accounting/data`.

3. Select one of the nodes in your cluster and log into that node as `root`.

4. On the selected node, add a Pacemaker resource for the manual GuardPoint using the following command:

```
 pcs resource create mysql-mgp ocf:heartbeat:mgp mgpdir=/var/lib/m
ysql \
[start services=true] [stop_services=false]
```

where:

a. `start_services=true` is an optional command that tells the resource agent to start the CTE services before enabling any manual GuardPoints. The options are `true` or `false`, and the default is true.

b. `stop_services=false` is an optional command that tells the resource agent to stop the CTE services after disabling any manual GuardPoints. The options are `true` or `false`, and the default is false.

5. Create the required resource groups, colocation settings, and constraints.

> **Warning**
>
> **If the resource groups and colocation contstraints are not configured properly, the PCS cluster could run resources on multiple nodes. If the ordering constraints are not set properly, the cluster could fail to start because the system tries to mount the GuardPoint before the file system has been mounted or tries to unmount the file system before the GuardPoint has been disabled. Make sure that the following resource groups and constraints are set properly**.

To do so:

a. Create a MySQL file system group (`mysql-fsg`) that contains `mysql-fs` and `mysql-mgp` using the following command:

```
pcs resource group add mysql-fsg mysql-fs mysql-mgp
```

b. Create a MySQL service group (`mysql-sg`) that contains `mysql-server` and `mysql-vip` using the following command:

```
 pcs resource group add mysql-sg mysql-vip mysql-server
```

c. Configure the colocation and ordering constraints so that:

- `mysql-sg` **and** `mysql-fsg` **are colocated.**
- `mysql-fs` **starts before** `mysql-mgp`**.**
- `mysql-mpg` **stops before** `mysql-fs`**.**
- `mysql-fsg` **starts before** `mysql-sg`**.**

To do so, use the following commands:

```
pcs constraint colocation add mysql-sg with mysql-fsg
pcs constraint order start mysql-fs then start mysql-mgp
pcs constraint order stop mysql-mgp then stop mysql-fs
pcs constraint order start mysql-fsg then start mysql-sg
```

6. To check the status of the cluster after the configuration is complete, use the `pcs status` command.

Using CTE with SQL Server on Linux on Red Hat 8

Considerations and Requirements

- System Requirements:
  - **SQL Server 2019**. Other versions of SQL Server are not supported with CipherTrust Transparent Encryption.
  - **Red Hat 8**. Other versions of Red Hat are not supported with CipherTrust Transparent Encryption.

Note

Other database applications may be used instead of MySQL, but Thales has only tested this feature with MySQL.

- CipherTrust Transparent Encryption agent supports manual GuardPoints on a device or folder for both standalone and cluster SQL Server database. Automatic GuardPoints are supported on a standalone SQL Server database. Automatic GuardPoints are supported on a standalone SQL Server database but not for cluster SQL Server databases.

- The CipherTrust Transparent Encryption resource agent supports GuardPoints created from standard or Live Data Transformation policies. It does not support IDT-Capable GuardPoints or in-place Device GuardPoint GuardPoints.

Create a CTE guarded standalone SQL Server database

1. Configure the target database data and log directory location as mssql, type:

   sudo /opt/mssql/bin/mssql-conf set /u01/app/mssql/data sudo /opt/mssql/bin/mssql-conf set /u01/app/mssql/log sudo systemctl restart mssql-server

2. Guard targeted directory with standard encryption policy using CS1 or CBC keys of either 128aes or 256aes encryption.

   Example

   Policy Type: **Standard**

Rule	Value
** Security Rules**	
Action	all_ops
Effect	permit, audit, applykey
** Key Selection Rule**	
Key	aes128 or 256 with CS1 or CBC

```
# secfsd -status guard
```

Response

```
    GP                        Policy                    Type
ConfigState  Status   Reason
    ----------                ------                    ----
-----------  ------   ------
    /u01/app/mssql/data     encrypt_cs1_aes128_all   local
guarded      guarded  N/A
    /u01/app/mssql/data     encrypt_cbc_aes256_all   local
guarded      guarded  N/A
```

3. Create a target SQL Server database on a guarded directory as mssql, type:

   ```
   sqlcmd -S localhost -U SA -P '<password>'
   CREATE DATABASE <targetdb>
   go
   ```

4. Verify that data and log files are residing in the newly designated directory:

   ```
   use <targetdb>
   SELECT filename from sysfiles
   GO
   ```

Expected results: Should see data/log file in a guarded directory:

```
    /u01/app/mssql/data/testdb.mdf
   /u01/app/mssql/log/testdb_log.ldf
```

Convert a baseline SQL Server database to a CTE GuardPoint using Dataxform

1. Shutdown the SQL Server service, type:

   ```
   # sudo systemctl stop mssql-server
   # systemctl status mssql-server
   ```

2. Create an online policy using CS1 or CBC keys of either 128aes or 256aes encryption.

   Example of aes256

   Policy Name: `dxf_clear_to_cbc_aes250_onhost`

   Policy Type: Standard

Rule	Value
** Security Rules**	
Action	key_ops

Rule	Value
Effect	permit, applykey
** Key Selection Rule**	
Key	clear_key
Data Transformation	
Key	cbc_aes256_onhost

3. Add the GuardPoint with the policy `dxf_clear_to_aes256_onhost` to the targeted baseline database data/log directory, type:

   `# secfsd -status guard`

   Response

```
GuardPoint             Policy                          Type
ConfigState  Status   Reason
----------            ------                          ----
-----------  ------   ------
/u01/app/mssql/log    dxf_clear_to_cbc_aes250_onhost  local
guarded      guarded  N/A
/u01/app/mssql/data   dxf_clear_to_cbc_aes250_onhost  local
guarded      guarded  N/A
```

4. Run dataxform with the above policy as the root user. cl :::text dataxform --rekey --cleanup_on_success --gp /u01/app/mssql/dataxform/data dataxform --rekey --cleanup_on_success --gp /u01/app/mssql/dataxform/log

   Example

```
# dataxform --rekey --eanup_on_success --gp /u01/app/mssql/
dataxform/data
```

   Response

```
Checking if `/u01/app/mssql/dataxform/data` is a guard point with
a rekey policy applied
`/u01/app/mssql/dataxform/data` is a guard point with a rekey
policy applied
About to perform the requested data transform operation
```

```
-- Be sure to back up your data
-- Please do not attempt to terminate the application
Do you wish to continue (y/n)?y
.
.
.
Data transform for guard point /data1/oracle/oradata/xformdb
finished
About to remove the data transformation status files
Do you wish to continue (y/n)?y
Removal of data transformation status files completed
```

5. Unguard GuardPoints `/u01/app/mssql/dataxform/data` and `/u01/app/mssql/dataxform/data` from policy `dxf_clear_to_aes256_onhost`.

6. Guard these GuardPoints again with a standard policy which encrypts using the same open aes256 key policy as `dxf_clear_to_cbc_aes256_onhost`.

```
# secfsd -status guard
```

Response

```
GuardPoint             Policy                 Type    ConfigState
Status    Reason
----------             ------                 ----    -----------
------    ------
/u01/app/mssql/data    encrypt_cbc_aes256_all  local   guarded
guarded   N/A
/u01/app/mssql/log     encrypt_cbc_aes256_all  local   guarded
guarded   N/A
```

7. Start the database and then test to see if the data is accessible, type:

```
# sudo systemctl start mssql-server
# sqlcmd -S localhost -U SA -P '<password>'
use <targetdb>
SELECT filename from sysfiles
go
```

Expected results: You should be able to log into your targeted transformed database and query system files.

Create a CTE guarded Standalone SQL Server database using LDT

1. Enable the LDT setting in CipherTrust Manager using one of the following methods:

   - Install agent to your targeted database host with LDT enabled Configuring CTE with CipherTrust Manager.
   - Enable LDT on an existing CipherTrust Transparent Encryption installed host using the CM GUI.

2. Create a targeted new baseline in the SQL Server database:

   Example

```
sudo /opt/mssql/bin/mssql-conf set filelocation.defaultdatadir /
var/opt/mssql/LDT/data
sudo /opt/mssql/bin/mssql-conf set filelocation.defaultlogdir /
var/opt/mssql/LDT/data
sudo systemctl restart mssql-server
systemctl status mssql-server
sqlcmd -S localhost -U SA -P '<password>'
CREATE DATABASE `<targetdb>`
SELECT Name from sys.Databases
go
```

3. Create an LDT encryption policy using CS1 or CBC keys of either 128aes or 256aes.

   Example

Field	Value
Name	LDT_clear_to_CS1_aes256_onhost_key
Expiration Date	`<target date>`
Algorithm	AES256 or AES128
Encryption Mode	CBC_CS1 or CBC
Key Type	Cache on Host
Automatic Key Rotation	Selected
Key Version Life Span (days)	`<target #>`

4. Create an online LDT policy using the key created in the previous step.

   Example for CBC_CS1 with AES256

   Policy Name: LDT_clear_to_CS1_aes256_onhost

   Security Rules

Rule	Action
Order 1	
Action	key_op
Effect	Permit, Apply Key
Browsing	No

Rule	Action
Order 2	
Action	all_ops
Effect	Permit, Audit, Apply Key
Browsing	Yes

Rule	Action
Order 3	
Action	all_ops
Effect	Deny, Audit
Browsing	Yes

Key Selection Rule:

Key	Description
Order 1	
Current Key	clear_key
Transformation Key	LDT_clear_to_CS1_aes256_onhost_key
Exclusion Rule	No

5. Stop the SQL Server, type:

```
# sudo systemctl stop mssql-server
# systemctl status mssql-server
```

6. Guard the new LDT baseline database with the LDT policy created in the previous step.

```
# secfsd -status guard
```

Response

```
GuardPoint                          Policy
Type        ConfigState   Status    Reason
----------                  -------
----        -----------   ------    ------
/var/opt/mssql/LDT/data   LDT_clear_to_CS1_aes256_onhost
local       guarded       guarded   N/A
```

Wait for the Rekey Status in CipherTrust Manager to display: **Rekeyed**.

7. Run LDT with the same policy to verify the transformation of the baseline database.

```
# voradmin ldt attr get <data/log directory of baseline database>
```

Example:

```
# voradmin ldt attr get /var/opt/mssql/LDT/data
```

Response

```
LDT stats: version=6, rekey_status=rekeyed
    Number of times rekeyed:                  1 time
    Rekey start time:                         2022/09/02 05:41:26
    Last rekey completion time:               2022/09/02 05:41:28
    Estimated rekey completion time:          N/A
    Policy key version:                       0
    Pushed Policy key version:                0
    Policy ID:
        1001
    Data stats:
        total=16.0MB, rekeyed=11.0MB
        truncated=0.0MB, sparse=5.0MB
    File stats:
        total=2, rekeyed=2, failed=0
        passed=0, skipped=0, created=0, removed=0, excluded=0
```

Expected results:

- No errors in output
- rekey_status=rekeyed

> **Note**
>
> Do not start the next step until rekey_status = rekeyed.

8. Restart the SQL Server, type:

```
# su - mssql
# sudo systemctl start mssql-server
# systemctl status mssql-server
```

9. Access the targeted database and query the data, type:

```
sqlcmd -S localhost -U SA -P '<password>'
use <targetdb>
use <targetdb>
```

```
SELECT filename from sysfiles
go
```

Expected results: You should be able to log into your targeted database and query system files.

Configuring Support for SAP HANA

This section describes SAP HANA, which provides automatic host-failover support. It contains the following topics:

- Overview
- Customizing CTE for SAP HANA in HA Mode
- Using SAP HANA with LDT
- Setting Memory Allocation

Overview

SAP HANA provides automatic host-failover support. CTE works with HANA fibre storage systems to enable and disable GuardPoints when a protected host starts, stops, or fails over to standby host.

CipherTrust Transparent Encryption - SAP HANA (CTE-SAP HANA) supports non-shared storage where each HANA node has its own separate storage volumes. CTE-SAP HANA provides customized scripts to support startups, shutdowns, and fail overs.

HANA attaches logical unit number (LUNs) or logical volume management (LVMs) using a Fibre Storage Connector (fcClient) providers. Thales provides hooks that are called by the HANA fcClient providers that manage guarding or unguarding of storage locations.

> **Note**
>
> Thales recommends using host groups to manage configuring in a clustered host environment.

Customizing CTE for SAP HANA in HA Mode

Caution

The following procedure only applies if multiple HANA nodes are configured in a high availability (HA) environment. If you are installing CTE on a single HANA node, do not change the default HANA or CTE settings.

1. Go to the installation directory:

   ```
   cd /opt/vormetric/DataSecurityExpert/agent/secfs/saphana
   ```

2. If required, edit the appropriate CTE `fcClient` refined script:

Script file	Use
`fcClientRefinedVTE.py`	fcClient provider for LUN
`fcClientLVMRefinedVTE.py`	fcClientLVM provider for LVMs

3. Copy the appropriate script file to a shared location that is accessible to all nodes. In a HANA cluster environment, all nodes require access to the CTE scripts.

4. Edit the storage section of the `global.ini` file to indicate the corresponding CTE `fcClient` as the High Availability (HA) provider and to point to the location of the CTE script.

   LUN Example

   ```
   [storage]
      ha_provider = fcClientRefinedVTE
      ha_provider_path = /hana/shared/myFcClient
   ```

   If necessary, enable debug tracing:

   ```
   [trace]
      ha_provider = debug
   ```

```
   ha_fcclient = debug
   ha_fcclientrefinedvte = debug
```

LVM Example

```
[storage]
   ha_provider = fcClientLVMRefinedVTE
   ha_provider_path = /hana/shared/myFcClient
```

If necessary, enable debug tracing:

```
[trace]
   ha_provider = debug
   ha_fcclientlvm = debug
   ha_fcclientlvmrefinedvte = debug
```

5. Use the same CTE agent with all hosts, including standby hosts.

6. Ensure that `/etc/sudoers` includes the following:

   ```
   <sid>adm ALL=NOPASSWD: /usr/bin/secfsd
   ```

7. Enable the guard paths at the mount-point level.

   For example, individual guards were placed on `/hana/data/HAN/mnt00001, /hana/data/HAN/mnt00002`, and so forth.

   a. Use a similar naming practice for log partitions, such as `/hana/log/HAN/mnt00001`, and so forth.

   b. Place the guard at the mount-point level. The guarded paths must match the corresponding data and log mount paths.

8. Configure GuardPoints as type `manual`. You must enable and disable the GuardPoints immediately after the device is attached, or just prior to detachment. The reason for manual GuardPoints is that it invokes guarding and unguarding from within the HANA during the startup, shutdown, or failover process. The process resembles that of mounting and unmounting guarded auto-mount points.

9. Configure all GuardPoints so that they are available in the standby host, so that any data and log partitions that fail over from any host can be guarded on the standby.

   Thales recommends that you configure all GuardPoints on all hosts, because a failed-over active host can then become the new standby, and will require all available GuardPoints.

Example

The following is an example of the data and log volumes for the host that are mounted.

```
 /dev/mapper/VG_HAN_DATA_1-LV_HAN_DATA_1       793971096  3059712
750579916   1% /hana/data/HAN/mnt00001
/hana/data/HAN/mnt00001                    793971096  3059712
750579916   1% /hana/data/HAN/mnt00001
/dev/mapper/VG_HAN_LOG_1-LV_HAN_LOG_1      496233160  2461764
468564152   1% /hana/log/HAN/mnt00001
/hana/log/HAN/mnt00001                     496233160  2461764
468564152   1% /hana/log/HAN/mnt00001
```

Note that partition `mnt00002` is also configured, although not currently mounted by HANA. The `secfsd` status output should show the GuardPoint configuration as follows:

```
 GuardPoint                 Policy    Type     ConfigState  Status
Reason
----------                 ------    ----     -----------  ------
------
/hana/data/HAN/mnt00001  my-pol    manual  guarded      guarded      N/
A
/hana/log/HAN/mnt00001   my-pol    manual  guarded      guarded      N/
A
/hana/data/HAN/mnt00002  my-pol    manual  unguarded    not guarded
Inactive
/hana/log/HAN/mnt00002   my-pol    manual  unguarded    not guarded
Inactive
```

For more information, see SAP HANA Fiber Channel Storage Connector Admin Guide.

Using SAP HANA with LDT

SAP HANA is compatible with LDT with the following changes:

- You must add additional CTE commands to the HANA administrator entry. Using a text editor, edit `/etc/sudoers` and add entries for `/usr/bin/voradmin` and `/usr/bin/vmsec`:

  ```
   hanadm ALL=NOPASSWD: /usr/bin/secfsd,/usr/bin/voradmin,/usr/bin/
  vmsec,/sbin/multipath,/sbin/multipathd,/etc/init.d/multipathd,/
  usr/bin/sg_persist,/bin/mount,/bin/umount,/bin/kill,/usr/bin/
  lsof,/sbin/vgchange,/sbin/vgscan
  ```

- If you are using an `ext3` file system, you must mount it with extended attributes. Using a text editor, edit the storage section of the `global.ini` file, type:

  ```
  partition_*_data__mountOptions = -o user_xattr
  partition_*_log__mountOptions = -o user_xattr
  ```

Setting Memory Allocation

There is a limitation in memory allocations for SAP HANA with asynchronous direct I/O. When you use CTE in conjunction with applications like SAP HANA that can process large numbers of direct I/O writes through the Linux AIO interface, CTE can allocate more memory than is desirable.

To limit the amount of memory that CTE consumes for AIO buffers, use the following configuration to limit the amount of memory CTE consumed for AIO buffers:

```
max_aio_memory_limit <MB>
```

The MB value specifies how much memory to allocate to temporary DIO buffers.

Note

If you do not specify a value, the default is 0, which has no memory bounding effect.

Set the option by echoing a value into the `/opt/vormetric/DataSecurityExpert/agent//secfs/.sec/conf/` configuration file. For example:

```
echo 1024 > /opt/vormetric/DataSecurityExpert/agent/ secfs/.sec/conf/
max_aio_memory_limit
```

This limits the memory consumed by AIO buffers to 1GB.

Note

You **must restart** CTE after changing any values in the configuration directory to make the changes effective.

Using CTE with GlusterFS

This chapter describes how to configure CTE with GlusterFS. It contains the following sections:

- Overview
- Considerations and Requirements
- Configuring GlusterFS for CTE

Overview

GlusterFS is a scalable network file system suitable for data-intensive tasks such as cloud storage and media streaming. The Gluster share can also be mounted as an NFS client.

Thales provides support for GuardPoints in a Gluster environment.

Considerations and Requirements

- Gluster is supported on Red Hat 7 and Red Hat 8. Other versions of Red Hat are not supported.
- Clients must have gluster-fuse installed.
- Encryption keys for GlusterFS GuardPoints must use the CBC or CBC-CS1 encryption mode.
- The following CTE features are not supported in GlusterFS:
  - Auto-mounted file systems.
  - CTE-Live Data Transformation.
  - IDT-Capable GuardPoints.

Note

In a multi-node configuration, Thales has tested only the close-to-open consistency for GlusterFS. Other consistencies offered by GlusterFS are not guaranteed to work with CTE.

Configuring GlusterFS for CTE

1. On the Gluster server, configure the following Gluster volume properties for all configurations:

   - `performance.flush-behind` should be off.
   - `network.remote-dio` should be disabled.
   - `performance.strict-o-direct` should be on.

   For example, if the Gluster volume name is MyGlusterVolume, you would enter the following commands:

   ```
   gluster vol set MyGlusterVolume performance.flush-behind off
   gluster vol set MyGlusterVolume network.remote-dio disable
   gluster vol set MyGlusterVolume performance.strict-o-direct on
   ```

2. If you want to use CBC-CS1 encryption keys, configure the following additional Gluster volume properties on the Gluster server:

   - `performance.read-ahead` should be off.
   - `performance.cache-size` should be 1GB.
   - `performance.write-behind` should be off.
   - `performance.client-io-threads` should be on.

   For example:

   ```
   gluster vol set MyGlusterVolume performance.read-ahead off
   gluster vol set MyGlusterVolume performance.cache-size 1GB
   gluster vol set MyGlusterVolume performance.write-behind off
   gluster vol set MyGlusterVolume performance.client-io-threads on
   ```

3. On the clients, do the following based on your configuration:
   - If the kernel version is *earlier* than version 4.0.0, the Gluster share must be mounted with the `use-readdir=no` mount option.

     ```
     mount -t glusterfs -o acl,use-readdirp=no MyGlusterVolume /
     gluster
     ```

   - If you want to use CBC-CS1 encryption keys, the Gluster share must be mounted with the `direct-io-mode=enable` mount option.

     ```
     mount -t glusterfs -o acl,direct-io-mode=enable
     MyGlusterVolume /gluster
     ```

   - If you are using only CBC encryption keys with kernel versions 4.0.0 and later, no changes need to be made on the clients.

NetApp Snapshot Directory

This section describes SecFS support for NetApp .snapshot directory over NFS. It contains the following topics:

- Overview
- Accessing Snapshots
- Enabling Snapshots
- Dataxform Considerations

Overview

The NetApp snapshot directory contains ONTAP snapshot data entries for a specific live volume. Each snapshot is a read-only volume that is automatically mounted over NFS.

A snapshot copy is a read-only image of a traditional, or FlexVol volume, or an aggregate, that captures the state of the file system at a specific point in time.

Data ONTAP maintains a configurable snapshot copy schedule that creates and deletes snapshot copies automatically for each volume.

Accessing snapshots

By default, every volume contains a directory named .snapshot through which users can access previous versions of files. Users can gain access to snapshot copies depending on the file-sharing protocol used, NFS or CIFS. You can also prevent access to snapshot copies.

Snapshot files carry the same read permissions as the original file. A user who has permission to read a file in the volume, can also read that file in a snapshot copy. A user without read permission to the volume cannot read that file in a snapshot copy.

> **Note**
>
> Snapshot copies do not have write permissions.

Snapshot directories only display at the mount point, although they actually exists in every directory in the tree. This means that the .snapshot directory is accessible by name in each directory, but is only seen in the output of the `ls` command at the mount point. The snapshots are stamped with the date and time.

