Cohesity and Entrust KeyControl

Integration Guide

2026-04-15

© Entrust Corporation. All rights reserved.

Table of Contents

1. Introduction . . . 1