Enabling Snapshots

The NetApp storage administrator, or the OnTap device, must configure this feature. No configuration is required through CTE. CTE guards the client directory mounting the OnTap data volume over NFS.

> **Note**
>
> The NetApp documentation is located here: https://nt-ap.com/2vEnEeJ

Dataxform Considerations

You cannot transform snapshot directory entries with Dataxform with a new key, because the snapshots are read only. You must keep previous keys and alter the running security policy accordingly to maintain access to the older snapshot entries alongside any new snapshots taken with the new key.

Also, any snapshots that get created during the data transform process (this may take a long time) have to be discarded/deleted as it may contain a mix of data blocks encrypted with both old and new keys.

Best Practices

Maintaining keys for access to older snapshots can be tedious and cumbersome. Therefore, the simplest and safest practice is to delete all old snapshots once the data is transformed with a new key.

This allows for all new snapshots to be readable with the new key while old keys can be discarded, unless used in other security policies.

Using CTE with Quantum StorNext

This chapter describes how to configure CTE and Quantum StorNext devices to interoperate to allow CTE policies to apply to storage managed by Quantum StorNext.

This section contains the following topics:

- Overview of using CTE with Quantum StorNext
- CTE and Quantum StorNext compatibility
- Setting up CTE and Quantum StorNext integration
- Stop secfs before upgrading StorNext LAN clients

Overview of using CTE with Quantum StorNext

Quantum StorNext Fibre Channel-connected devices provide shared file access to third party storage for workstation clients and are optimized for simultaneous access to very large files such as video files. The Quantum StorNext file system is known as SNFS or by its older name, CVFS.

You can encrypt and control access to SNFS files with policies by installing CTE Agents on Linux clients that are configured for access to the SNFS file system. Some limitations apply to this integration, such as supported operating systems, supported SNFS features, concurrent read/write access by multiple clients, and GuardPoint settings (see the next section for more information about these limitations).

CTE and Quantum StorNext Compatibility

The following sections list the supported operating systems and CTE settings supported for use with Quantum StorNext file systems. Important unsupported configuration parameters are also listed.

Supported StorNext Server and Client Configurations

The CTE integration with SNFS file systems works only with certain SNFS versions, SNFS storage policies, and client operating systems.

Configuration parameter	Linux
StorNext (SNFS) operating system version	6.x
StorNext metadata controller (MDC) server OS type	Linux MDC supported
StorNext replication policy	Not supported
StorNext deduplication policy	Not supported
StorNext truncation	Supported
StorNext full and partial backup	Supported
StorNext expand file system	Supported
StorNext data migration	Supported
StorNext read-ahead cache	Disable for use with CTE
Client operating systems	Red Hat Enterprise Linux (RHEL) 7.x
StorNext LAN client	DLC
StorNext mount method: locally mounted directory	Supported
StorNext mount method: CIFS	Not supported
StorNext mount method: NFS	Not supported

Supported GuardPoint and Key Settings for SNFS File Systems

When configuring CTE GuardPoints and keys for SNFS, keep in the mind the compatibility limitations listed in the following table.

!! note

```
Because AES-CBC-CS1 keys are not supported on Windows, do not create a policy on Linux that uses AES-CBC-CS1 keys if access to the same SNFS GuardPoint is required by both Windows and Linux LAN clients.
```

Configuration element	Linux
Offline data transformation	Supported
Live Data Transformation (LDT)v	Not supported
Key manager compatibility	See the *Compatibility Matrix for CTE Agent with Data Security Manager or the Compatibility Matrix for CTE Agent with Data Security Manager* for your CTE version
Guard unstructured data	Supported
Guard structured data	Not supported
GuardPoint type: Directory (including entire SNFS volume)	Supported
GuardPoint type: Raw device	Not supported
GuardPoint type: Block device	Not supported
GuardPoint mount option: manual guard	Supported
GuardPoint mount option: auto guard	Supported
GuardPoint mount option: automount	Not supported
AES-CBC key type	Supported
AES-CBC-CS1 key type	Supported

Supported Concurrent Access Read/ Write Scenarios

If you want to allow access by multiple clients (users) to CTE-protected SNFS files under the same GuardPoint, just read-only access is supported. StorNext file locking is not implemented in CTE, so there is currently no way to prevent concurrent conflicting writes to the same file. As a result, Thales does not support write access to the same GuardPoint from multiple clients.

To enable read access to the same GuardPoint from multiple clients, ensure that all clients are configured to use the same policy and key.

Configuration parameter	Linux
Read/write access from a single LAN client to a GuardPoint	Supported
Read/write access from two or more LAN clients to the same GuardPoint	Not supported
Read-only access from one, two, or more LAN clients to the same GuardPoint	Supported

Setting up CTE and Quantum StorNext Integration

For the most part, CTE integration with Quantum StorNext is the same as for any standard file system. The next section provides an overview of the steps involved in making CTE work with SNFS. Later sections provide more information about the steps that are new or differ significantly from a typical CTE setup.

Integration Task Overview

The table below provides an overview of the steps involved in setting up SNFS and CTE to work together. As noted in the table, some of these tasks are described in the documentation for your selected key manager. Some of these steps may need to be performed by other staff members at your organization if you have divided the security administration duties as recommended by Thales and you don’t have access to the key manager.

Task	Key configuration notes	For more information
Install and configure a Quantum StorNext MDC server for use with CTE	Disable the StorNext read-ahead cache. Only certain StorNext policies, features, and mount types are supported. See Supported StorNext Server and Client Configurations.	See Installing and Configuring a Quantum StorNext MDC Server for Use with CTE.
Install and configure Quantum StorNext clients for use with CTE	Ensure that SNFS starts before secfs. See Ensuring that the StorNext SNFS File System Starts Before secfs. Only certain operating systems are supported. See Supported StorNext Server and Client Configurations.	See Installing and configuring Quantum StorNext DLC Clients for Use with CTE.
Create a domain for one or more SNFS hosts, or add them to an existing domain	No difference from standard CTE agent configuration.	See “Domain Management” in your key manager documentation.
Add the host to the key manager	No difference from standard CTE agent configuration.	See “Configuring Hosts and Host Groups” in your key manager documentation.
Install and register the CTE Agent on the host system	No difference in installation.	See Getting Started with CTE for Linux
Create encryption keys (optional)	No difference from standard CTE agent configuration.	See “Managing Keys” in your key manager documentation. For information about AES-CBC-CS1 keys, see Enhanced Encryption Mode.
Configure host groups containing one or more StorNext LAN clients (optional)	No difference from standard CTE agent configuration.	See “Configuring Hosts and Host Groups” in your key manager documentation.
Configure policies (including user, process, and resource	No difference from standard CTE agent configuration.	See “Configuring Policies” in your key manager documentation.

Task	Key configuration notes	For more information
sets) to control access or enable encryption		
Configure one or more GuardPoints	Some GuardPoint settings are not supported. See Supported GuardPoint and Key Settings for SNFS File Systems.	See “Managing GuardPoints” in your key manager documentation

Installing and Configuring a Quantum StorNext MDC Server for Use with CTE

Install and configure a Quantum StorNext metadata controller (MDC) server using the Quantum StorNext documentation as a guide. The CTE integration works with Linux StorNext MDCs. Ensure that you configure the StorNext server to work with the settings supported by CTE as listed in Supported StorNext Server and Client Configurations. For example, you must disable the StorNext read-ahead cache and only certain StorNext policies, features, and mount types are supported.

Installing and Configuring Quantum StorNext DLC Clients for Use with CTE

Install and configure Quantum StorNext DLC clients using the Quantum StorNext documentation as a guide. The CTE integration works with Linux StorNext DLCs.

Ensure that you configure DLC clients to work with the settings supported by CTE as listed in Supported StorNext Server and Client Configurations. For example, only certain operating systems are supported.

Note

Read-only access is supported if multiple StorNext LAN clients will access files in the same GuardPoint. For more information, see Supported Concurrent Access Read/Write Scenarios.

Ensuring that the StorNext SNFS File System Starts Before secfs

For CTE to function properly for Linux SNFS clients, the SNFS service must start before the CTE `secfs` service. Add an entry for the SNFS file system to `/etc/fstab` on each Linux client that has a CTE agent installed on it. Use the following format:

```
/snfs_share /stornext/snfs1 cvfs defaults,diskproxy=client 0 0
```

In this example, `/snfs_share` should be a share that has been exported from the StorNext Server. It should not be a local disk. You may have completed this configuration step as part of the StorNext LAN client installation. See the Quantum StorNext documentation for more details.

Installing the CTE Agent on Each StorNext LAN client

Install a CTE Agent on each computer that is set up as a StorNext LAN client and for which you want to set policies. For supported operating systems, see the table in Supported StorNext Server and Client Configurations.

Use any installation method supported for your operating system. For details, see Getting Started with CTE for Linux.

Stop secfs Before Upgrading StorNext LAN Clients

Before you upgrade StorNext Linux LAN clients, disable or stop the CTE `secfs` service. To stop the `secfs` service, follow these steps:

1. Log in as root on the computer that contains the LAN client that you want to upgrade.

2. Type the following command: `/etc/vormetric/secfs stop`

After you upgrade the StorNext client, start `secfs` again by logging in as root and running the `/etc/vormetric/secfs start` command.

Using CTE with McAfee Endpoint Security for Linux Threat Prevention

McAfee Endpoint Security for Linux Threat Prevention detects malware such as viruses and handles the malware according to polices that you configure in McAfee ePO. This chapter describes how to configure McAfee and CipherTrust Transparent Encryption to work together.

This section contains the following topics:

Supported McAfee Versions and Linux Operating Systems

In general, Thales has verified that CipherTrust Transparent Encryption is compatible with McAfee version 10.6.5 and later on Red Hat Enterprise Linux (RHEL) 7 and RHEL 8. See the most recent *Compatibility Matrix for CipherTrust Transparent Encryption Agent with CipherTrust Manager* or *Compatibility Matrix for CipherTrust Transparent Encryption Agent with Data Security Manager* for details about the versions of McAfee Endpoint Security for Linux Threat Prevention that have been verified to work with CipherTrust Transparent Encryption.

Ensuring the Correct McAfee Service Startup and Shutdown Order

CipherTrust Transparent Encryption services and McAfee services must be started and stopped in the correct order to prevent problems with any data that is guarded by

CipherTrust Transparent Encryption. This order is important any time these services need to be started or stopped, such as:

- During the normal startup and shutdown of your Linux host.
- Before enabling a scheduled upgrade of CipherTrust Transparent Encryption.
- Before performing a manual upgrade of CipherTrust Transparent Encryption.
- As needed for maintenance or troubleshooting.

Ensuring the Correct McAfee Service Order in systemd

Configuring the proper startup and shutdown order of CipherTrust Transparent Encryption and McAfee services in `systemd` ensures that the services start in the right order during system startup and shutdown. This is also important if you configure a scheduled upgrade of CipherTrust Transparent Encryption, as CipherTrust Transparent Encryption services will need to be shut down during the upgrade.

The following McAfee services must be configured to start after CipherTrust Transparent Encryption services:

- For McAfee 10.6.6 or later:
  - `mfetpd.service`
  - `mfeespd.service`
- For McAfee 10.6.5 or earlier:
  - `isectpd.service`
  - `isecespd.service`

To configure this behavior, add these services to the `Before=` line in the `secfs-fs-barrier.service` file on your system. The order of these services on the `Before=` line in the `secfs-fs-barrier.service` file does not matter. For more information, see [Location of Application Unit Configuration Files] and [Adding Applications to the secfs-fs-barrier.service File].

Starting or Stopping McAfee and CipherTrust Transparent Encryption Manually

When you manually start or stop McAfee and CipherTrust Transparent Encryption, you must do so in the correct order.

To manually stop McAfee and CipherTrust Transparent Encryption:

1. Stop McAfee services using one of the following:

   For McAfee 10.6.6 or later:

   ```
   systemctl stop mfetpd.service mfeespd.service
   ```

   For McAfee 10.6.5 or earlier:

   ```
   systemctl stop isecespd.service isectpd.service
   ```

2. Stop CipherTrust Transparent Encryption:

Linux distributions that support `systemd`	`/etc/vormetric/secfs stop`
Linux distributions that do not support `systemd`	`service secfs stop`

To manually start McAfee and CipherTrust Transparent Encryption:

1. Start CipherTrust Transparent Encryption:

Linux distributions that support `systemd`	`/etc/vormetric/secfs start`
Linux distributions that do not support `systemd`	`service secfs start`

2. Start McAfee services using one of the following:

   For McAfee 10.6.6 or later:

   ```
   systemctl start mfetpd.service mfeespd.service
   ```

   For McAfee 10.6.5 or earlier:

   ```
   systemctl start isecespd.service isectpd.service
   ```

Excluding CTE protected directories with McAfee

Starting with CipherTrust Transparent Encryption v7.2.0, two conflicts exist between McAfee’s On-Access Scan and CipherTrust Transparent Encryption when both applications are installed, regardless of installation order. This conflict occurs because McAfee's On-Access Scan tries to scan files in the initial startup (protected) directory, `/opt/vormetric/DataSecurityExpert/agent/secfs/.sec/` whenever those files are accessed by CipherTrust Transparent Encryption's own processes and utilities. Access to a CipherTrust Transparent Encryption protected directory, and its subdirectories, is restricted so that no other process or utility can access them.

The fix is to add the CipherTrust Transparent Encryption protected directories to the McAfee On-Access Scan exclusion list:

1. Change the directory. At the command line, type:
   - For **McAfee 10.6.5** or prior versions:

     `#cd /opt/isec/ens/threatprevention/bin/`
   - For **McAfee 10.6.6** or subsequent versions:

     `#cd /opt/McAfee/ens/tp/bin`

2. Add the exclusion to McAfee's On-Access Scan, type:

   ```
   '--addexclusionrw --excludepaths "/opt/vormetric/
   DataSecurityExpert/agent/secfs/.sec/" --excludesubfolder'
   ```

The entire exclusion list command should look like the following:

- **McAfee 10.6.5** or prior versions:

  ```
  ./isecav --setoasprofileconfig --profile standard --addexclusionrw
  --excludepaths "/opt/vormetric/DataSecurityExpert/agent/
  secfs/.sec/" –excludesubfolder
  ```

- **McAfee 10.6.6** or subsequent versions:

  ```
  ./mfetpcli --setoasprofileconfig --profile standard --
  addexclusionrw --excludepaths "/opt/vormetric/DataSecurityExpert/
  agent/secfs/.sec/" –excludesubfolder
  ```

Note

You only need to run this command once.

Updating McAfee

It is not necessary to shut down CipherTrust Transparent Encryption services when you update McAfee Endpoint Security to a new version. Follow the update procedure described by McAfee. Before updating, ensure that the new version of McAfee is compatible with CipherTrust Transparent Encryption as described in [Supported McAfee Versions and Linux Operating Systems].

Note

When you upgrade McAfee, make sure that the current McAfee services are configured to start *after* the CipherTrust Transparent Encryption services in `systemd`. The McAfee service names depend on the version of McAfee that you are using. For details, see [Ensuring the Correct McAfee Service Order in systemd].

Virus Scanning Behavior Differences for CIFS and NFS GuardPoints

By default, on McAfee Endpoint Security, on-access virus scanning for remotely mounted file systems such as CIFS and NFS, is disabled. However, for GuardPoints configured on CIFS and NFS volumes, this default is ignored. So on-access virus scanning is always on for GuardPoints configured on CIFS and NFS volumes. This means that if a process attempts to save an infeCipherTrust Transparent Encryptiond file to a GuardPoint configured on a CIFS or NFS volume, the infeCipherTrust Transparent Encryptiond file will be discovered immediately, if it matches the McAfee malware detection algorithm and handled according to the appropriate malware policy in McAfee ePO.

Using CTE with Trend Micro Deep Security Software

Trend Micro's Deep Security software provides comprehensive security in a single solution that is purpose-built for virtual, cloud, and container environments. Thales has verified certain versions of this Trend Deep product for compatibility with CTE on Red Hat Enterprise Linux (RHEL) 7 and RHEL 8.

This section contains the following topics:

- Ensuring Correct Deep Security Service Startup Order
- Updating Deep Security

Ensuring Correct Deep Security Service Startup Order

CTE services and Deep Security services must be started and stopped in the correct order to prevent problems with your data that is guarded by CTE. This order is important any time these services need to be started or stopped, such as:

- During normal startup and shutdown of your Linux host.
- Before enabling a scheduled upgrade of CTE.
- Before performing a manual upgrade of CTE.
- As needed for maintenance or troubleshooting.

Ensuring Correct Deep Security Service Startup Order in systemd

Configuring the proper startup and shutdown order of CTE and Trend Micro's Deep Security services in systemd ensures that the services start in the right order during system startup and shutdown. This is also important if you configure a scheduled upgrade of CTE.

The following Deep Security service must be configured to start after CTE services:

`ds_agent.service`

To configure this behavior, add this service to the `Before=` line in the `secfs-fs-barrier.service` file on your system. The order of these services on the `Before=` line in the `secfs-fs-barrier.service` file does not matter. See Location of Application Unit Configuration Files for the location of the `secfs-fs-barrier.service` file on your system. See Adding Applications to the secfs-fs-barrier.service File for information about how to add services to the `secfs-fs-barrier.service` file.

Ensuring Correct Deep Security Service Startup Order Manually

Perform the following commands in this order if you need to stop Deep Security and CTE services manually:

1. Stop Deep Security services:

```
systemctl stop ds_agent.service
```

2. Stop CTE:

Linux distributions that support systemd	`/etc/vormetric/secfs stop`
Linux distributions that do not support systemd	`service secfs stop`

Perform the following commands in this order if you need to start Deep Security and CTE services manually:

1. Start CTE:

Linux distributions that support systemd	`/etc/vormetric/secfs stop`
Linux distributions that do not support systemd	`service secfs stop`

2. Start Deep Security services:

```
systemctl start ds_agent.service
```

Updating Deep Security

It is not necessary to shut down CTE services when you update Trend Micro Deep Security to a new version. Follow the Update Deep Security software described by Trend Micro.

CTE for Cloud Object Storage

This section discusses how to configure CTE for Cloud Object Storage buckets. It contains the following topics:

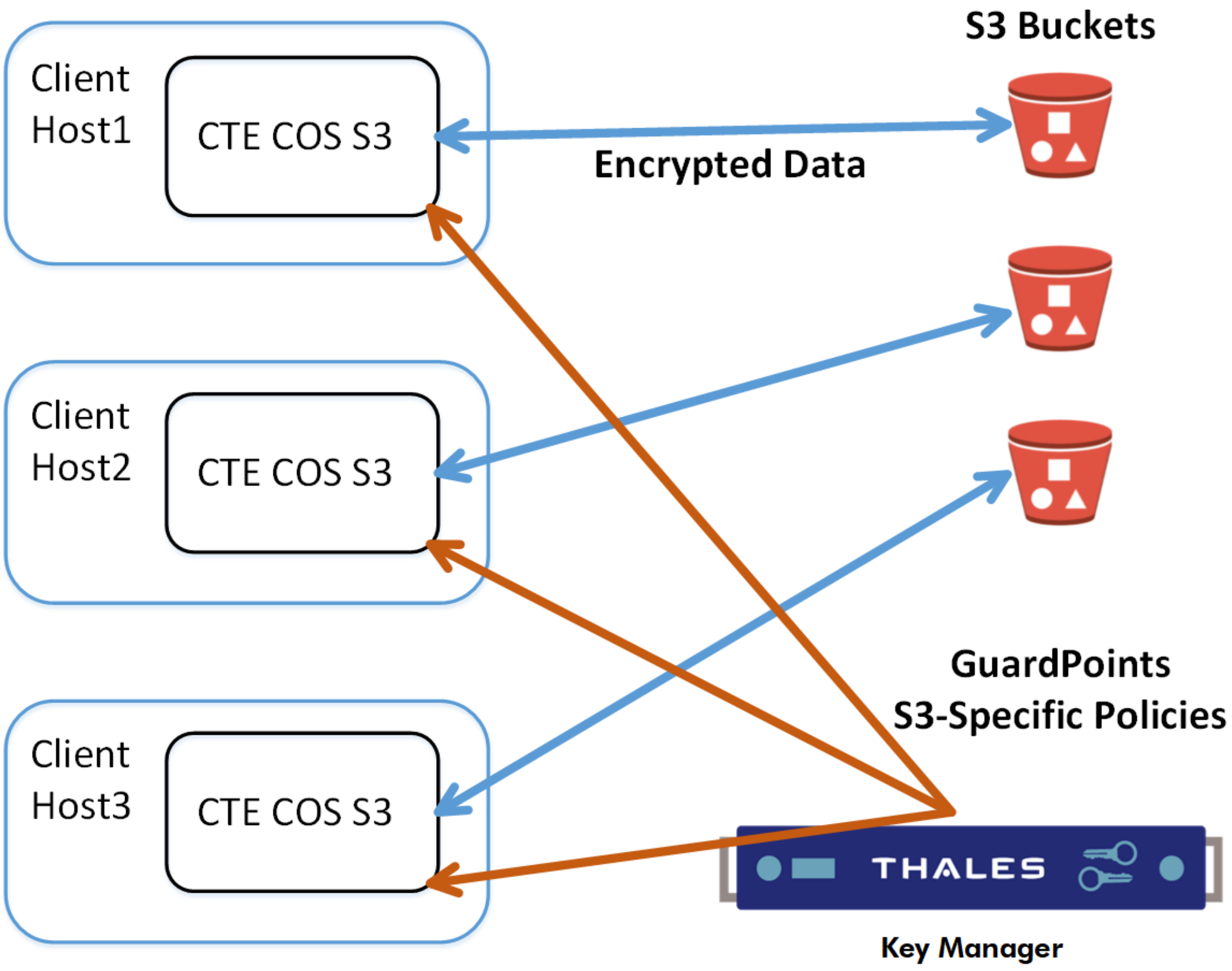
Overview

CipherTrust Transparent Encryption for Cloud Object Storage (CTE COS) is an object storage service that you can use to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. CTE COS provides management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements.

CTE COS for Amazon Simple Storage Service (CTE COS S3) is an extension to CTE for inline encryption and local host access controls involving applications performing REST-API-based operations to S3 object stores.

CTE COS S3 is not a replacement for AWS IAM access controls which are enforced independently at the AWS server end.

The following diagram shows the high-level CTE COS S3 architecture:



Supported operations

- CTE COS S3 will process only AWS S3 REST https calls issued by applications. The interception is done by a bundled TLS proxy with additional CTE services.
- CTE's traditional access controls and inline encryption are transparently applied during the following operations:
  - Create a bucket
  - Write an object
  - Read an object
  - Delete an object
  - List objects

Limitations

- CTE COS S3 is supported only on RHEL 8 and RHEL 9.
- Only REST API based S3 protocol-aware applications are supported.
- Only locally generated self-signed COS Proxy CA Certificates are supported.
- AWS S3 URL path validations are not currently implemented. CTE COS S3 requires that the user must specify the correct URLs for bucket paths.
- CTE COS S3 permits only AES CBC-CS1 encryption. Encrypted files in protected buckets therefore will prepend the 4K embedded header used in CTE's AES CBC-CS1 encryption. The key manager will enforce usage of AES CBC-CS1 keys within S3 bucket policies.

Multi-part Upload Restrictions

- All upload operations for a multi-part upload must be conducted from the same host.
- Maximum file size upload is 5TB.
- Part sizes during uploads must be identical. The part numbering must be in sequence starting with 1, e.g. 1,2,3,4,… Not 10, 20, 30, etc.
- The part size specified in CTE S3’s AWS credential file must match the part size specified within the application.
- Thales recommends that the Content-MD5 header be included in the request message.

System and Software Requirements

- The RHEL 8, or 9 Linux hosts must meet the standard CTE requirements.
- You must have an AWS account
- We recommend that you guard the bucket by restricting it with a CTE COS S3 Role to prevent accidental data corruption from connections outside the control of CTE COS S3. For details, see Optionally Configure a CTE COS S3 Role for Guarded Buckets.
- You must download and install the pre-requisite rpm packages before you install CTE COS S3. For details, see Install Required Linux Packages.
- The CTE Agent must be installed in the default directory on the host. You cannot change the default path if you also enable CTE COS S3.

Client Software Requirements

- Clients must exist on the same host as CTE COS S3. External Client connections will be rejected by the COS Proxy.
- Clients must be AWS S3 protocol aware, and can directly connect to a S3 bucket without the use of an intermediary service, such as the AWS S3 Management Console website.
- Clients must be configured to divert outgoing network connection to a COS proxy (default is localhost:3128 or 127.0.0.1:3128). For more information, see Configure the AWS CLI Network Proxy.
- Clients must use the TLS 1.2 / TLS 1.3 network encryption protocol over TCP/IP to establish connections to the COS proxy.
- Clients must be configured to use the COS CA Root Certificate of the COS proxy to verify and authenticate TLS connections. For more information, see Configure the AWS CLI to use the COS Root CA Certificate.

CTE COS S3 Installation Overview

In order to configure CTE COS S3, you must complete the following tasks.

> **Note**
>
> The AWS CLI is used as a sample application to explain the detailed installation procedure.

1. Install the required packages for CTE COS S3. For details, see Install Required Linux Packages.
2. Install CTE and generate the local COS Proxy CA Certificate. For details, see Install CTE with COS Service.

Note

You can only install the Cloud Object Storage feature during CTE installation. You cannot install it post installation.

1. Configure the client to use the COS Proxy CA Certificate. For details, see Configure the AWS CLI to use the COS Root CA Certificate.
2. Configure the client to use the COS proxy port. For details, see Configure the AWS CLI Network Proxy.
3. Configure CTE COS S3 with AWS Credentials. For details, see Configure CTE COS S3.
4. Optionally configure Role & IAM policies for CTE COS S3 for guarded buckets. For details, see Optionally Configure a CTE COS S3 Role for Guarded Buckets.
5. Configure the guarded buckets. For details, see Guard an AWS Bucket.

Install Required Linux Packages

CTE COS S3 requires the following prerequisite packages:

- `boost-regex`
- `boost-system`
- `boost-thread`
- `lib-curl`
- `libtool-ltdl`
- `libxml2`
- `epel-release`.
- `cryptopp` // This package must be installed after `epel-release`.
- `log4cpp` // This package must be installed after `epel-release`.

For example:

```
sudo yum install boost-regex boost-system boost-thread libcurl libtool-
ltdl libxml2 epel-release
sudo yum install cryptopp log4cpp
```

CTE COS S3 supports both Python2 and Python3. If both versions of Python are available, CTE COS S3 will use Python3. For either Python package, you also need to install the Python modules "`boto3`" and "`future`" after you install the main python package.

- Example for Python2:

```
 sudo yum install python-pip
sudo pip install boto3 future
```

- Example for Python3:

```
 sudo yum install python3 python3-pip
sudo pip3 install boto3 future
```

Install CTE with COS Service

Answer the following prompts as described in this section.

1. When the installer asks about Cloud Object Storage, type **Y** and follow the prompts as shown.

   > **Note**
   >
   > You can only install the Cloud Object Storage feature during CTE installation. You cannot install it post installation.

   Do *not* change the default CTE installation directory. CTE must be installed in the default location if you enable Cloud Object Storage.

   Do you want this host to have Cloud support enabled on the server? (Y/N) [N]: Y

```
 CTE COS CA Cert is located in /opt/vormetric/DataSecurityExpert/a
gent/squid/etc/cosCA.crt
 Clients must be updated to use the new CA Certificate

 Generating certificate signing request for the kernel component..
.done.
 Signing certificate...done.
 Generating EC certificate signing request for the vmd...done.Sig
```

```
ning certificate...done.
 Generating EC certificate signing request for the vmd...done.
 Signing certificate...done.

 The following is the fingerprint of the EC CA certificate.
Please verify that it matches the fingerprint shown on the Dashbo
ard page of the Management Console.
 If they do not match, it can indicate an unsuccessful setup or
an attack.
 B0:93:C7:67:07:C9:CB:09:E2:21:F1:5C:8A:C8:79:8F:03:86:21:F2
 Do the fingerprints match? (Y/N) [N]: Y

 Successfully registered the CipherTrust Encryption Expert File S
ystem Agent with the primary CipherTrust Data Security Server on s
ecurity.manager.example.com.
 Starting CTE Cloud Service
 Installation success.
```

Configure the AWS CLI to use the COS Root CA Certificate

You must configure the AWS CLI to use the COS root CA certificate. To do so, edit `~/.aws/config` and add the following line to the AWS cli configuration file:

```
ca_bundle = /opt/vormetric/DataSecurityExpert/agent/squid/etc/cosCA.crt
```

For example:

```
 cat ~/.aws/config
[default]
output = json
Region = us-west-1
ca_bundle = /opt/vormetric/DataSecurityExpert/agent/squid/etc/cosCA.crt
```

Configure the AWS CLI Network Proxy

All communications between client applications and the AWS server must be done through the COS proxy and the environment variable `HTTPS_PROXY` or `https_proxy` should be set. If both variables are defined, then the AWS CLI will use `https_proxy`.

For example:

```
Export HTTPS_PROXY=localhost:3128
```

Configure CTE COS S3

In order for CTE COS S3 to do transparent encryption and decryption, all requests sent to the AWS S3 server must be generated and signed using valid AWS credentials. In order to retrieve these credentials for CTE COS S3, you can use any of the following methods:

- **The User Supplies the Credentials for the User's AWS Account**

  The application can send the long-term credentials from the user's AWS account to CTE COS S3. These credentials do not expire.

  For the AWS CLI, the credentials are in the credential file `~/.aws/credential`. The user credentials consists of an Access Key ID and a Secret Access Key. For details about accessing the user's credentials, see your AWS documentation.

  To add the AWS credentials to CTE COS S3, use the `voradmin cos s3 cred add` command:

  ```
  voradmin cos s3 cred add [<aws_key_id> <aws_secret_key>]
  ```

  where:

  - `<aws_key_id>` is the AWS secret key ID from the `.aws/credentials` file.
  - `<aws_secret_key>` is the AWS secret key from the same file.

  For example:

  ```
  voradmin cos s3 cred add AKIA****P KQSm****D
  ```

- **The User Supplies Temporary Security Credentials**

You can use temporary security credentials, which expire after a short period of time. Usually temporary security credential are obtained through IAM roles and other features of the AWS Security Token Service.

Use the `voradmin cos s3 cred add` command, described above, to add the temporary credential to CTE COS S3.

- **CTE COS S3 Captures Temporary Credentials**

  CTE COS S3 can capture a temporary, newly generated, security credential and automatically add it to CTE COS S3 if the application generates the temporary security credential using the AWS Security Token Service with one of the following 3 actions and `HTTPS_PROXY` is set to `localhost:3128`.

  - `AssumeRole`
  - `AssumeRoleWithSAML`
  - `AssumeRoleWithWebIdentity`

  No other action is required from the application or user.

- **CTE COS S3 Retrieves Credentials from EC2 Instance Metadata Service**

  When CTE COS S3 is installed on AWS EC2 instance with an attached role, CTE COS S3 automatically retrieves the credential from Instance Metadata Service and uses it. However, if CTE COS S3 already has a valid credential given by the user or admin using the `voradmin` command, then than that credential will be used instead. For information about setting up an IAM Role, see your AWS EC2 documentation.

  No action is required from application or user.

Setting the Default Chunk Size

Note

If a chunk size is configured in the AWS CLI configuration, you must configure the same chunk size for CTE COS S3.

The default chunk size for multi-part uploads is 8 MB. To change the chunk size, use the following command:

```
voradmin cos s3 chunk [<aws_key_id> <aws_secret_key>] [<chunk_size>]
```

where:

- `<aws_key_id>` is the AWS secret key ID from the `.aws/credentials` file.
- `<aws_secret_key>` is the AWS secret key from the same file.
- `<chunk_size>` is the number of MB per chunk that you want to use for multi-part uploads. Enter an integer between 5 and 5120.

The `voradmin` command prompts for any of the optional parameters that you do not specify on the command.

For example, to set a chunk size of 250MB, you would enter:

```
voradmin cos s3 chunk AKIA****P KQSm****D 250
```

Optionally Support other 3rd Party S3 Compatible Storage

CTE COS S3 has a configuration file that lists all supported S3 compatible cloud storages. It is located at:

```
`<installation directory>/agent/squid/etc/cos.conf`
```

To support a new 3rd party S3 compatible storage:

1. Add a new section to the configuration file with its name and endpoint base url. The following is an example with Wasabi Cloud Storage.

   ```
   [wasabi]
   endpoint = wasabisys.com
   ```

2. Restart CTE COS service using the following command:

   ```
   # voradmin cos service stop
   # voradmin cos service start
   ```

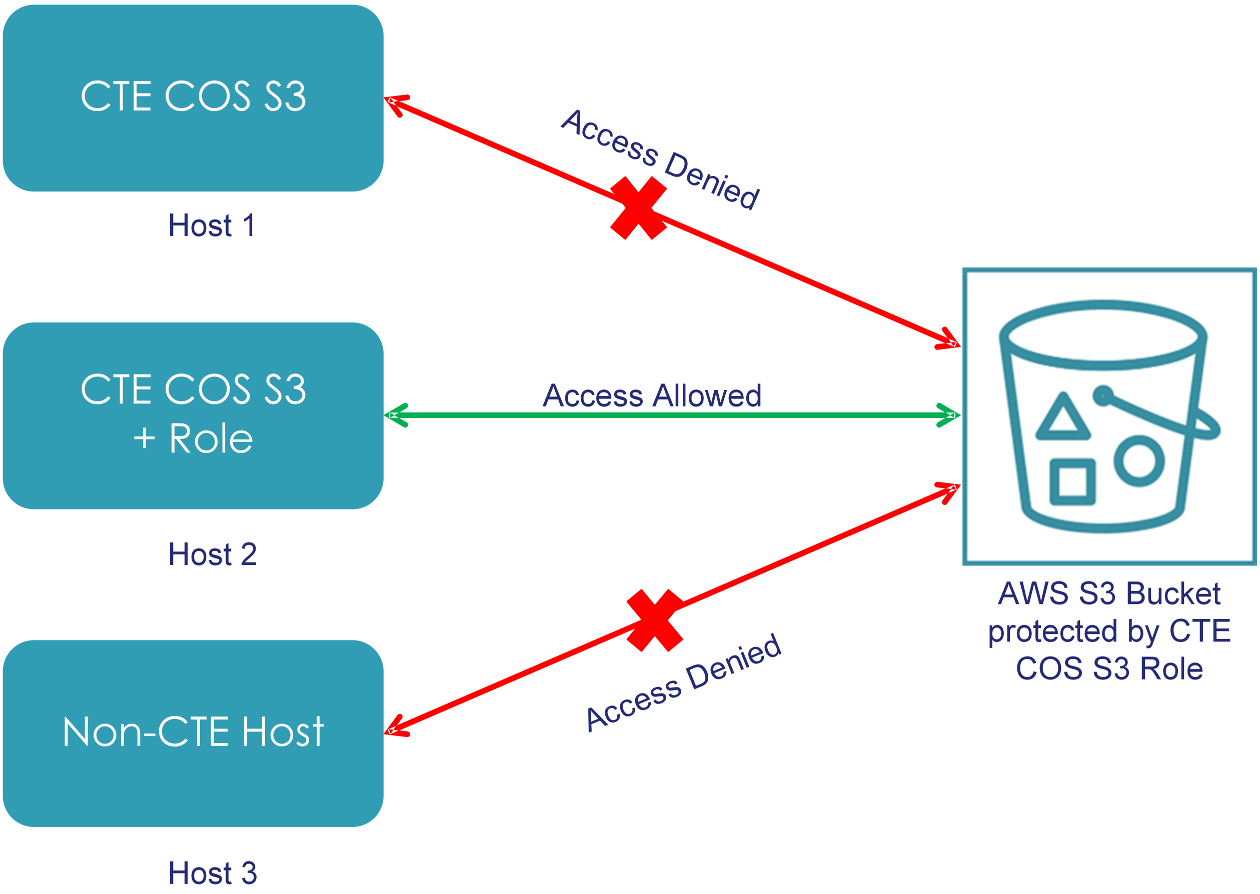
3. Configure credentials and other needed items following the guideline for the 3rd party S3 compatible storage.

Configure a CTE COS S3 Role for Guarded Buckets

AWS provides the IAM Role feature that contains certain specific permissions. A user can assume the IAM Role and therefore take on those permissions. CTE COS S3 provides a special feature using this IAM Role to prevent access to one or more buckets outside the CTE COS S3 protection as shown in the following diagram:

Note

This step is optional.



Only hosts that are configured with the CTE COS S3 Role can access the protected buckets. All other access attempts from any hosts where the CTE COS S3 Role is not configured, including attempts made through the AWS S3 Console, are blocked for the protected buckets.

This is a one-time configuration process. After the CTE COS S3 Role is configured on a host by the system or security administrator, all authorized users in the host can access the protected buckets.

Prerequisites

To set up a CTE COS S3 Role, you need a delegated IAM user, role, and policy. The delegated IAM user should be created by the AWS Administrator without any specific privileges. The role and policy must be created by a user who has at least the following privileges:

- `iam:ListPolicies`
- `iam:CreatePolicy`
- `iam:GetPolicyVersion`
- `iam:ListRoles`
- `iam:ListRolePolicies`
- `iam:ListAttachedRolePolicies`
- `iam:AttachRolePolicy`
- `iam:UpdateAssumeRolePolicy`
- `iam:CreateRole`
- `iam:GetRole`

Procedure

1. In the IAM Management Console, create a policy that allows access to specific S3 resources. You can leave the policy open to include all S3 resources in the account or include only those buckets that require CTE protection. Make sure you name the policy something that you will remember.

   > **Tip**
   >
   > You can also create the policy as an inline policy after you create the CTE COS S3 Role later in this procedure.

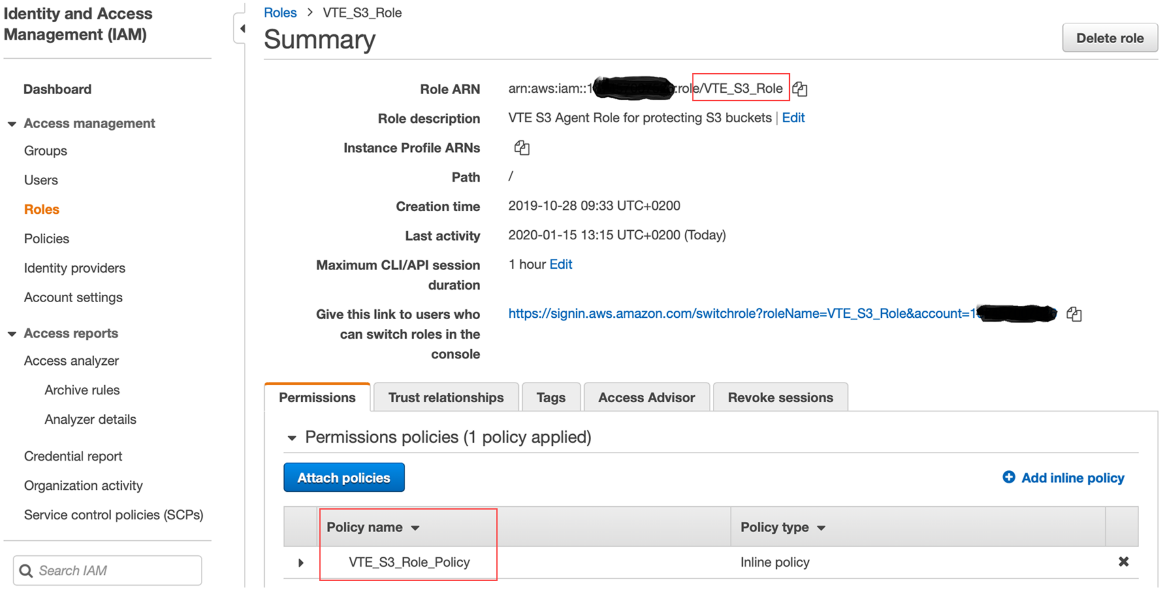
   For example, you can create a policy called VTE_S3_Role_Policy that allows access to the single S3 bucket `vte-cos-s3-rtb`. To verify that the policy restricts access to that bucket, you can look at the Resource allocation in the Policy summary.

The full JSON for the the VTE_S3_Role_Policy is:

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
        "Sid": "VisualEditor1",
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": [
           "arn:aws:s3:::vte-cos-s3-rtb",
           "arn:aws:s3:::vte-cos-s3-rtb/*"
           ]
      }
   ]
}
```

2. Create a new role that you can use for the CTE COS S3 Role. For example, you could call the role VTE_S3_Role.

3. Assign the CTE COS S3 policy you created to the role, or click **Add inline policy** to create a new policy. For example:



Summary

Role ARN: arn:aws:iam::[redacted]role/VTE_S3_Role

Role description: VTE S3 Agent Role for protecting S3 buckets | Edit

Instance Profile ARNs:

Path: /

Creation time: 2019-10-28 09:33 UTC+0200

Last activity: 2020-01-15 13:15 UTC+0200 (Today)

Maximum CLI/API session duration: 1 hour Edit

Give this link to users who can switch roles in the console: https://signin.aws.amazon.com/switchrole?roleName=VTE_S3_Role&account=[redacted]

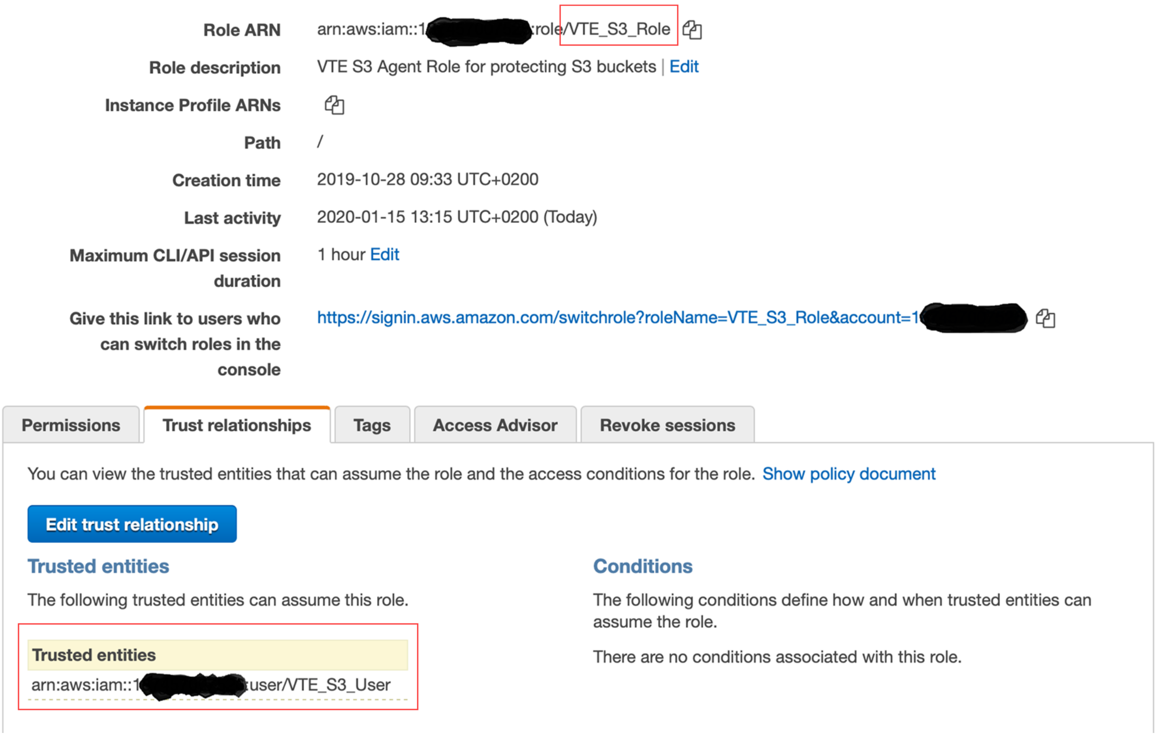
Permissions policies (1 policy applied)

Policy name	Policy type
VTE_S3_Role_Policy	Inline policy

4. Create a delegated IAM user for the CTE COS S3 Role. The user does not require any privileges because its only job is to assume the CTE COS S3 Role.

The delegated IAM user can either be in the same account as the role or it can be in a different trusted account. For example, you could create a user called VTE_S3_User with no privileges.

5. Set "Trusted Entities" to the delegated IAM user on the **Trust relationships** tab. For example:



Role ARN	arn:aws:iam::1[redacted]role/VTE_S3_Role
Role description	VTE S3 Agent Role for protecting S3 buckets | Edit
Instance Profile ARNs	
Path	/
Creation time	2019-10-28 09:33 UTC+0200
Last activity	2020-01-15 13:15 UTC+0200 (Today)
Maximum CLI/API session duration	1 hour Edit
Give this link to users who can switch roles in the console	https://signin.aws.amazon.com/switchrole?roleName=VTE_S3_Role&account=1[redacted]

You can view the trusted entities that can assume the role and the access conditions for the role. Show policy document

Trusted entities

The following trusted entities can assume this role.

Trusted entities
arn:aws:iam::1[redacted]user/VTE_S3_User

Conditions

The following conditions define how and when trusted entities can assume the role.

There are no conditions associated with this role.

6. Configure the CTE COS S3 Role with the credentials of the delegated IAM user you created earlier by entering the following command in the command line on the host system:

```
voradmin cos s3 role config [<aws_key_id> <aws_secret_key>
<user_arn> <role_arn>]
```

where:

- `<aws_key_id>` is the AWS secret key ID for the delegated IAM user that you created.
- `<aws_secret_key>` is the AWS secret key or the delegated IAM user that you created.
- `<user_arn>` is the Amazon Resource Name for the delegated IAM user that you created.
- `<role_arn>` is the Amazon Resource Name for the delegated IAM Role that you created.

If you omit any of the optional parameters, the `voradmin` command prompts you for that information.

For example, if the AWS account number for the delegated IAM user is 1XXXXXXXXXXX, the user name is VTE_S3_User, and the CTE COS S3 Role is VTE_S3_Role, you would enter:

```
voradmin cos s3 role config AKIA****P KQSm****D arn:aws:iam::
1XXXXXXXXXXX:user/VTE_S3_User arn:aws:iam::1XXXXXXXXXXX:role/
VTE_S3_Role
```

After you configure the user and assign the CTE COS S3 Role, CTE will access the S3 bucket through the delegated IAM user account using temporary credentials that CTE regenerates periodically. These credentials are maintained entirely by CTE and are never exposed to end users.

Secure an S3 Bucket with the CTE COS S3 Role

When you enable the CTE COS S3 Role for a bucket, the associated bucket policy prevents unauthorized users from accessing the contents of the bucket. To enable the CTE COS S3 Role for a bucket, use the following command: :::yaml voradmin cos s3 role secure-bucket

where:

- `<aws_key_id>` is the AWS secret key ID for the delegated IAM user that you created.
- `<aws_secret_key>` is the AWS secret key for the delegated IAM user that you created.
- `<cos name>` is the cloud service name from CTS COS configuration file, `<installation directory>/agent/squid/etc/cos.conf`.
- `<region>` is the region where the S3 bucket is located in the targeted cloud storage service.
- `<bucket_name>` is the name of the S3 bucket on which you want to enable the CTE COS S3 Role.

For example, if the bucket name is vte-cos-s3-rtb, you would enter:

```
voradmin cos s3 role secure-bucket AKIA****P KQSm****D aws us-west-1
vte-cos-s3-rtb
```

Disable the CTE COS S3 Role for an S3 Bucket

To remove the CTE COS S3 Role restrictions associated with a bucket, use the following command:

```
voradmin cos s3 role release-bucket <key_id> <secret key> <cos name>
<region> <bucket_name>
```

where:

- `<aws_key_id>` is the AWS secret key ID for the delegated IAM user that you created.
- `<aws_secret_key>` is the AWS secret key for the delegated IAM user that you created.
- `<cos name>` is the cloud service name from CTS COS configuration file, `<installation directory>/agent/squid/etc/cos.conf`.
- `<region>` is the region where the S3 bucket is located in the targeted cloud storage service.
- `<bucket_name>` is the name of the S3 bucket on which you want to disable the CTE COS S3 Role.

For example, if the bucket name is vte-cos-s3-rtb, type:

```
voradmin cos s3 role release-bucket AKIA****P KQSm****D aws us-west-1
vte-cos-s3-rtb
```

Guard an AWS Bucket

To Guard an AWS bucket, you must:

1. Create a CBC_CS1 key. For CM, see Creating a New Keyin the CM Administrator Guide.
2. Create a Cloud Object Storage (COS) policy. See Creating a COS Policy.
3. Apply the policy to the host. For CM, see Managing GuardPointsin the CTE Administrator Guide.

Additional COS Proxy Root CA Certificate Information

The CTE COS CA Certificate, not to be confused with the Kernel and VMD Kernel Certificates, is used with the COS internal Proxy Certificate Authority and must be used by Clients to validate Certificates received during their TLS connection handshake. The default COS CA Self-Signed root CA is automatically created using a locally generated Public/Private Key with the following parameters:

```
CERT_FIELD_PARAM="/C=OZ/ST=Munchkin-land/L=Emerald City/O=ACME Inc/
OU=ACME Deliveries/CN=localhost"

SUBJECT_ALT_NAME_PARAM="DNS.1:localhost,IP.1:127.0.0.1"
```

To view the currently installed Certificate for the COS Proxy CA, use the `voradmin cos ca_cert display` command.

In the context of the internal COS Proxy CA, the FQDN of 'localhost' would be the correct value, as well as the loop-back IP address of 127.0.0.1 This results in the following locally generated Root CA Certificate.

```
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
2a:28:2c:c5:d6:3b:05:11:fe:6e:32:1d:aa:35:29:44:e5:0d:ce:bf
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=OZ, ST=Munchkin-land, L=Emerald City, O=ACME Inc, OU=ACME
Deliveries, CN=localhost
Validity
Not Before: Feb 11 18:19:33 2020 GMT
Not After : Feb 10 18:19:33 2021 GMT
Subject: C=OZ, ST=Munchkin-land, L=Emerald City, O=ACME Inc, OU=ACME
Deliveries, CN=localhost
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
```

```
00:b9:e6:60:c9:00:f8:00:83:b7:1b:ff:b2:31:eb:
66:5a:eb:21:87:1c:aa:3d:71:b8:08:42:4d:82:6c:
9a:5c:c7:d0:ad:ec:11:9b:be:80:15:55:ab:bc:38:
11:9c:80:c4:1e:63:31:ae:b7:33:8f:88:0b:c2:ca:
e9:e8:0d:78:5a:19:e3:d9:45:fd:4c:b4:81:24:ea:
d3:d4:b9:d2:14:07:e0:33:df:b9:75:36:57:16:4d:
6e:ee:bf:5f:1d:13:14:10:d1:ba:29:0e:1e:11:38:
84:78:8a:e8:ed:1a:24:f7:6a:ac:87:66:9b:21:23:
7b:2c:44:b3:33:6c:04:b7:aa:8c:d3:64:d2:5e:b6:
56:b5:46:54:a9:37:06:c8:e5:30:5f:2a:ba:78:00:
4a:2f:f1:66:a0:1f:fd:26:05:8d:e0:da:23:1e:1b:
1e:a8:ee:77:73:76:32:3c:5e:01:aa:0f:d5:8b:ac:
a9:08:7e:50:63:5e:88:95:e5:5f:dc:1d:7b:b0:59:
50:c1:56:ba:e6:11:da:c6:c5:79:3e:a6:46:f2:39:
db:6a:9d:aa:da:ff:68:d0:39:9c:fd:5a:d5:0e:3e:
41:07:62:32:c0:be:4f:92:56:34:92:c8:1d:bd:87:
ec:e5:3b:44:a0:8f:8c:09:f9:37:40:df:b3:24:bb:
8d:67
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints: critical
CA:TRUE, pathlen:0
X509v3 Key Usage:
Certificate Sign, CRL Sign
X509v3 Subject Key Identifier:
C5:19:E5:41:B7:69:E7:10:27:2D:F6:49:0D:46:0A:4B:FE:8C:7E:CB
X509v3 Authority Key Identifier:
keyid:C5:19:E5:41:B7:69:E7:10:27:2D:F6:49:0D:46:0A:4B:FE:8C:7E:CB
X509v3 Subject Alternative Name:
DNS:localhost, IP Address:127.0.0.1
X509v3 Issuer Alternative Name:
DNS:localhost, IP Address:127.0.0.1
Signature Algorithm: sha256WithRSAEncryption
2d:3c:2b:93:c0:61:1d:35:d7:f2:5f:5c:e8:0d:61:57:f2:a8:
e0:ec:98:74:02:b5:c4:78:a4:2f:b5:2b:b4:96:56:17:93:89:
eb:45:ac:df:1e:1b:e0:d5:38:da:55:62:61:97:5b:d9:9e:31:
9b:71:f1:17:37:31:5d:12:0f:5e:c1:ea:29:ee:b2:97:6e:7c:
c0:97:a9:8d:a9:2c:c0:68:e4:fa:b1:21:f8:50:b8:c0:2e:51:
```

```
fd:f2:5b:4d:41:72:0c:48:a2:db:47:14:66:20:c7:62:bd:33:
e8:a4:f4:22:c9:07:0f:0d:58:a0:9e:a1:f9:96:9c:97:c1:28:
6a:18:6f:ea:b9:28:42:48:5a:5c:da:98:22:9f:05:59:27:82:
3f:3d:4e:0b:9d:37:04:76:0e:ec:d9:f1:25:c8:78:78:fc:31:
d0:cb:24:db:47:96:7c:fa:dc:0d:14:6c:13:44:8d:87:5b:82:
d2:0f:a9:8c:48:bd:a6:b1:b9:0c:bb:50:14:70:d0:8b:7b:8c:
a5:e5:52:83:47:25:15:d6:d0:17:e0:9f:f7:99:d0:2e:17:93:
c5:38:e0:b8:c8:d4:f2:ed:39:99:ec:19:cf:5e:39:78:7b:5f:
07:48:4b:df:ec:d9:94:c5:aa:df:4d:a9:a5:a9:e3:88:74:0e:
d7:74:83:87
```

If you want to change the defaults, you can use the silent install option with the `CERT_FIELD_PARAM` and `SUBJECT_ALT_NAME_PARAM` set to the desired values, or you can replace the default Certificate using the `voradmin cos ca_cert` command. For more details, see the voradmin manpage.

Protecting Python Programs with CTE

CTE uses SHA-256 to generate binary signatures that can be used to allow certain trust levels for a customer application. Capturing the signature of a binary allows CTE to determine if the application's binary has changed in any way, because those changes might mean the security of the binary has been compromised.

While this mechanism works well for executable binaries to verify an application's validity at run time, this mechanism does *not* work for applications implemented with interpreter languages because CTE can only access the interpreter's signature to validate the file. The program itself is something that is executed by the interpreter and thus CTE cannot hook into this process. Scripting languages like Python are particularly difficult because many Python programs include other Python files, which means there is no single program signature that defines the execution path.

While CTE cannot use signature sets to protect traditional Python programs, tools like PyInstaller allow for the creation of a self-contained Python executable that CTE can sign and protect.

Benefits of Using PyInstaller with CTE

Using PyInstaller allows the creation of self-contained Python binary packages. These packages contain the Python interpreter, the Python program, and all required modules and libraries. Because each package includes all of the files it needs, it is immune to changes to the Python modules or libraries on the system. This not only allows CTE to sign the binary but it also adds extra protection against an attacker trying to gain access through modifications of the underlying libraries.

Another benefit is that it provides security administrators with a stable process that reasonably immune to system updates done by a system administrator. Once the application binary has been created and signed, the behavior of the program will not change because of updates made through the OS packaging system or Python's PIP package manager.

Getting PyInstaller

PyInstaller is an OpenSource project. The instructions for installing PyInstaller, as well as the documentation, are available on the project page.

Example Usage

The exact usage of PyInstaller arguments varies based on the details of the Python program being packaged into an executable. This example shows what is needed to generate a self-contained AWS CLI binary with PyInstaller. It uses the `--onefile` flag to generate a single, self-contained binary. It assumes that the pip3 Python package manager is already installed on the system.

Installing PyInstaller

The following command installs PyInstaller in `${HOME}/.local/bin/PyInstaller`:

```
pip3 install --user PyInstaller
```

Installing the Amazon AWS Command Line Utility

The following command installs the AWS utility in `${HOME}/.local/bin/aws`:

```
pip3 install --user awscli
```

Creating a Runtime Hook

The following command creates a runtime hook that changes the Python `botocore root` directory within the packaged binary. This is necessary to make sure that we use the `botocore` packaged by PyInstaller. The hook is save to a file named `change-botocore-root.py`. For more information on runtime hooks, see the PyInstaller documentation.

```
 cat <<EOF > ${HOME}/change-botocore-root.py
import sys
import botocore

if getattr(sys, 'frozen', False):
   botocore.BOTOCORE_ROOT = sys._MEIPASS
EOF
```

In order to get the AWS CLI utility working with PyInstaller, we need to specify some hidden imports as well as add some folders from the `awscli` and `botocore` packages into the executable. For more information about hidden imports and adding files to the executable, see the PyInstaller documentation.

```
 ${HOME}/.local/bin/PyInstaller  --onefile \
  --hiddenimport=awscli.handlers \
  --hiddenimport=pipes \
  --add-data=${HOME}/.local/lib/python3.6/site-packages/awscli/
data:data \
  --add-data=${HOME}/.local/lib/python3.6/site-packages/botocore/
data:data \
  --runtime-hook=${HOME}/change-botocore-root.py \
  ${HOME}/.local/bin/aws
```

The resulting binary will be saved in the current working directory as `dist/aws`.

Enable COS on an Agent with no COS Service

To enable COS on a system that has CTE installed but no COS Service:

1. Uninstall CTE completely.

2. Reinstall CTE and select **Yes** when asked:

   ```
   Do you want to configure this host for Cloud Object Storage? (Y/N) [N]: Y
   ```

Creating a COS Policy

Prerequisites for a COS Policy

1. Install Required Linux Packages

2. Generate a self-contained AWS CLI binary with PyInstaller

   Make sure to:

   a. Install PyInstaller

   b. Install the Amazon AWS Command Line Utility

   c. Create a Runtime Hook

3. Configure a CTE COS S3 Role for Guarded Buckets

User Set

- Create User Sets by Adding Members Manually

Key Rule

- Add Key Rules

- For COS keys:

  - **Algorithm**: AES

  - **Object Type**: Symmetric Key

  - **Size**: 256

  - **Encryption Mode**: CBC-CS1

  - **Key Usage**: Encrypt, Decrypt

    - **Exportable**: On

Process Set

- The **Directory** is the directory where you created the binary when you generated a self-contained AWS CLI binary with PyInstaller in step 2 of the prerequisite section above.
- The **File** is the file created during that process.

Uninstall COS from an Agent

Currently, the only method for uninstalling COS is as follows:

1. Uninstall CTE completely.
2. Delete the host/client from CM.
3. Re-register the host/client.
4. Reinstall CTE and select **NO** when asked:

   ```
   Do you want to configure this host for Cloud Object Storage? (Y/N) [N]: N
   ```

CTE with Teradata Database Appliances

CipherTrust Transparent Encryption (CTE) offers IDT-Capable GuardPoints on Linux for protecting data on raw devices. The IDT solution offers data encryption, data transformation, and access control on storage devices. The principles behind CTE's IDT-Capable GuardPoints can also be applied to protect Teradata Database Appliances. (For details about IDT, see CipherTrust In-Place Data Transformation for Linux.)

This section contains the following topics:

- IDT-Capable GuardPoints and Teradata Database Appliances
- Requirements and Considerations
- Guarding a Teradata Database Device
- Changing the Encryption Key on Teradata Devices
- Access Rules to Apply on the Teradata Database Appliance
- Replication of IDT Metadata Files Across Members of a Clique

- Generating IDT-capable Metadata for Teradata Storage Expansion
- Best Practices
- Uninstalling CTE from the Teradata Cluster
- Alerts and Errors

IDT-Capable GuardPoints and Teradata Database Appliances

The Teradata Database Appliance must be protected by an IDT-Capable GuardPoint. The In-Place Data Transformation (CTE-IDT) feature offers capabilities such as initial data transformation and access control on protected raw devices. The transformation capability transforms existing clear-text data on a device to cipher-text, and allows for subsequent re-transformation using another encryption key.

You must be familiar with CTE and CipherTrust Manager support for IDT-Capable GuardPoints before you can configure CTE to work with Teradata. For details about CTE-IDT, see CipherTrust In-Place Data Transformation for Linux.

Requirements and Considerations

Location of the CTE Private Region

+

The CTE Private Region contains the metadata CTE requires in order to support initial transformation of data on the device and subsequent data transformation to other encryption keys. By default, CTE creates the CTE Private Region at the beginning of the guarded device. If data already exists on the device, CTE requires that the device be expanded by 63 MB to make room for the CTE Private Region. The existing data in the first 63MB of the device is then migrated into the expanded space and the beginning of the device is reserved for the CTE metadata. This data relocation is completely transparent to applications and users.

With a Teradata Database Appliance, however, CTE cannot create the CTE Private Region at the beginning of the Teradata pdisk devices because the disks in the Appliance cannot be expanded. Therefore, for Teradata Databases, CTE stores the metadata in a special directory called the CTE Metadata Directory located at: `/var/`

`opt/teradata/vormetric/vte-metadata-dir`. This directory contains all of the metadata for every Teradata Database device that is protected by CTE.

While this does not affect the functionality of CTE, it does affect the way administrators need to back up the Teradata Database because both the Teradata Database and the metadata directory must be backed up together. You will not be able to restore a Teradata Database without access to the associated metadata in the CTE metadata directory.

Metadata File Access and Teradata Clusters

A Teradata cluster can contain multiple hosts. The members of a cluster that share access to pdisk devices belong to a clique. When you create an IDT-Capable GuardPoint on a pdisk, the metadata for that GuardPoint must be available to all members of the clique. CTE automatically replicates the metadata files across the members of a clique when the metadata is created or changed. For details, see Replication of IDT Metadata Files Across Members of a Clique.

Additional Requirements and Considerations

- The Teradata kernel must be at minimum version 4.4.140-96.54.TDC-default on every node of the Teradata Database Appliance on which you plan to install CTE. Refer to the release notes document for CTE releases for compatibility requirements between the kernel releases from Teradata, and the CTE releases.

Caution

- **Be sure the version of the Teradata Database is fully compatible with CTE.**
  • The Parallel Upgrade Tool (PUT) component of Teradata Database has been enhanced to discover CTE protected devices. The Parallel Upgrade Tool (PUT) component of Teradata (TDput) must be version `TDput-03.09.06.09` or higher. This PUT component must be available in your Teradata Database.
- **The Parallel Database Extensions (PDE) component of Teradata (ppde) must be version `ppde-16.20.53.07` or higher. This PDE component must be available in your Teradata Database.**
- **The Teradata I/O Scheduler (tdsched) component must be version `01.04.02.02-1` or higher. This I/O Scheduler component must be available in your Teradata Database.**
- **Contact your Teradata Customer Support Representative if you are unsure of the availability of this functionality in your Teradata cluster.**

- The CTE Agent must be installed in `/opt/teradata/vormetric` on every node in the Teradata cluster. In order to specify this location, use the `-d` option when installing CTE. For example:

  ```
  ./vee-fs-7.3.0-135-sles12-x86_64.bin -d /opt/teradata/vormetric
  ```

Note

The CTE Agent can be installed without stopping the Teradata Database service.

- The CTE metadata directory `/var/opt/teradata/vormetric/vte-metadata-dir` must be guarded by the Administrator to prevent accidental modification or deletion of the CTE metadata files. If the CTE metadata directory is not guarded, any attempt to configure or enable an IDT-Capable GuardPoint on the Teradata appliance will be rejected.

  The standard CTE policy associated with the metadata directory must:

  - Deny all users (including the root user) the ability to modify or remove any files in the metadata directory.

- Specify the key `clear_key` in the key rule so that the metadata is stored in clear text.
- The following table displays how to set up your policy for guarding the CTE metadata directory:

Security Rules	
Action	Effect
Read	Permit, Audit
all_ops	Deny, Audit
Key Selection Rules	
Key	
clear_key	

- Stop the Teradata Database service when upgrading an existing CTE Agent installation, type:

```
tpareset -x Stopping Database
```

- Stop the Teradata Database service so that CTE can rekey any guarded devices. The service must remain stopped until CTE has finished rekeying all of the guarded devices on the appliance.

Calculate Encryption/Key Rotation Time for Teradata Databases

The Thales Engineering team has a set of scripts which help to calculate the estimated encryption time for CTE on a Teradata database cluster.

Using the `dd` command, the scripts simulate the I/O sequences for encrypting `pdisks` during initial encryption. The scripts perform only read operations for simulation. Two parameters are required for simulation: the amount of data to read from each `pdisk`, and the IDT concurrency level to be applied during simulation. The larger the data size, the more sample performance data will be available for estimation. Thales recommends a minimum of 4GB for data size.

The concurrency level is the IDT concurrency level for encrypting IDT configured devices. You can specify the number of segments to transform concurrently using `-c` option of the `voradmin` command when initializing the device. Choose a concurrency level that does not affect performance of your production workload. By default, CTE-IDT transforms 8 segments concurrently, if the concurrency level has not been specified through the voradmin command. You can specify 16 for concurrency level if the backend storage for `pdisks` uses high speed flash drives, otherwise use 8.

> **Note**
>
> When choosing the concurrency level for your system you **must** consider the number of CPU cores, the total IOPS of your storage system and production workload, the size of the device to transform, and the duration for the data transformation.

The scripts collect the performance data in log files during execution. The logs are saved in `/tmp/cte_data` and `/tmp/perfmon_cte_dd.log`. The Thales team will need those logs to help you with encryption time estimation.

Obtaining the Scripts and Calculating Encryption Time

1. Download the scripts:

    - con-pdisks-dd.sh

    - con-pdisks-test.sh

2. Modify the bash variable `PDISKS` in the script: `con-pdisks-dd.sh`

    Example

    ```
    PDISKS="list of pdisks assigned to this node ..."
    PDISKS="/dev/sda1 /dev/sda2 /dev/sdb1 /dev/sdb2"
    ```

3. Divide the disks assigned to the clique, equally amongst the nodes of the clique.

4. Copy all of the scripts to the same directory on each Cluster.

5. Run `cond-pdisks-test.sh` concurrently on all of the nodes with the selected devices.

    ```
    # sh con-pdisks-test.sh `<number of GBs to read from each
    `pdisk`>`  `<concurrency level to be applied when reading
    `pdisks`>`
    ```

Example

```
# sh con-pdisks-test.sh 4  16
```

6. Contact Thales Technical Support. They will need the collected data to calculate encryption time.

Note

1. The scripts only perform read operations to generate IO load similar to the IDT encryption process, meaning, there is no write operations. Thales factors in a few parameters to estimate rekey time.
2. Thales recommends executing the scripts when the database is not busy.
3. If there are multiple cliques, and if the user requires estimation for each clique, collect the data for each clique separately.

Guarding a Teradata Database Device

To guard a Teradata Database device:

- Install CTE on the Teradata Database Appliance
- Identify Devices to be Guarded with IDT-Capable GuardPoints
- Select the Initial Configuration Method
- Initialize and Guard the Database Devices Using the Standard Initialization Method
- Automating and Reducing the Duration of Initial Data Transformation
- Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method

Note

You must use the **In-Place Data Transformation** policy type when guarding Teradata devices.

Install CTE on the Teradata Database Appliance

CTE must be installed on every node in the Teradata cluster, including any Hot Standby Nodes. When you install CTE on every node in your Teradata cluster and register each node with CipherTrust Manager, the following are required:

- You must have AES-NI key available on the host.
- The Teradata kernel must be at version 4.4.140-96.54.TDC-default, or higher, on the Teradata Database Appliance on which you plan to install CTE. Refer to CTE release notes for compatibility between CTE and the Teradata kernel releases.
- Your version of Teradata Database must be fully compatible with CTE for supporting the pdisk devices protected as IDT-Capable GuardPoints.
- The CTE Agent must be installed in `/opt/teradata/vormetric`. In order to specify this location, use the `-d` option when installing CTE. For example:

```
./vee-fs-7.3.0-135-sles12-x86_64.bin -d /opt/teradata/vormetric
```

Warning

Before you install or upgrade CTE, you must stop the Teradata Database and any other applications that access the database devices directly or though the database service. Failure to stop the applications will result in a failure to install or upgrade CTE.

Caution

Be sure to install the CTE Agent in a file system with sufficient free storage space. The minimum available free space is the number of pdisks in your cluster multiplied by 63MB.

Identify Devices to be Guarded with IDT-Capable GuardPoints

Storage devices in Teradata clusters are known as pdisks. A pdisk is a slice of a LUN exported from the backend storage system attached to the nodes in a cluster. For example, the pdisk named `/dev/pdisk/dsk304` is a symbolic link to the slice on the LUN called `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3`, as shown below:

```
 ls -l /dev/pdisk/dsk304
lrwxrwxrwx 1 root root 70 May 18 12:21 /dev/pdisk/dsk304 -> /dev/disk/b
y-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

CTE guards the devices represented by the pdisks in the Teradata cluster. In the example above, the device that you must configure and guard as an IDT-Capable GuardPoint is `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3`.

!! note

```
You cannot use the symbolic link name when initializing or guarding the
storage devices.
```

To identify the disks available to the Teradata cluster and require CTE protection, look in `/dev/pdisks` on each node in the cluster. Any symbolic links that point to a Linux device path are the disks that should be initialized.

> **Caution**
>
> **Do *not* enter the path name of the pdisk device when initializing it for guarding or in the CipherTrust Manager when you actually guard the device.**

Select the Initial Configuration Method

Select the initial configuration method that you want to use. Thales supports two methods for the initial configuration of Teradata Database devices:

- **Standard Initialization Method**

  This method configures devices for initial encryption of existing clear-text data on the database devices. The time required to encrypt may take hours or even days, depending on the volume of data, the number of database devices and nodes, and the bandwidth of the storage back-end of database devices. For details, see Initialize and Guard the Database Devices Using the Standard Initialization Method.

- **Backup/Restore Initialization Method**

  This method skips the initial encryption step but requires a full backup of the database and the engagement of Teradata Customer Support to assist you with some configuration steps at the Teradata database level in preparation for the full restore of the database after the database devices are protected. The length of time required for this method depends on the length of time it takes to back up, and then fully restore, your Teradata database. For details about using this method, see Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method.

Initialize and Guard the Database Devices Using the Standard Initialization Method

The following procedure describes how to perform the initial configuration of your database devices using the Standard Initialization Method. This procedure encrypts the existing data in place and does not require you to backup your Teradata database when initially deploying CTE. It may, however, take several hours, or even days, to complete depending on the volume of data, the number of database devices and nodes, and the bandwidth of the storage back-end of database devices.

> **Warning**
>
> **For each device, you must designate *one and only one* of the nodes in the cluster as the initial node. This is the node on which you plan to initialize and guard the device for the first time. The designated node must be the only one that guards the device until the entire initial data transformation process has completed. DO NOT initialize or guard any device on multiple nodes in the cluster simultaneously, because multiple nodes attempting to transform the same data can corrupt the data on the entire device**.

1. Shut down the Teradata Database. You cannot initialize an online database device.

2. For each device, designate one of the nodes in the cluster as the initial node that you will use for the initial data transformation when the device is guarded for the first time.

3. Log into the designated node in the Teradata Database Appliance. Type:

   ```
    voradmin idt config [ external] [new|xform] [-c <n>] <device-name>
   ```

   - **[new|xform]** (required)

     Indicates whether data already exists on the device. If the device contains no data, specify `new`. If the device contains data that you want to keep, specify `xform`. Most installations of Teradata Appliance are expected to have pdisks populated with data, therefore, most often you will use the `xform` option. When you use `xform`, CTE will transform all existing data on the device from clear-text to cipher-text as soon as you guard the device on the Appliance. The device will be inaccessible until the transformation is complete, and the device must remain offline to the Teradata Database service during the entire transformation process. No user access will be permitted until all of the data has been transformed.

- **-external** (required)

  You *must* use this option when initializing any Teradata device. With this option, CTE writes the CTE Private Region to a metadata file located in the CTE metadata directory. For details, see Location of the CTE Private Region.

- **-c <n>** (optional)

  If you use this option, CTE sets the number of data transformation jobs to run in parallel to the number specified in `<n>`. `<n>` can be an integer between 1 and 60, (default: 8).

  Each data transformation job transforms 1MB worth of data and requires CPU resources in addition to three I/O operations as part of data transformation. Each job reads 1MB of data from the device, preserves the data in the CTE private region, rekeys the data to cipher-text, and writes the transformed data to the device. If you increase the number of parallel jobs, the data transformation process will complete faster but there will be an increased performance impact on the system. Only increase the `-c` option if you are certain that the system resources are available to handle the additional load.

  The value for the `-c` option you specify here remains in effect for all subsequent data transformations (such as any data rekeys) until you specify a new value.

- **<device-name>** (required)

  The name of the Teradata Database device that you want to initialize.

  For example:

  ```
  voradmin idt config -external xform -c 20 \
  /dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
  ```

4. Repeat the `voradmin idt config` command for each device that you want to initialize. If you want to distribute the initial data transformation or subsequent rekey load on all of the disks across all the nodes in the cluster, make sure that you run the `voradmin idt config` command for each device on the node you

designated for data transformation, *excluding* the HSN node. The node on which you run the `voradmin idt config` command is the node that performs the data transformation on the device. Do *not* designate the HSN node for data transformation because the devices on the HSN node may be reserved and therefore not available for IO operations.

Guard the Devices as IDT-Capable GuardPoints

Warning

For each device, you must designate a node in the cluster as the initial node on which you plan to initialize and guard the device for the first time. The designated node must be the only one that accesses the device until the entire initial data transformation process has completed. This requires guarding each device at the designated client level rather than at the client group level. DO NOT initialize or guard any device on multiple nodes in the cluster simultaneously, because multiple nodes attempting to transform the same data can corrupt the data on the entire device.

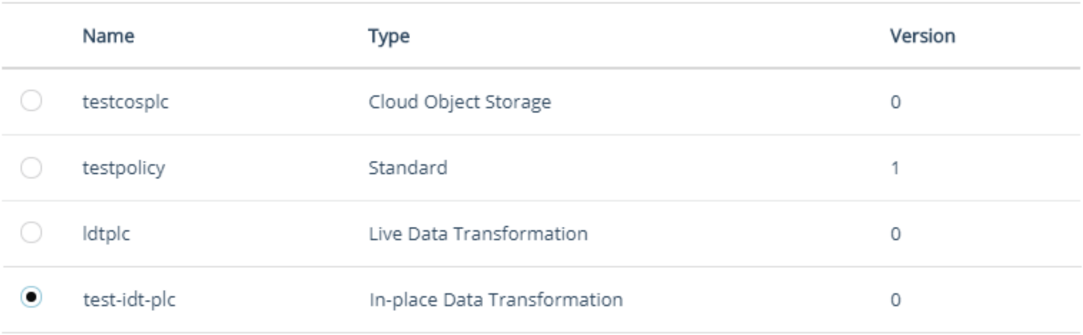
To create an IDT GuardPoint:

1. On the **GuardPoints** tab, click **Create GuardPoint**.

2. Select a **Policy**. This is a mandatory field.

   a. Click **Select** next to the Policy field.

   b. Select an *In-place Data Transformation* policy. If no policy exists, create one, as described in Creating Policies.



	Name	Type	Version
○	testcosplc	Cloud Object Storage	0
○	testpolicy	Standard	1
○	ldtplc	Live Data Transformation	0
●	test-idt-plc	In-place Data Transformation	0

   c. Click **Select**.

Note

When an IDT policy is selected, the read-only **In-place Data Transformation** toggle is displayed on the Create GuardPoint dialog box.

3. Select the **Type** of device to protect. This is a mandatory field. The options for a **In-place Data Transformation** policy are:

Type	Description
Auto Raw or Block Device	Select for IDT policies for raw (block) devices.
Manual Raw or Block Device	Select for IDT policies for raw (block) devices to be guarded manually.

Note

Manual Raw or Block Device are guarded and unguarded (for example, mounted and unmounted) by running the `secfsd -guard` and `secfsd -unguard` commands. Do not run the `mount` and `umount` commands to swap GuardPoint nodes in a cluster configuration.

4. Specify the **Path** to be protected. This is a mandatory field. Options to specify the GuardPoint paths are:

   - **Enter/Browse Path: Select this option, and enter the GuardPoint paths by either typing or clicking the Browse button.**

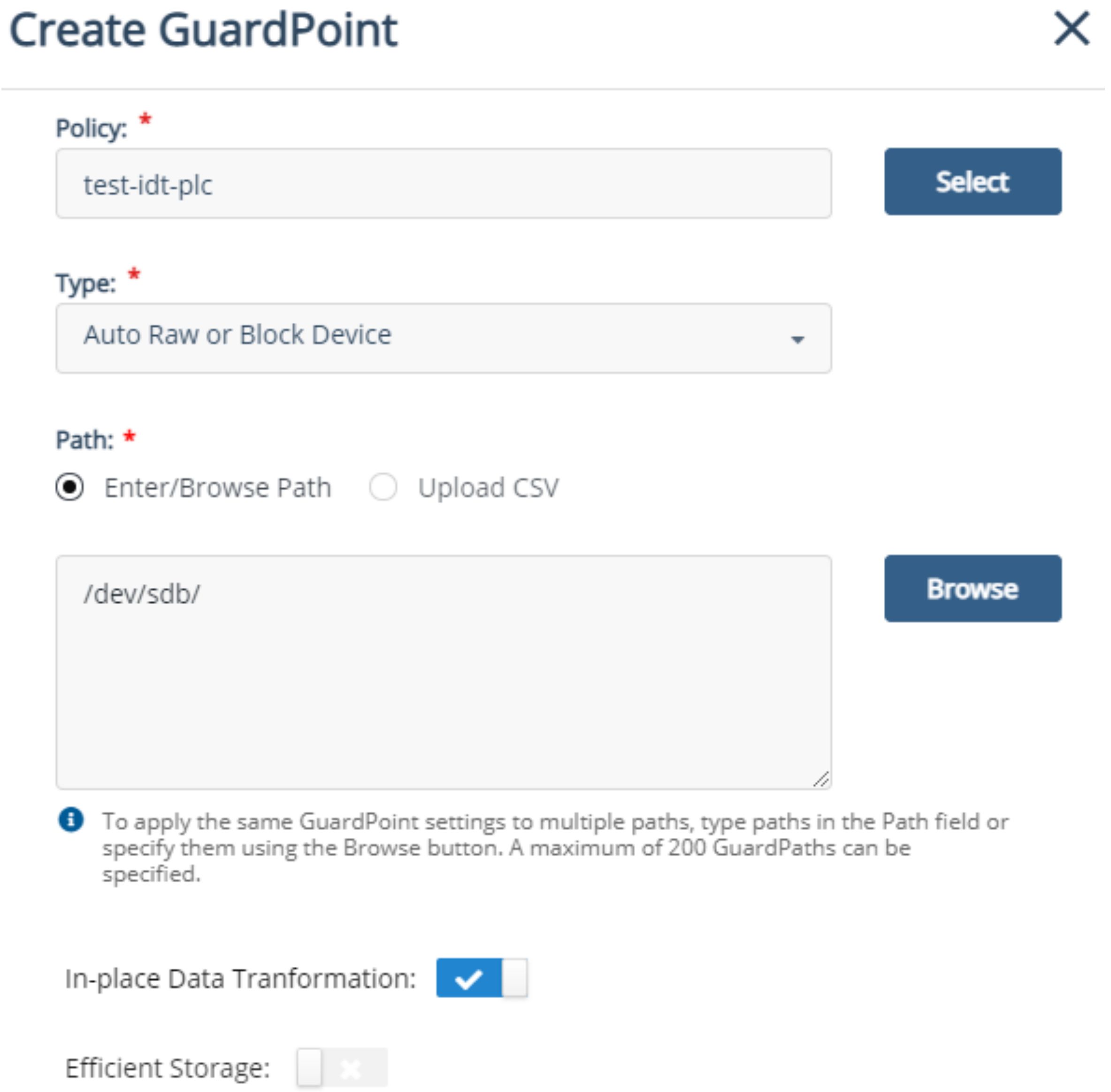
Note

A maximum of 200 GuardPaths can be specified using the **Enter/Browse Path** option.

- **Click Browse to select a path by browsing the client file system. This method prevents typographical errors and verifies client availability. This is the recommended method to specify individual paths.**

  File system of a client that is not registered with the CipherTrust Manager cannot be browsed.

- **Alternatively, if you know the path, manually enter full paths of one or more directories in the given text box. Enter one path per line.**



Create GuardPoint

Policy: *

test-idt-plc

Type: *

Auto Raw or Block Device

Path: *

● Enter/Browse Path ○ Upload CSV

/dev/sdb/

To apply the same GuardPoint settings to multiple paths, type paths in the Path field or specify them using the Browse button. A maximum of 200 GuardPaths can be specified.

In-place Data Tranformation: ✓

Efficient Storage: ✗

5. Make sure the In-place Data Transformation option is selected.

6. Click **Create**. A message appears prompting to confirm the reuse of these GuardPoint settings on another path.

   - Click **Yes** to use the same settings on another path. The Use Settings on Another Path dialog box is displayed. Perform the following steps:

     a. Specify a new **Path**.

     b. Click **Add Path**. The newly added path appears under the Paths list on the left. Similarly, add as many paths as required.

     c. Click **OK**.

   - Click **No** if you do not want to use the same settings on another path.

CipherTrust Manager pushes the policy and the GuardPoint configuration to the node in the cluster. The CTE agent on the node writes the IDT Device Header into the CTE Private Region in the local CipherTrust Transparent Encryption metadata directory on the node.

If this is a new device, the status immediately changes to guarded. If there is existing data on the device, CTE begins transforming the data from clear-text to cipher-text as soon as the GuardPoint configuration is available and the device status changes to guarded. The device will remain inaccessible until this data transformation completes. The length of time required to transform the data depends on the amount of existing data and the number of parallel data transformation jobs specified on the `voradmin config` command. To see the data transformation progress, use the `voradmin idt xform status <device-name>` command, as described in Viewing Device and Data Transformation Status.

Devices with existing data are transformed from clear-text to cipher-text using the encryption key specified in the selected policy through the In-Place Data Transformation (CTE-IDT) process. For details on how CTE does data transformation on IDT GuardPoint, see CipherTrust in-Place Data Transformation for Linux. ccc

After the device is initialized and guarded, the protected device must be accessed through the device pathname corresponding to the secvm device. For example, if you guard the Linux device `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3,` the pathname becomes `/dev/secvm/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3` as soon as the process is complete.

7. After all of the Teradata Database devices have been guarded, disable the guarded IDT-GuardPoints on each designated node and then enable those GuardPoints at the Client Group level on all the nodes in the clique.

8. After guarding your devices and before starting your database, you must change the current configuration of your cluster to reflect the status of the pdisk devices guarded as IDT GuardPoint. See the **Rebuild Vconfig Only** option of the Teradata Parallel Upgrade Tool (PUT) service to apply the guarded status of the pdisk devices into your database configuration. The **Rebuild vconfig** process applies the guarded configuration status of each pdisk and resets the pdisk symbolic link to the CipherTrust Transparent Encryption `secvm` device.

Warning

You must complete this step before starting your database. Failure to do so will result in database failures and potential corruption of your database.

For example, the pdisk `/dev/pdisk/dsk304` is linked to the secvm device after **Rebuild Vconfig Only** commits the guarded status of pdisk on each node:

```
# ls -l /dev/pdisk/dsk304
lrwxrwxrwx 1 root root 70 May 18 12:21 /dev/pdisk/dsk304 -> /dev/s
ecvm/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part
```

9. After all of the Teradata Database devices have been guarded in the clique and the **Rebuild Vconfig Only** on TDput has been executed, start the Teradata Database, type:

```
# /etc/init.d/tpa start
```

Viewing Device and Data Transformation Status

After you guard a Teradata Database device, you can view the status of that device using the `voradmin idt status [xform] <device-name>` command, where:

- **xform** (optional)

If you specify this option, CTE shows the status of any data transformation processes happening on the device. If you do not specify this option, CTE displays the IDT Device Header for the device.

- **<device-name>** (required)

The name of the Teradata Database device that you want to initialize. This must be the actual device name and not the symbolic pdisk name.

For example, if you want to view the status of device `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3`, you would enter:

```
 voradmin idt status /dev/disk/by-id/
tdmp-360080e500043092c0000b46f5c34c018-part3
```

If you want to check the data transformation progress on the device `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3`, you would enter:

```
 voradmin idt status xform \
      /dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

The **Status** field displays **In-Progress** if a data transformation process is running, and **Completed** if the process has finished.

Automating and Reducing the Duration of Initial Data Transformation

CTE for Teradata allows Teradata customers to run their database services on top of a high performing encrypted storage infrastructure, solving many security and compliance concerns. Teradata systems are usually mission critical and cannot afford to be offline for significant periods of time. The problem, therefore, is how to convert an existing Teradata system from one that is unencrypted, to one that is encrypted. Thales calls this conversion process rekeying, or data transformation.

The following steps outline the process the Teradata operations team should follow to encrypt the platform while minimizing system downtime and operational disruptions. The process leverages the fact that a Teradata system is composed of multiple disks, which create a redundant platform, and effectively approaches the conversion process by breaking it into smaller, more manageable, and less disruptive steps. Operators should prepare for data transformation and schedule time to complete the transformation process while the Teradata database services are down. The process supports recovery from unexpected IO failure during the transformation process of migrating data from unencrypted to encrypted. This document describes the procedure and execution of the transformation process using a script that orchestrates the entre transformation process across multiple nodes in a clique, and significantly reduces the entire data transformation time.

Warning

Thales recommends that you backup the Teradata database before starting the transformation process.

Use the `cte-pdisk-idt.sh` script to perform the initial rekey using embedded metadata on the Teradata `pdisks` that will be configured for IDT using external metadata. This reduces the time needed to encrypt the `pdisks` because the external metadata is a bottleneck when performing rekey on multiple concurrent devices. The script works by saving the last 63 MB of a device to a file, and then configuring the device as IDT with embedded metadata. The last 63 MB is saved because devices configured for IDT with embedded metadata relocate the first 63 MB to the end of the device to make space for the metadata. Upon completion of rekey, the script then restores the first 63 MB from the end of the device and the last 63 MB for the saved file before converting the device to IDT with external metadata.

Note

- This script **only** works for CTE 7.3.0 or subsequent versions
- This script **only** works for initial rekey.

Getting Help

Recovery from interruption during the preparation step requires manual intervention. Recovery from unexpected failure occurs when the GuardPoints are guarded again. You can then restart the script in the monitor mode to resume monitoring the transformation progress.

In the event that the script is interrupted, or a system crash, contact Thales Support. If possible, include the following information to facilitate recovery:

- Output of the script
- Output of the `secfsd -status guard` command (if the interruption did not crash the system)
- Contents of the the `/tmp/pdisks.list` file
- Contents of the `/tmp/monitor-idt.disks` file

- Contents of the `/opt/teradata/vormetric/agent/secfs/.sec/conf/sod_config file`
- Contents of the `/var/opt/teradata/vormetric/vte-metadata-dir` directory (including file sizes)

Install and Setup

1. Install the agent in `/opt/teradata/vormetric` and setup the external metadata directory `/var/opt/teradata/vormetric/vte-metadata-dir` on all nodes in the clique.
   - **See** Install CTE on the Teradata Database Appliance **for more information on installing CTE on Teradata.**

> **Note**
>
> Your policy must allow `dd`, `mv`, `rm`, and `stat` access to the external metadata directory and `dd` access to the device.

2. Copy the `cte-pdisk-idt.sh` script from `/opt/teradata/vormetric/agent/secfs/.sec/bin/` to `/root`.
3. Shutdown Teradata, type:

```
tpareset -x -f 'Setting up CTE'
```

4. Generate a list of disks to encrypt, type:

```
ls -l /dev/pdisk/dsk* | awk '{ print $(NF) }' > /tmp/all_pdisks.list
```

> **Note**
>
> This command generates a file with a list of disks in the format: `/dev/disk/by-id/*`.

5. Divide `/tmp/all_pdisks.list` into separate files based on the number of active (non-HSN) nodes in the clique. For example, if the list contains 300 disks and there are 4 active nodes, there should be 4 files with 75 disks in each.

6. Copy each file as `/tmp/pdisks.list` to the corresponding host that will oversee encrypting those disks. For example, if there are 300 disks in a 4-node clique, each node should have a unique list of 75 disks.

7. Add a manual IDT GuardPoint on each node for the disks in `/tmp/pdisks.list`

> **Note**
>
> The script only toggles the GuardPoints in the `/tmp/pdisks.list` file so it is safe to add manual GuardPoints for all of the devices on all of the nodes. However, automated metadata distribution does not work if the GuardPoint is present on all nodes as it expects the node performing transformation to be the only node with the GuardPoint at this point.

Configure and Initialize

1. To configure the devices for IDT on each node, save the last 63 MB, and start the initial rekey, type:

   ```
   /root/cte-pdisk-idt.sh -m pre -y -f /tmp/pdisks.list
   ```

2. To monitor transformation with the following and wait for transformation to complete, type:

   ```
   /root/cte-pdisk-idt.sh -m mon
   ```

3. To finish the setup, type:

   ```
   /root/cte-pdisk-idt.sh -m post -f /tmp/pdisks.list
   ```

   This restores the data to the first and last 63 MB of the device, restores the external metadata, and sets the `pdisk symlinks` to their associated `secvm` device.

4. Verify that identical metadata is present for all devices on all of the nodes, type:

   ```
   ls -1 /var/opt/teradata/vormetric/vte-metadata-dir | wc -l
   ```

   If identical metadata is not present on all of the nodes in the same clique, manually synchronize the contents of the external metadata directory.

5. Unguard all of the nodes from the security server and switch to auto-guard GuardPoints.

6. Use the Teradata PUT tool to update the `pdisk` symbolic links permanently.

7. Start Teradata to verify that the database can start, type:

```
/etc/init.d/tpa start
```

8. Clean up the files created by the script, type:

```
rm -f /tmp/all_pdisks.list /tmp/pdisks.list.done /tmp/
cte_pdisk_idt_pre.out
```

Script Help Output

```
    # /root/cte-pdisk-idt.sh -h
```

Response

```
    Usage   /root/cte-pdisk-idt.sh [-h] [-y] [-t #] [-m <mon|pre|
post>]
                < [-p </dev/pdisk/dsk#>] </dev/disk/by-id/scsi-...>
                -f <batch_file> >

    -f  <batch_file> File with list of disks to transform
                File should have one entry per line
                Format is,
                    /dev/disk/by-id/tdmp-xxx
                    /dev/disk/by-id/tdmp-yyy
                    ...
                or
                    /dev/pdisk/dsk# -> /dev/disk/by-id/tdmp-xxx
                    /dev/pdisk/dsk# -> /dev/disk/by-id/tdmp-yyy
                    ...

                Format for the latter option can be generated with,
                    ls -l /dev/pdisk/dsk* | awk '{ print $(NF - 2), \
                    $(NF - 1), $NF }'

    -m mon      Monitors IDT status on disks undergoing transformation
started with the '-m pre' option

    -m full     Performs the entire transformation process (default)

    -m pre      Performs steps to set up and begin encryption of a dev
ice

    -m post     Performs steps after device has been encrypted to rest
ore borrowed space
                With the 'full' mode, the script will wait while the d
evice undergoes encryption. The 'pre' mode will start
                encryption and then terminate the script and wait for
a user to rerun the script with the 'post' mode after encryption is co
mplete.

    -t #        Number of threads to use during IDT (default: 16, range
: 1 - 60)

    -p          Pdisk to link to after transformation is complete

    -y          Skips user prompts

    -h          Displays this help
```

Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method

The Standard Initialization Method encrypts the Teradata database devices using In-place data transformation. Because Teradata Appliance models range in both storage and node sizes within a cluster, the time it takes to encrypt the existing data in-place on these large scale models may exceed the desired time frame.

To address this issue on an existing Teradata database, you can configure CTE on Teradata Appliances using the Backup and Restore method to retain existing data. The total time required to configure your Teradata Appliances with CTE using this method depends on how long your backup/restore application takes to restore the database.

To initialize and guard the devices using the selected initialization method, you must:

1. Backup the entire Teradata database, including all meta files, required for a full restore of that database.
2. Install and configure CTE on the Teradata database devices.
3. Work with Teradata Customer Support to initialize a `sysinit` on the Teradata Appliance cluster in preparation for a full database restore.
4. Perform a full restore of the Teradata database from the backup to the CTE protected devices. CTE automatically encrypts the data as it is restored to each protected device.

Prerequisites

Before starting this process, complete the following prerequisites:

- Make sure you have met the requirements described in Requirements and Considerations.
- Create or identify the XTS/CBC-CS1 AES 256 key that you will use for the In-Place Data Transformation policy that you will apply to the IDT-Capable GuardPoints.
- Create a **Standard** policy that will be used to guard the CTE Metadata Directory (`/var/opt/teradata/vormetric/vte-metadata-dir`).
- Create an **In-Place Data Transformation** policy that will be used to guard the Teradata Database Devices. This policy must use an XTS/CBC-CS1 AES 256 key.
- Identify all of the devices that need to be guarded as described in Identify the Devices to Be Guarded.

Procedure

1. Using a Teradata certified backup application, backup your entire Teradata database. The backup needs to include the Data Dictionary, the user database,

and everything else that is required for a successful restore of your database. Contact your Teradata Customer Support representative for the requirements for a full backup.

2. After confirming that your backup of the Teradata database completed successfully, shutdown the Teradata database, type:

```
tpareset -x -f "shutting down for CTE installation"
```

3. Install CTE in `/opt/teradata/vormetric` on all nodes in the Teradata cluster and register them to a CipherTrust Manager server as described in Install CTE on the Teradata Database Appliance.

```
./vee-fs-7.3.0-135-sles12-x86_64.bin -d /opt/teradata/vormetric
```

4. Create the CTE metadata directory (`/var/opt/teradata/vormetric/vte-metadata-dir`) on all nodes in the cluster.

```
. pcl -shell "mkdir -p /var/opt/teradata/vormetric/vte-metadata-
dir"
```

5. If you are using the Teradata Intelibase model, complete the following step. If you are using the Teradata Inteliflex model, proceed to Step 6.

   a. Log on to the CipherTrust Manager, click the **Hosts/Clients** tab, and set the GuardPoint to the CTE Metadata Directory (`/var/opt/teradata/vormetric/vte-metadata-dir`) using the Standard policy that was created for the metadata directory on all the nodes in the Teradata cluster.

   b. On each node, identify the list of disks that will be configured and guarded as "new" devices as described in Identify the Devices to Be Guarded.

   c. On each node, configure each disk to be guarded as a new IDT-Capable device using the `voradmin idt config external new` command.

```
voradmin idt config -external new \
/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

> **Note**
>
> Make sure that you configure each device using the new option. This option tells CTE to guard the device without transforming the data.

d. Log on the CipherTrust Manager, click the **Hosts/Clients** tab, and select one of the host in the clique to set the GuardPoints for all the devices that have been initialized as new IDT-Capable devices using the In-Place Data Transformation policy that you created.

Repeat this step for all other hosts until all initialized devices have been guarded with an In-Place Data Transformation policy.

6. If you are using the Teradata Inteliflex model, complete the following step. Otherwise, proceed to Step 7.

   a. Designate one of the nodes in a clique as the node on which you will perform the first-time initialization and guarding procedures for each device. For example, if you have a 4 clique cluster, you must designate a total of 4 nodes, one from each clique.

   b. Log on to the CipherTrust Manager, click the **Host/Client Groups** tab, and create a host group for each clique. For example, if your Inteliflex cluster has 4 cliques, you would create 4 host groups, one for each clique. For example, cluster-clique-1, cluster-clique-2, cluster-clique-3, and cluster-clique-4.

   c. On the **Host/Client Groups** tab, click the name of one of the client groups you just created. Click on the **Members** tab and add the designated node from the clique as a member of this client group.

   Repeat this step until all designated nodes have been added to the client group that matches their clique. In the previous example, you would have one designated node in each of the 4 client groups that you created.

   d. On the **Host/Client Groups** tab, click the name of one of the host groups that you just created. Click on the **GuardPoints** tab and add a GuardPoint for the CTE Metadata Directory (`/var/opt/teradata/vormetric/vte-metadata-dir`) using the Standard policy that was created for the metadata directory on all the nodes in the Teradata cluster.

Repeat this step for each Host/Client Group you created for your cliques. In the previous example, you would add the metadata directory GuardPoint in each of the 4 host groups that you created.

e. On the designated node for each clique, identify the list of disks that will be configured and guarded as a “new” IDT device as described in Identify the Devices to Be Guarded.

f. Configure each disk device on the designated node to be guarded as a new IDT-Capable device, type:

```
voradmin idt config -external new \
 /dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

Note

Make sure that you configure each device using the new option. This option tells CTE to guard the device without transforming the data.

g. For each of the other designated nodes you identified, identify the list of devices to be guarded and configure them using the `voradmin idt config external new` command.

h. Repeat this configuration process until all devices on all designated nodes that you want to guard have been configured as new IDT-Capable devices.

i. On the **Host/Client Groups** tab, click the name of one of the clientgroups you just created. Click on the **GuardPoints** tab and add a GuardPoint for each device that you initialized as a new IDT-Capable device using an In-Place Data Transformation policy.

j. Repeat this step on each one of the host groups you created.

The guarded devices will immediately be set as guarded on the designated node through the client group. However, as part of Teradata cluster support, each guarded device will have its metadata file replicated across all nodes in a clique as described in Replication of IDT Metadata Files Across Members of a Clique. You must wait for metadata replication to complete for each clique in a cluster before continuing to the next step. This process takes approximately 1 second per disk that was guarded. So if you guarded 10 disks, you need to wait approximately 10 seconds. Verify that the number of metadata files on each clique matches with the designated node using the following command:

```
pcl -shell "ls /var/opt/teradata/vormetric/vte-metadata-dir |
wc -l"
```

The number of metadata files on each node in the clique should also match the number of devices in that clique. For example, if you have a clique with 10 devices in it, then the results of the above `pcl` command should show 10 metadata files on each node in the clique.

k. After metadata replication is completed on all cliques, log on the Key Manager and click the **Hosts/Client Group** tab, then click the name of one of the client groups you just created. Click the **Members** tab and add the rest of the nodes that belong to that clique to the Host Group. After all the nodes have been added, the Key Manager automatically pushes the existing GuardPoints out to the newly added members.

l. Repeat this step for each one of the Host Groups that you created for your cliques.

7. Verify that all of the nodes in a clique show that all of the devices as guarded.

```
pcl -shell "secfsd -status guard"
```

8. Work with Teradata Customer Support to perform a `sysinit` of the Teradata appliances in preparation for the full database restore.

9. Restore your Teradata database from backup.

Changing the Encryption Key on Teradata Devices

To meet various compliance requirements, you may want to change the key that CTE used to encrypt the Teradata Database devices. This process is called “Key rotation” or “Rekey”.

If you want to rekey a Teradata Database device, you can either:

- Follow the process for standard IDT-Capable GuardPoints. For details, see Changing the Encryption Key on Linux IDT-Capable Devices.
- Use the Backup/Restore method as described in Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method. The only differences between the initial configuration described in that section and rekey process are:
    - You do not need to change the policy assigned to the CTE Metadata Directory.
    - You need to create a new In-Place Data Transformation policy, that specifies a key rule with the existing key used to encrypt the Database devices in the **Current Key** field, and the new XTS/CBC-CS1 AES 256 key that you want to use to encrypt the data in the **New Key** field.
    - When you re-guard the designated node in each clique, make sure that you use the new policy so that CTE knows it needs to re-encrypt the data with the new key after you restore the devices from the backup.

Note

Make sure that you stop the Teradata database before you rekey the devices.

Warning

Rekeying a device must be executed only on *one and only one* of the nodes in the cluster. DO NOT prepare the device for rekey or guard the device with the new rekey policy on multiple nodes in the cluster because doing so initiates the data transformation process on the device on multiple nodes. The simultaneous attempts to rekey the data can cause data corruption of all of the data on the entire device.

Access Rules to Apply on the Teradata Database Appliance

This section provides the instructions for creating a sample policy and signature sets specific to a Teradata Database Appliance to deny unprivileged users access to clear-text data on guarded devices. Such a policy requires the inclusion of specific security rules to restrict access to the Teradata Device GuardPoints for specific set of users, groups, and/or processes.

> **Note**
>
> You **must** add binaries to client settings to allow trusted processes to access policies. Failure to do so results in a `User not Authenticated` error which prevents access for user-based policies. For example, when using `tvsa_agent` for a Teradata cluster, you must add `/usr/pde/bin/tvsa_agent` as an authenticator in the client settings.

1. In the CipherTrust Manager, edit the host/client entry for the Teradata Database Appliance.

2. Go to the **Host/Client Settings** tab and add the following entries to the list of binaries:

   - |authenticator|/usr/pde/bin/pdemain
   - |authenticator|/opt/teradata/sm3g/bin/tdsmagent
   - |authenticator|/usr/pde/bin/pcl
   - |authenticator|/opt/teradata/TDput/bin/putservices
   - |authenticator|/usr/pde/bin/tvam

3. Create a signature set for system processes that require access to guarded devices. The following system processes on the Teradata Database Appliance must be permitted access:

   - /usr/lib/systemd/systemd-udevd
   - /sbin/dmsetup
   - /opt/teradata/vormetric/agent/vmd/bin/get_disks
   - /sbin/pvdisplay
   - /sbin/lvdisplay
   - /usr/sbin/parted

   You can include additional processes as needed.

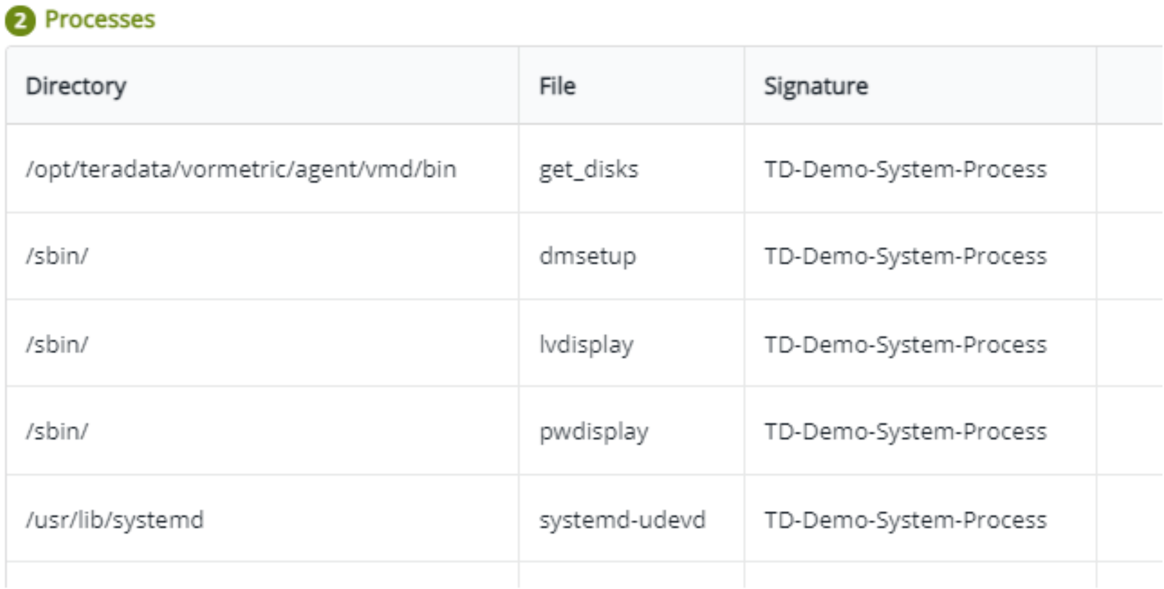
4. Create a signature set for Teradata Database processes that require access to guarded devices. The following directories contain the Teradata binaries that require access:

   - /usr/pde/bin/*

  - /usr/tdbms/bin/*
  - /opt/teradata/gsctools/bin/*
  - /opt/teradata/TDput/bin/*

  You can include additional processes as needed.

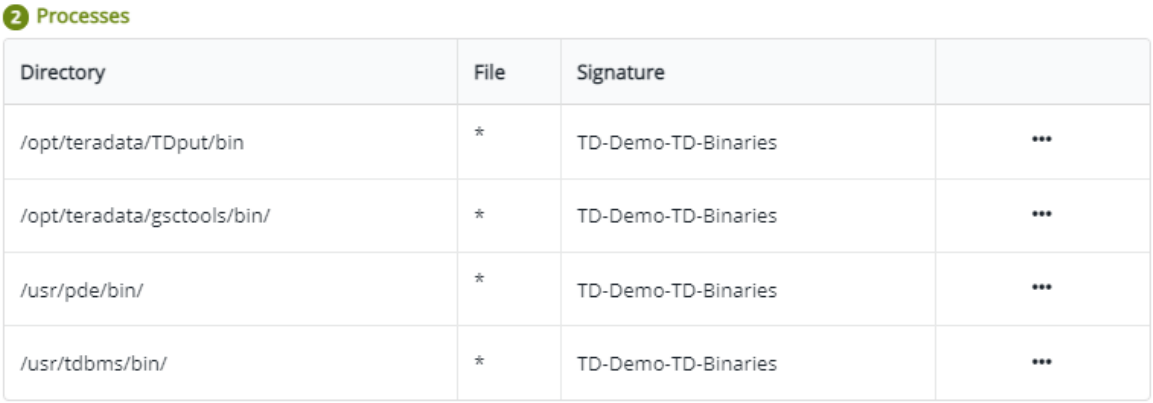
5. Create a system-level process set to associate the system-level processes listed in step 3, with the signature set you created in step 3. In the following example, this process set is called TD-Demo-system-process.



2 Processes

Directory	File	Signature	
/opt/teradata/vormetric/agent/vmd/bin	get_disks	TD-Demo-System-Process	
/sbin/	dmsetup	TD-Demo-System-Process	
/sbin/	lvdisplay	TD-Demo-System-Process	
/sbin/	pwdisplay	TD-Demo-System-Process	
/usr/lib/systemd	systemd-udevd	TD-Demo-System-Process	

6. Create a Teradata Database process set to associate the signature set for the Teradata Database binaries created in step 4.



2 Processes

Directory	File	Signature	
/opt/teradata/TDput/bin	*	TD-Demo-TD-Binaries	•••
/opt/teradata/gsctools/bin/	*	TD-Demo-TD-Binaries	•••
/usr/pde/bin/	*	TD-Demo-TD-Binaries	•••
/usr/tdbms/bin/	*	TD-Demo-TD-Binaries	•••

7. Create the user set for the Teradata trusted group on your appliance.

8. If you have any other trusted groups that you want to include in your security rule, add a user set for each one of those trusted groups. You can add as many user sets to the policy as you need.

9. Create a user set for the root user and any other privileged users you want to add.

10. Verify that you have the complete signature sets, process sets, and user sets as shown in the examples below.

11. Apply the policy to the Teradata Device GuardPoints.

Replication of IDT Metadata Files Across Members of a Clique

For Teradata, metadata for IDT-Capable GuardPoints is stored in external files in the CTE metadata directory. Because the Teradata appliance is a cluster of multiple hosts/ clients that share access to the same devices across multiple nodes, a metadata file must be replicated across the nodes that are in the same clique within the cluster, including Hot Standby nodes. Replication is required when the initial data transformation has been completed, and it is required again during subsequent rekeys. The availability of the metadata files on all members of the clique is critical for high availability of Teradata database in the event of a node failure because the data on the device cannot be accessed without the encryption key stored in the metadata.

Upon completion of data transformation on each device, the CTE Agent on that device automatically replicates the metadata file for that device to the other nodes in the clique using the Teradata `pcl` command. CTE uses `pcl` to both determine the other members of the clique and to replicate the metadata to those other members. When the metadata is replicated on the remote nodes, any existing metadata for the recently-transformed device already on those nodes is replaced with the updated metadata files sent by the CTE Agent. This replacement is achieved by sending the updated metadata file to all of the remote nodes through `pcl` and replacing each existing metadata file on the remote nodes with the most recent metadata files.

Specific Issues to Consider

This section describes specific problems that may be encountered during data transformation and metadata replication. Manual user intervention will be required to recover from the reported issues. For troubleshooting and recovery steps, see Alerts and Errors.

General PCL Error

If the `pcl` command fails during the replication process, a message indicating the error will be logged. The messages are tagged with IDT-TD-ALERT. For example:

- IDT-TD-ALERT: Node did not respond to `pcl` command
- IDT-TD-ALERT: Failed to distribute IDT-Capable metadata file to remote nodes

These errors indicate that the node specified in the first error did not respond to a `pcl` command during metadata distribution. As a result, the metadata distribution must be manually performed across the clique before access to the device is possible.

Offline Node in Clique During Data Transformation

If a node is offline during data transformation and metadata replication process, CTE will log a message that metadata replication to the target node failed. The administrator will be required to manually replicate the metadata file to the offline node when the node comes online. The metadata file must be replicated before the database is brought up.

To do so:

- Run the `voradmin td distribute <device name>` command to distribute the metadata file of each device. The command will copy, or update, the metadata file on the agent that has come online. For example:

```
 voradmin td distribute \
/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

Adding a New Node to a Clique

When you add a new node to an existing clique, the metadata files of all disks shared in the clique must be manually replicated to the newly added node before any of the guarded devices on the new node are enabled. This should be done manually using the `voradmin td distribute` command, as described above, by the Teradata administrator as part of joining the cluster.

Interoperability with Host Groups

For Teradata, when you create a new IDT-Capable GuardPoint using the `voradmin idt config xform` command, or when you subsequently rekey an existing IDT-Capable GuardPoint with the `voradmin idt rekey` command, the device must be intialized and guarded on one and only one of the nodes in the cluster. That means the IDT-Capable GuardPoint cannot be part of a client group when an IDT-Capable GuardPoint is created or rekeyed because membership in a client group means that any data transformation on any member of the client group is initiated for that member on all nodes in the client simultaneously. In the case of a Teradata cluster, multiple nodes simultaneously trying to perform data transformation on a particular device can lead to data corruption of all data on the entire device.

After the device has been guarded, or rekeyed, and the metadata files have been replicated to other members of the Teradata clique, then you can then rejoin the clientwith the client group.

To configure a new device whose host/client is part of a host/client group:

1. Make sure that there is no GuardPoint for the device at the client group level.
2. Designate one of the nodes in the cluster as the node that you will use to initialize the GuardPoint and for the initial data transformation when the device is guarded for the first time.
3. On the designated node, initialize the device using the `voradmin idt config -external xform <device name>` command. For details, see Initialize and Guard the Database Devices Using the Standard Initialization Method.
4. Guard the device on the designated node using an In-Place Data Transformation policy. For details, see Guard the Devices as IDT-Capable GuardPoints on CM.
5. Wait for the data transformation to complete on the host and for CTE to replicate the metadata to the other members of the clique.
6. You can verify that the metadata file has been distributed to the other nodes in the clique by running `md5sum /var/opt/teradata/vormetric/vte-metadata-dir/vormetric/secvm_<device>_metadata` on each node in the clique.
7. Remove the IDT-Capable GuardPoint you added earlier in this procedure.
8. Guard the device through the client group to make sure that all nodes in the cluster recognize it as guarded.

** To rekey a new device whose host/client is part of a host/client group:**

1. Unguard the IDT-Capable GuardPoint through the client group and make sure that it has been removed from all nodes in the cluster.

2. Designate one of the nodes in the cluster as the node that you will use to prepare the GuardPoint for rekeying and to perform the data transformation when the device is guarded with the new policy.

3. On the designated node, prepare the device for rekey using the `voradmin idt rekey <device name>` command.

4. On the other nodes in the clique, make sure that the device metadata has been renamed running `ls /var/opt/teradata/vormetric/vte-metadata-dir/vormetric/secvm_<device>_metadata*` on each of the other nodes. On these other nodes, the metadata file for the device should be renamed to `/var/opt/teradata/vormetric/vte-metadata-dir/vormetric/secvm_<device>_metadata_xforming`.

5. Guard the device on the designated node with the In-Place Data Transformation policy that specifies the new key you want to use for the device.

6. Wait for the data transformation process to complete, and then make sure that the metadata for the device has been updated on the other nodes in the clique and the renamed metadata files have been removed. Each node should have identical copies of `/var/opt/teradata/vormetric/vte-metadata-dir/vormetric/secvm_<device>_metadata` and `/var/opt/teradata/vormetric/vte-metadata-dir/vormetric/secvm_<device>_metadata_xforming` should *not* exist on any node.

7. Remove the IDT-Capable GuardPoint that you created earlier in this procedure on the designated node.

8. Guard the device through the client group to make sure that all nodes in the cluster recognize it as guarded.

Generating IDT-capable Metadata for Teradata Storage Expansion

Teradata Disk Expansion with IDT-Capable Devices

You can use the `voradmin td expand` command to configure new disks as IDT-capable devices during Teradata disk expansion.

> **Note**
>
> - Guard each new device using the same policy applied to an existing guarded IDT-Capable device.
> - Only use this option on new disks added during Teradata disk expansion.

After the new disk is visible to all members of the clique, perform the following steps to configure the new disk:

1. Configure the new disk using the following command on any node in the clique.

   ```
   voradmin td expand <newDiskName> <existingDiskName>
   ```

   > **Note**
   >
   > Do not run this command on the same device on multiple nodes.

2. Guard the new device on the CipherTrust Manager on all members of the clique.

For disk expansion that includes the addition of a new node to the clique, perform the following steps:

1. Install CTE on the new node and register the node to the CM, but **DO NOT** add the node to the client group, if using one.

2. On the new node, create the CTE Metadata directory, type:

```
mkdir -p /opt/teradata/vormetric/vte-metadata-dir
```

3. On one of the existing nodes in the clique, run the `voradmin td distribute` option to copy all existing metadata to the new node, type:

   ```
   voradmin td distribute < node1 name, node2 name, etc..> <device>
   ```

4. On the same node selected in step 3, use the `voradmin td expand` option to configure the new disk as an IDT-Capable device, type.

   ```
   voradmin td expand <newDiskName> <existingDiskName>
   ```

   > **Note**
   >
   > The `td expand` option must be run on each new disk.

5. On CipherTrust Manager, guard the new disk in the client group. You can verify that the new disk is now guarded on all nodes except the new node.

6. On the CipherTrust Manager, add the new node as a member to the client group. Once the new node is added as a member, CipherTrust Manager will push the security policy to the new node and guard all of the disks with metadata files on the new node.

Best Practices

- Using a of Host/Client Group to Guard Metadata Directories
- Using a Host/Client Group for Guarding Teradata Devices in a Clique
- Best Practice for Preparation for Initial Data Transformation or Rekey

Using a of Host/Client Group to Guard Metadata Directories

Thales recommends that customers use a host group exclusively for guarding and management of the CTE external metadata directories across all nodes in Teradata cluster. (For details about the CTE metadata directory, see Location of the CTE Private Region.)

By using a Host/Client Group, you can apply the same policy to the external metadata directory (`/opt/vte/vte-metadata-dir`) on all of the CTE protected nodes in the cluster.

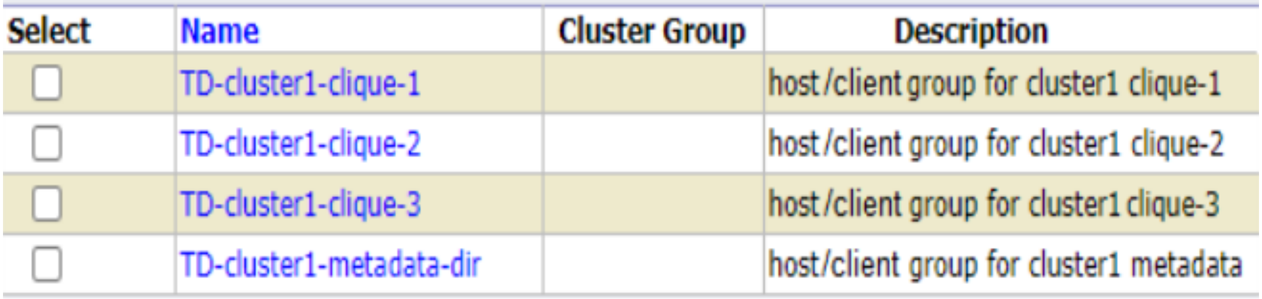
Using a Host/Client Group for Guarding Teradata Devices in a Clique

Thales recommend that customers use a Host/Client Group to guard and manage the devices shared by all CTE protected nodes in a clique. Each clique in the cluster should be associated with a separate Host/Client Group. When configuring the Host/Client Group, the members for that client group are the nodes that belong within a clique in the cluster.

Example 1: If you have a cluster consisting of 4 cliques where there are 2 nodes sharing disks in each clique, you would create 4 Host/Client Groups, one for each clique. In each Host/Client Group, you would add the 2 nodes that are associated with the clique.

Example 2: If you have a cluster consisting of 1 clique where there are 6 nodes sharing disks in the clique, you want to create 1 client group with all 6 members that belong within that clique.

Below is an example of the Host Groups for a cluster consisting of three cliques and one metadata group:



Select	Name	Cluster Group	Description
☐	TD-cluster1-clique-1		host/client group for cluster1 clique-1
☐	TD-cluster1-clique-2		host/client group for cluster1 clique-2
☐	TD-cluster1-clique-3		host/client group for cluster1 clique-3
☐	TD-cluster1-metadata-dir		host/client group for cluster1 metadata

Best Practice for Preparation for Initial Data Transformation or Rekey

With large amounts of data managed in Teradata clusters, duration of initial data transformation and/or subsequent rekey can be very long and challenging because the database must be stopped during the data transformation process. Although you cannot transform data without shutting down the database, you may be able to reduce the transformation time by distributing the transformation of the guarded devices across multiple nodes in your cluster.

Because multiple nodes in a cluster within a clique share access to the same devices, you can designate a subset of devices to each node within the clique for initial data transformation or subsequent rekey. This requires unguarding all of the IDT-Capable devices in the Host/Client Group associated with the clique, and re-enabling each IDT-Capable GuardPoint on the node designated for the data transformation of that IDT-Capable GuardPoint. The device then remains unguarded on other nodes until the data transformation completes.

After the data transformation completes, you can then enable the guarded devices at the Host/Client Group level. Enabling the guarded device at Host/Client Group level enables the guarded devices on all of the nodes within the clique to share access to the device.

After transforming data and guarding each device on all of the nodes, you can then restart the database.

As described in Interoperability with Host Groups, the metadata file for each device is replicated to the hosts in the cluster that share access to the device. As noted, replication of metadata files is automatic.

Uninstalling CTE from the Teradata Cluster

The following procedure describes how to remove CTE from your Teradata cluster. Note that uninstalling CTE will unprotect the pdisk devices that CTE is protecting/ encrypting. Because the data on the pdisks remain encrypted after uninstalling CTE, the entire clear-text data must be restored to the pdisks from backup. Therefore, you must have a full backup of your database to restore to your cluster before you uninstall CTE. See the Teradata Administration Guide or your Teradata Customer Support Representative for information about backing up your database in full.

> **Note**
>
> You must be familiar with CTE operations for protecting Teradata clusters and IDT-Capable GuardPoints.

To uninstall CTE:

1. Make sure you have access to a full database backup so that you can restore your Teradata database after uninstalling CTE.

2. Shut down your Teradata database.

3. Use the `secfsd -status guard` command to view the full list of device GuardPoints guarded by CTE on each node of your cluster. From this information, create a consolidated list of GuardPoints with one entry per device in the consolidated list. Note that the devices accessed by the nodes in the same clique are shared devices on each node. The consolidated list must include all devices that are protected in your cluster.

4. Perform the following steps on the CipherTrust Manager:

   a. Click on the client group name associated with your Teradata cluster and then click the **GuardPoints** tab.

   b. Select the device names listed in GuardPoints table. The devices listed in GuardPoints table are the same devices listed in the consolidated list of devices compiled in step 3.

   c. Disable the selected device GuardPoints on all nodes in the cluster.

   d. Wait for device status to change from green circle to red circle in the Status column for all the selected devices.

   e. Select the same devices.

   f. Click **Unguard** to remove the selected device GuardPoints on all nodes in the cluster.

5. Perform the following steps on one of the nodes of every clique in your cluster:

   a. Run the `voradmin idt delete` command to delete the IDT-Capable GuardPoint configuration recorded for each device. This command automatically deletes the GuardPoint configuration across the nodes in the same clique.

   ```
   voradmin idt delete <device name>
   ```

   b. Repeat the previous step on the other devices in your cluster, if any.

6. Go back to the Key Manager and do the following:

   a. Select the Metadata Directory GuardPoint in the GuardPoints table of the same host group.

b. Click **Disable** to disable the Metadata GuardPoint directory on every node in the cluster.

c. Select the Metadata Directory GuardPoint in the GuardPoints table again and then click **Unguard** to remove the Metadata Directory GuardPoint on all the nodes in the cluster.

7. On every node in the cluster:

   a. Run the following command to stop CTE service on each node:

   ```
   /etc/vormetric/secfs stop
   ```

   b. Run the following command to uninstall the CTE Agent on each node:

   ```
   /opt/teradata/vormetric/agent/vmd/bin/uninstall
   ```

To restore a full backup of the database to your cluster, contact your Teradata Customer Support Representative (CSR) to perform System Initializer and do a full restore.

Alerts and Errors

This section lists the alerts and errors that may be encountered during system operations on Teradata Appliance. Refer to Alerts and Errors listed in the chapter on IDT-Capable Device GuardPoints for additional Alerts and Errors.

General Errors

The errors in this section indicate what step failed during CTE system operations for Teradata clusters and will always be paired with an operation error which indicates which CTE system operation failed. For example:

- IDT-TD-ALERT: Node did not respond to pcl command
- IDT-TD-ALERT: Failed to complete rekey on remote nodes

This pairing of errors indicates that the steps after data transformation were unsuccessful and that the `voradmin` command `voradmin td rekeyed <device name>` is the one that failed.

For details about the `voradmin td rekeyed` command, see the `voradmin` manpage.

IDT-TD-ALERT: Node <node name> did not respond to pcl command

The node listed did not respond to a pcl command to perform an operation.

Solution: Check if the node is still online and responding. If it is, run the `voradmin` command to repeat the failed operation. This error message will always be accompanied by one of the other IDT-TD-ALERTS which will determine which `voradmin` command to run to correct the error.

IDT-TD-ALERT: Node <node name> failed to perform voradmin task

The node listed failed to perform a `voradmin` command invoked through `pcl`.

Solution: Correct the issue blocking `voradmin` and run the `voradmin` command to repeat the failed operation. This error message will always be accompanied by one of the other IDT-TD-ALERTS which will determine which voradmin command to run to correct the error.

IDT-TD-ALERT: Failed to find clique for disk <device>

CTE failed to locate the clique for the device specified.

Solution: Verify that `/usr/lib/tmpfiles.d/pdisk.conf` is properly configured on all nodes in the Teradata cluster.

IDT-TD-ALERT: Failed to move or rename IDT-Capable metadata file on remote nodes

This error can be due to two issues; one, metadata files that have been distributed to the remote nodes cannot be moved into the metadata directory, or two, metadata files on remote nodes cannot be renamed in preparation for rekey.

Solution: For the first issue, rerun `voradmin td distribute <node name> <device name>` command once any other problems have been resolved. For the second issue rerun `voradmin td rekey <device name>` to prepare for rekey again once any other problems have been resolved.

For details about the `voradmin td` commands, see the `voradmin` manpage.

IDT-TD-ALERT: Failed to get GuardPoint status on remote nodes

The GuardPoint for the device is still present on the CipherTrust Manager for remote nodes or CTE failed to ascertain the GuardPoint status for remote nodes.

Solution: Verify that the GuardPoint for the device has been removed from the CipherTrust Manager for remote nodes and then rerun the `voradmin` command. This error message will always be accompanied by one of the other IDT-TD-ALERTS which will determine which `voradmin` command to run to correct the error.

Operation Errors

These errors indicate which CTE system operation failed and what command should be run upon resolution of the issue that caused the error.

IDT-TD-ALERT: Failed to distribute IDT-Capable metadata file to remote nodes

CTE failed to copy the IDT-Capable metadata file from the current node to all nodes in the clique.

Solution: Correct any issues blocking `pcl` or `voradmin` and then run `voradmin td distribute <node name> <device name>` to attempt copying the metadata file over to the remote node.

For details about the `voradmin td distribute` command, see the `voradmin` manpage.

IDT-TD-WARNING: Failed to delete IDT-Capable metadata file on remote nodes

CTE was unable to remove the IDT-Capable metadata file for a device from a remote node.

Solution: Run `voradmin idt delete <device name>` on the remote node after correcting issues impeding voradmin. Members of the clique that have successfully removed the metadata file will not be adversely affected if an attempt to remove an already de-configured device occurs.

IDT-TD-ALERT: Failed to complete rekey on remote nodes

CTE was unable to complete the rekey operation on remotes nodes to make the device available for guarding again.

Solution: Correct the issue and run `voradmin td rekeyed <node name> <device name>` to signal the remote node that rekey is complete and provide an updated copy of the metadata file to that remote node.

Integrating CTE with an Exasol Database

This document describes how to integrate CTE with an Exasol database.

Test Environment

- CTE Agent: 7.2.0.128
- CipherTrust Manager: 2.8.0
- OS: RHEL/CentOS 7.9
- Exasol DB: 7.1.12
- File System: LVM (raw device)

Steps

To integrate CTE with an Exasol database:

1. Install and Register the CTE Agent
2. Configure the Client Settings
3. Back up the Exasol Database
4. Delete the Exasol Database
5. Create a GuardPoint
6. Update the Secvm Device
7. Restore the Exasol Database

8. Validations

Install and Register the CTE Agent

1. Install the CTE Agent on the client machine where the Exasol database is configured.

2. Register the CTE Agent with the CipherTrust Manager.

Refer to CTE - Agent Quick Start Guide for details.

Configure the Client Settings

1. Log on to the CipherTrust Manager.

2. Open the **Transparent Encryption** application.

3. Go to **Clients** > **Client** and select the **Client**. This is the client machine where the Exasol database is configured.

4. Click **Client Settings**.

5. In the **Settings** field, add the `hddident` and `cos_storage` processes as authenticators.

   ```
   |authenticator|/7.1.12/image/usr/opt/EXASuite-7/EXAClusterOS-7.1.12/sbin/hddident
   |authenticator|/7.1.12/image/usr/opt/EXASuite-7/EXAClusterOS-7.1.12/libexec/cos_storage
   ```

6. Click **Apply**.

Back up the Exasol Database

Now, on the CTE client (like PuTTY or MobaXterm), back up the Exasol database. To do so:

1. Log on to an SSH client.

2. Run the following commands:

   ```
   cd ExasolSetup/
   ssh -i id_rsa -p 2233 localhost
   ```

```
dwad_client storage-backup PROD_A
dwad_client pdd-proc PROD_A
dwad_client stop-wait PROD_A
```

Delete the Exasol Database

After backing up, delete the Exasol database that includes the EXAStorage volumes and disks, and then stop the Exasol service. On an SSH client, run the following commands:

```
 dwad_client stop-wait PROD_A i.e, dwad_client stop <DB_NAME>
dwad_client del PROD_A
csinfo -v
csvol -d -v 0
csinfo -H
cshdd -r -h /dev/exa1/lvol0 -n 11
logout
systemctl stop exasol
```

Create a GuardPoint

Create the required GuardPoint on the CTE client from the CipherTrust Manager GUI. While creating the GuardPoint:

- Select the Type of the device as **Auto Raw or Block Device**.
- Enter the **Path** of the logical volume, for example, `/dev/exa1/lvol0`.
- Select the Policy Type as **Standard**.
- Create a User Set named `root` and grant it the permission to perform all Actions and Effects.

Refer to Creating GuardPointsfor details.

Update the Secvm Device

On the SSH client:

1. Run the following command:

   ```
   secfsd -status devmap
   ```

   The output similar to the following is displayed:

```
Secvm Device                  Native Device
------------                  -------------


/dev/secvm/dev/exa1/lvol0     /dev/exa1/lvol0
```

In the above sample output, `/dev/secvm/dev/exa1/lvol0` is the Secvm device.

2. Copy the Secvm device.

3. Delete the metadata from the `/7.1.12/exa/metadata` folder.

```
cd /7.1.12/exa/metadata/
rm -rf *
```

> **Note**
>
> It is recommended to delete the metadata before updating the Secvm device.

4. Update the Secvm device in the `EXAConf` file.

a. Navigate to `/7.1.12/exa/etc/`.

b. Open the `EXAConf` file in a text editor.

c. Search for `Disk`, and update the **Devices** entry with the copied Secvm device.

```
[Node : 11]
    PrivateNet = 10.164.13.247/24
    PublicNet =
    Name = n11
    Affinity = 11
    UUID= 159D338FDBOBA36BACF2CC104224C683D33726E8
    DockerVolume = n11
    ExposedPorts = 8563:8574, 2580:2591
    [[Disk disk1]]
        Devices = /dev/secvm/dev/exa1/lvole
```

d. Search for `Checksum` and set the value of Checksum as **COMMIT**.

   e. Save the changes and close the file.

5. Start the Exasol service by running `systemctl start exasol`.

6. Verify the status of the Exasol service by running `systemctl status exasol`. The status must be `active (running)`.

Restore the Exasol Database

On the CTE client, restore the Exasol database by running the following commands:

```
dwad_client pdd-restore PROD_A
dwad_client storage-restore PROD_A
dwad_client pdd-proc PROD_A
```

Validations

1. Start a new session on the SSH client.

2. Encrypt the data on the initial device by running `strings /dev/exa1/lvol0|more`.

```
f~N#
pU_t
H9rg,1
UcEa
\>,u
B ~ A
9DQ&
YsE#
kel4X
=:69m
hdHA
k\$<
|HUI|
rpt7
`YI&?
* 8!
g#,—Bzw
_w_0aV
y7zk
```

```
B#+9
--More--
```

3. Encrypt the data on the guarded device by running `strings /dev/secvm/dev/exa1/lvol0|more` for the following:

   - Policy with `all_ops` action, and `Permit` and `ApplyKey` effects.

     ```
      ID               =159D338FDBOBA36BACF2CC104224C683D33726E8
     NAME              =/dev/secvm/dev/exa1/lvolo
     STORAGE           =CSR_4096_7855104_2
     FORMATED          =YES
     CHECKSUM          =bcc800bd0f72aee46daf4f8cdf8fef4e

     persistent (256K)
     ```

   - Policy with `all_ops` action and `Permit` effect.

     ```
      cL]b
     uzx=3
     -B=IQ_
     1vaT#
     _"a%
     YkFL
     3So@
     g@Cp
     !HtIZ}I
     RA3N2V
     xC~oP(
     f~N#
     pU_t
     H9rg,1
     UcEa
     \>,u
     B ~ A
     9DQ&
     YsE#
     kel4X
     =:69m
     ```

```
hdHA
k\$<
|HUI|
rpt7
`YI&?
* 8!
g#,—Bzw
_w_0aV
y7zk
B#+9
--More--
```

- Policy with `all_ops` action and `Deny` effect.

```
 [root@NOIENC1PFL-IR22 ~]# strings /dev/secvm/dev/
exa1/1v010[more
strings: /dev/secvm/dev/exa1/1v010: Permission denied
[r00t@NOIENC1PFL-IR22 ~]# I
```

Refer to Security Rules for more information on actions and effects.

4. Encrypt the data on the original hard disk (excluding the disk headers) by running `strings /dev/sdc|more`.

```
 creation_time = 1661854578
creation_host = "NOIENC1PFL-IR22" segment_count = 1
segment1 {
start_extent = 0
extent_count = 7680
type="striped"
stripe_count = 1
stripes "pv0", 0
[
# Generated by LVM2 version 2.02.187(2)-RHEL7 (2020-03-24): Tue
Aug 30 15:46:18 2022
contents = "Text Format Volume Group"
version = 1
description=""
creation_host = "NOIENC1PFL-IR22"
```

```
creation_time = 1661854578
CLJb
uzx=3
-B=10
1wRvT#
^a%
YkFL
3So@
g@cp
;NzH <01
((6k
@x/1g
BaP>'>
tn/q
d)W2
ar1;
```

Note

- LDT is not tested as LDT does not support raw devices.
- Initial data transformation is not required as the data is deleted during the Exasol installation.

Integrating CTE with a Couchbase Database

This document describes how to integrate CTE with a Couchbase database.

Test Environment

- CTE Agent: 7.2.0.128
- CipherTrust Manager: 2.8.0
- OS: RHEL/CentOS 8.2
- Couchbase: 7.1

- File System: XFS

System Requirement Specifications

- RAM: 16 GB
- Storage: 32 GB
- CPU: 3 GHz

Steps

To integrate CTE with a Couchbase database:

1. Create a Couchbase Cluster
2. Install and Register the CTE Agent
3. Create the GuardPoints
4. Create a Bucket and Run N1QL Query

Create a Couchbase Cluster

Create a Couchbase cluster of one or more nodes. You will install Couchbase Server and CTE Agent on these nodes.

Perform the following steps on all the nodes:

1. Log on to the SSH client.

2. Install the Couchbase Server.

   ```
   dnf install -y https://packages.couchbase.com/releases/couchbase-release/couchbase-release-1.0-x86_64.rpm
   dnf install -y couchbase-server
   ```

   Output:

   ```
   [root@NOIENC1PFL—IR30 ~]# rpm -qa | grep couch
   couchbase-server-7.1.1-3175.x86_64
   couchbase-release-1.0-11.x86_64
   ```

3. Ensure that the `couchbase-server` service is **running**.

```
service couchbase-server status
```

Output:

```
[root@NOIENC1PFL—IR3O ~]# service couchbase—server status
Redirecting to [bin/systemctl status couchbase-server.service
o couchbase—server.service — Couchbase Server
Loaded: loaded (/usr/lib/systemd/system/couchbase—server.service;
enabled; vendor preset: disabled)
Active: active (running) since Thu 2022-07-28 10:28:39 IST; 1
months 15 days ago
Docs: httgs:[zdocs.couchbase.com
Main PID: 2577 (beam.smp)
Tasks: 424 (limit: 23815)
Memory: 1.36
CGroup: /system.slice/couchbase—server.service
        |-2577 /opt/couchbase/lib/erlang/erts—12.1.5/bin/beam.smp
-A 16 -sbwt none -- -root /opt/couchbase/lib/erlan
        |-2600 /opt/couchbase/lib/erlang/erts-12.1.5/bin/epmd -
daemon
        |-2681 erl_chi1d_setup 200000
        |-2701 /opt/couchbase/bin/gosecrets
        |-2705 /opt/couchbase/lib/erlang/erts-12.1.5/bin/beam.smp
-A 16 -sbt u -P 327680 -K true -swt low -sbwt none
        |-2727 er1_chi1d_setup 200000
        |-2746 sh -s disksup
        |-2748 /opt/couchbase/lib/erlang/lib/os_mon—2.7.1/priv/
bin/memsup
        |-2749 /opt/couchbase/lib/erlang/lib/os_mon—2.7.1/priv/
bin/cpu_sup
        |-2750 portsigar for ns_1@cb.local 2577
```

4. Initialize the node.

```
/opt/couchbase/bin/couchbase-cli node-init -c <Node_IP> -u <USERN
AME> -p <PASSWORD> --node-init-data-path /opt/couchbase/var/lib/
couchbase/data --node-init-index-path /opt/couchbase/var/lib/
couchbase/data --node-init-eventing-path /opt/couchbase/var/lib/
```

```
couchbase/data --node-init-analytics-path /opt/couchbase/var/lib/
couchbase/data --ipv4
```

Output:

```
    [root@NOIENC1PFL-IR30 ~]# /opt/couchbase/bin/couchbase-cli
node-init -c 10.164.11.191 -u placeholdername -p placeholderpwd --
node-init-data-path /opt/couchbase/var/lib/couchbase/data --node-
init-index-path /opt/couchbase/var/Iib/couchbase/data --node-init-
eventing-path /opt/couchbase/var/lib/couchbase/data --node-init-
analytics-path /opt/couchbase/var/lib/couchbase/data --ipv4
    SUCCESS: Node initialized
    [root@NOIENC1PFL-IR30 ~]#
```

After initializing all the nodes to be part of the cluster:

1. Create the cluster on one of the nodes.

   ```
   /opt/couchbase/bin/couchbase-cli cluster-init -c 10.164.11.191 --
   cluster-username <CLUSTER_USERNAME> --cluster-password <CLUSTER_P
   ASSWORD> --services data,index,query --cluster-ramsize 512 --
   cluster-index-ramsize 256
   ```

   Output:

   ```
    [root@NOIENC1PFL-IR30 ~]# /opt/couchbase/bin/couchbase-cli
   cluster-init -c 10.164.11.191 --cluster-username admin --cluster-
   password pvlinux --services data,index,query --cluster-ramsize 512
   --cluster-index-ramsize 256
   SUCCESS: Cluster initialized
   [root@NOIENC1PFL-IR30 ~]#
   ```

   > **Note**
   >
   > The `ramsize` of the data, index, and other files can be changed according to the system availability.

2. Add the remaining nodes to the cluster.

```
/opt/couchbase/bin/couchbase-cli server-add -c 10.164.11.191:8091
--username <CLUSTER_USERNAME> --password <CLUSTER_PASSWORD> --
server-add 10.164.14.158 --server-add-username someName --server-
add-password somePassword --services data
```

Note

Disable and stop the firewall on all the nodes.

Output:

```
[root@NOIENC1PFL-IR30 ~]# /opt/couchbase/bin/couchbase-cli server-
add -c 10.164.11.191:8091 --username admin --password pvlinux --
server-add 10.164.14.158 --server-add-username someName --server-
add-password somePassword --services data
SUCCESS: Server added
[root@NOIENC1PFL-IR30 ~]#
```

3. Rebalance all the nodes from any node.

```
/opt/couchbase/bin/couchbase-cli rebalance -c 10.164.11.191:8091 -
-username <CLUSTER_USERNAME> --password <CLUSTER_PASSWORD>
```

Output:

```
[root@NOIENC1PFL-IR30 ~]# /opt/couchbase/bin/couchbase-cli
rebalance -c 10.164.11.191:8091 --username admin --password
pvlinux
SUCCESS: Rebalance complete
[root@NOIENC1PFL-IR30 ~]#
```

Note

Couchbase recommends to rebalance the nodes when they are added or removed, and on failover.

4. Ensure that all the nodes are in a healthy state.

```
/opt/couchbase/bin/couchbase-cli server-list -c localhost:8091 --
username <CLUSTER_USERNAME> --password <CLUSTER_PASSWORD>
```

Output:

```
[root@NOIENC1PFL-IR30 ~]# /opt/couchbase/bin/couchbase-cli server-
list -c localhost:8091 --username admin --password pvlinux
ns_1@10.164.11.191 10.164.11.191:8091 healthy active
ns_1@10.164.11.203 10.164.11.203:8091 healthy active
ns_1@10.164.14.158 10.164.14.158:8091 healthy active
```

The sample output above shows that all the nodes are `healthy`.

Install and Register the CTE Agent

1. Install the CTE Agent on all nodes of the Couchbase cluster.

2. Register the CTE Agent with the CipherTrust Manager.

Refer to CTE - Agent Quick Start Guide for details.

Create the GuardPoints

Perform the following steps on all the cluster nodes.

1. On the CTE client, stop the Couchbase service.

   ```
   service couchbase-server stop
   ```

2. On the CipherTrust Manager, create the GuardPoint. While creating the GuardPoint:

   - Enter the **Path** of the important Couchbase database, for example, `/opt/couchbase/var/lib/couchbase/data/`.

   - Select the Policy Type as **Standard**.

Note

- If the Couchbase buckets are already created, Dataxform needs to be performed.
- You can also create **LDT** policy.

- **Create a User Set with users `root` and `couchbase`, and give them the permission to perform all Actions and Effects.**

Refer to Creating GuardPointsfor details.

3. Ensure that the GuardPoint status is `guarded` on the CTE client.

```
secfsd -status guard
```

Output:

```
[root@N01ENC1PFL-IR30 data]# secfsd -status guard

GuardPoint
Policy                  Type        ConfigState     Status
Reason
----------
------                  ----        -----------     ------
------
/opt/couchbase/var/liblcouchbase/data
production_policy       local       guarded         guarded     N/
A
[root@N01ENC1PFL-IR30 data]#
```

Create a Bucket and Run N1QL Query

1. Log on to the Couchbase Web Console.
2. Create a new bucket. Refer to the Couchbase documentation for details. Alternatively, you can use an existing sample bucket.
3. On any CTE client, check for the newly created bucket in the bucket list.

```
/opt/couchbase/bin/couchbase-cli bucket-list -c localhost:8091 --username admin --password pvlinux
```

Output:

```
[root@NOIENC1PFL-IR30 ~]# /opt/couchbase/bin/couchbase-cli bucket-list -c localhost:8091 --username admin --password pvlinux
travel-sample
bucketType: membase
numReplicas: 1
ramQuota: 629145600
ramUsed: 55536304
[root@NOIENC1PFL-IR30 ~]#
```

4. Perform an operation on the bucket by running an N1QL query.

```
/opt/couchbase/bin/cbq -u=admin
select name from `travel-sample` WHERE type="airline" LIMIT 1;
```

Output:

```
[root@NOIENC1PFL-IR30 ~]# /opt/couchbase/bin/cbq -u=admin
 Enter Password:
 Connected to : http://localhost:8091/. Type Ctrl-D or \QUIT to exit.

 Path to history file for the shell : /root/.cbq_history
cbq> select name from `travel-sample` WHERE type="airline"  LIMIT 1;
{
    "requestID": "54f7edb8-4511-49f2-bead-8393e62d60d3",
    "signature": {
        "name": llj0.sonll
    },
    "results": [
    {
        "name": "40-Mile Air"
    }
    ],
    "status": "success",
    "metrics": {
    "elapsedTime": "27.916695ms",
    "executionTime": "27.827434ms",
    "resultCount": 1,
    "resultSize": 37,
    "serviceLoad": 12
    }
}
cbq>
```

Integrating CTE with an EnterpriseDB Postgres Advanced Server

This document describes how to integrate CTE with an EnterpriseDB (EDB) Postgres Advanced Server.

Test Environment

- CTE Agent: 7.2.0.128
- CipherTrust Manager: 2.9.0
- OS: RHEL 7.9
- EDB Postgres Advanced Server: 14.5
- File System: XFS

Steps

To integrate CTE with an EnterpriseDB Postgres Advanced Server:

1. Install the Prerequisite Packages

2. Install and Configure the EDB Postgres Advanced Server

3. Install and Register the CTE Agent

4. Integrate EDB with CTE

5. Create the GuardPoints

Install the Prerequisite Packages

Install the prerequisite packages by performing the following steps on the client machine.

1. Log on to the SSH client.

2. Run this command.

   ```
   yum -y install   https://dl.fedoraproject.org/pub/epel/epel-
   release-latest-7.noarch.rpm
   ```

Now you can install the EDB Postgres Advanced Server.

Install and Configure the EDB Postgres Advanced Server

Before starting the installation and configuration, ensure that the client machine is already registered with the RHEL. Perform the following steps.

1. Log on to the EDB.

2. Go to the EDB Profile section and copy the EDB repository credentials.

3. On the SSH client, install the repository configuration package.

   ```
   yum -y install https://yum.enterprisedb.com/edbrepos/edb-repo-
   latest.noarch.rpm
   ```

4. Update the credentials in `/etc/yum.repos.d/edb.repo` file with the copied credentials.

   ```
   sed -i "s@<username>:<password>@john:Kh0kVuJg7Wg60Gn1@" /etc/
   yum.repos.d/edb.repo
   ```

5. Install the EDB Postgres Advanced Server.

```
yum -y install edb-as14-server
```

Output:

```
Installed:
edb-as14-server.x86_64 0:14.5.0-1.rhel7

Dependency Installed:
edb-as14-pgagent.x86_64 0:4.2.2-1.rhel7
edb-as14-server-client.x86_64 0:14.5.0-1.rhel7
edb-as14-server-cloneschema.x86_64 0:1.16-1.rhel7
edb-as14-server-contrib.x86_64 0:14.5.0-1.rhel7
edb-as14-server-core.x86_64 0:14.5.0-1.rhel7
edb-as14-server-devel.x86_64 0:14.5.0-1.rhel7
edb-as14-server-docs.x86_64 0:14.5.0-1.rhel7
edb-as14-server-indexadvisor.x86_64 0:14.5.0-1.rhel7
edb-as14-server-libs.x86_64 0:14.5.0-1.rhel7
edb-as14-server-llvmjit.x86_64 0:14.5.0-1.rhel7
edb-as14-server-parallel-clone.x86_64 0:1.9-1.rhel7
edb-as14-server-pldebugger.x86_64 0:1.1-1.rhel7
edb-as14-server-plperl.x86_64 0:14.5.0-1.rhel7
edb-as14-server-plpython3.x86_64 0:14.5.0-1.rhel7
edb-as14-server-pltcl.x86_64 0:14.5.0-1.rhel7
edb-as14-server-sqlprofiler.x86_64 0:4.0-1.rhel7
edb-as14-server-sqlprotect.x86_64 0:14.5.0-1.rhel7
edb-as14-server-sslutils.x86_64 0:1.3-1.rhel7
Complete!
```

Note

An **enterprisedb** user will be created.

6. Initialize the database cluster.

```
PGSETUP_INITDB_OPTIONS="-E UTF-8" /usr/edb/as14/bin/edb-as-14-
setup initdb
```

7. Start the database server.

```
systemctl start edb-as-14
```

8. Connect to the database server.

9. As the **enterprisedb** user, open a *psql* session.

```
sudo su - enterprisedb
/usr/edb/as14/bin/psql -d edb -p 5444
```

> **Note**
>
> Now, you can create a database with tables and enter data in the tables, using the simple SQL commands.

10. Ensure that the Database server is running.

```
/usr/edb/as14/bin/pg_ctl status -D /var/lib/edb/as14/data
```

Output:

```
-bash-4.2$ /usr/edb/as14/bin/pg_ctl status -D /var/lib/edb/as14/
data
pg_ctl: server is running (PID: 43002)
/usr/edb/as14/bin/edb-postgres "-D" "/var/lib/edb/as14/data"
```

Install and Register the CTE Agent

1. Install the CTE Agent on the client machine where the EDB Postgres Advanced Server is installed and configured.

2. Register the CTE Agent with the CipherTrust Manager.

Refer to CTE - Agent Quick Start Guide for details.

Integrate EDB with CTE

On the SSH client, perform the following steps.

1. As a super (root) user, stop the EDB server.

   ```
   systemctl stop edb-as-14
   ```

2. Update the `edb-as-14.service` file.

   a. Navigate to `/lib/systemd/system/`.

   b. Open `edb-as-14.service` in a text editor.

   c. Below the **Unit** section, add the following line.

   ```
   Requires=secfs-fs-barrier.service
   ```

   d. Save the change and close the file.

3. Update the `secfs-fs-barrier.service` file.

   a. Navigate to `/lib/systemd/system/`.

   b. Open `secfs-fs-barrier.service` in a text editor.

   c. At the end of the **Before** section, add the following line.

   ```
   edb-as-14.service
   ```

   d. Save the change and close the file.

4. Reboot the system.

5. Stop the EDB server.

6. Restart the SecFS.

7. Start the EDB server again.

The setup is ready, now you can guard the EDB data and log directories.

Create the GuardPoints

Perform the following steps on the CTE client.

1. On the CTE client, stop the EDB server.

2. On the CipherTrust Manager, create the GuardPoints. While creating the GuardPoint:

   - Enter the **Path** of the data and log directories that are `/var/lib/edb/as14/data` and `/var/log/edb/as14`.

   - Select the Policy Type, you can select it either as **Standard** or **Live Data Transformation** (LDT).

   - Create a User Set with users `root` and `enterprisedb`, and give them the permission to perform all Actions and Effects.

   Refer to Creating GuardPointsfor details.

3. Once the policy is enabled, start the EDB server.

Integrating CTE with a MongoDB database

This document describes how to integrate CTE with a MongoDB database.

Test Environment

- CTE Agent: 7.2.0 and 7.3.0
- CipherTrust Manager: 2.8.0, 2.9.0, and 2.10.0
- OS: RHEL 7.9, Ubuntu 20.04, and Ubuntu 18
- MongoDB version: 3.6.8
- File System: XFS and EXT4, and NFS

Steps

To integrate CTE with a MongoDB database, install and register the CTE Agent and create appropriate GuardPoints.

Install and Register the CTE Agent

1. Install the CTE Agent on the client machine where the MongoDB database is installed and configured.
2. Register the CTE Agent with the CipherTrust Manager.

Refer to CTE - Agent Quick Start Guide for details.

Create the GuardPoints

Perform the following steps on the CTE client.

1. On the CTE client, stop the MongoDB database.
2. On the CipherTrust Manager, create a GuardPoint. While creating the GuardPoint:
   - Enter the **Path**, `/var/lib/mongo/`.
   - Select the Policy Type. You can select **Standard** or **Live Data Transformation** (LDT).
   - Create a User Set with users `mongod` and `usr/bin/mongod`, and give them the permission to perform all Actions and Effects. The default access is No Access.

   Refer to Creating GuardPointsfor details.
3. Once the policy is enabled, start the MongoDB database.

Integrating CTE with an Apache Cassandra database

This document describes how to integrate CTE with an Apache Cassandra database.

Test Environment

- CTE Agent: 7.2.0 and 7.3.0
- CipherTrust Manager: 2.8.0, 2.9.0, and 2.10.0
- OS: Ubuntu 20.04 and Ubuntu 18
- Apache Cassandra version: 3.6.8
- File System: XFS and EXT4

Steps

To integrate CTE with an Apache Cassandra database, install and register the CTE Agent and create appropriate GuardPoints.

Install and Register the CTE Agent

1. Install the CTE Agent on the client machine where the Apache Cassandra database is installed and configured.

2. Register the CTE Agent with the CipherTrust Manager.

Refer to CTE - Agent Quick Start Guide for details.

Create the GuardPoints

Perform the following steps on the CTE client.

1. On the CTE client, stop the Apache Cassandra database.

2. On the CipherTrust Manager, create a GuardPoint. While creating the GuardPoint:

    - Enter the **Path**, `/var/lib/33buntu33ra/data`.

    - Select the Policy Type. You can select **Standard** or **Live Data Transformation** (LDT).

    > **Note**
    >
    > Set the Key Selection Rules of the LDT policy as **AES 256**.

    - Create a User Set with user `cassandb`, and give user the permission to perform all Actions and Effects. The default access is No Access.

    Refer to Creating GuardPointsfor details.

3. Once the policy is enabled, start the Apache Cassandra database.

Integrating CTE with a Neo4j Database

This document describes how to integrate CTE with a Neo4j Database.

Test Environment

- CTE Agent: 7.3.0 and 7.4.0
- CipherTrust Manager: 2.10.0
- OS: Ubuntu 20.04
- Neo4j version: 4.1.12
- File System: XFS and EXT4

Steps

To integrate CTE with a Neo4j database, install and register the CTE Agent, configure the Client Settings, and create appropriate GuardPoints.

Install and Register the CTE Agent

1. Install the CTE Agent on the client machine where the Neo4j Database is installed and configured.
2. Register the CTE Agent with the CipherTrust Manager.

Refer to CTE - Agent Quick Start Guide for details.

Configure the Client Settings

Add the following paths to the client settings:

Privilege	Path to Binary
authenticator_euid	/usr/bin/neo4j
authenticator_euid	/usr/bin/java
authenticator_euid	/usr/share/neo4j/bin
authenticator_euid	/usr/bin/ls
authenticator_euid	/usr/bin/bash

Refere to Client Settingsfor details.

Create the GuardPoints

Perform the following steps on the CTE client:

1. On the CTE client, stop the Neo4j Database.

2. On the CipherTrust Manager, create a GuardPoint. While creating the GuardPoint:

   - Enter the **Paths** `/var/log/neo4j`, `/etc/neo4j`, `/usr/share/neo4j/bin`, `/var/lib/neo4j/data`, `/var/lib/neo4j/certificates`, `/var/lib/neo4j/plugins`, and `/var/lib/neo4j/import`.
   - Select the Policy Type as **Standard**.
   - Create a User Set with users `neo4j` and `root`, and give them the permissions to perform all Actions and Effects. The default access is No Access.

   Refer to Creating GuardPointsfor details.

3. Once the policy is enabled, start the Neo4j Database.

Support Contacts

If you encounter a problem while installing, registering, or operating the product, please refer to the documentation before contacting support. If you cannot resolve the issue, contact your supplier or Thales Customer Support.

Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at Thales Customer Support, is where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable database of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

Tip

You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the REGISTER link.

Telephone Support

If you have an urgent problem, or cannot access the Customer Support Portal, you can contact Thales Customer Support by telephone at +1 410-931-7520. Additional local telephone support numbers are listed on the support portal.

Email Support

You can also contact technical support by email at technical.support@Thales.com.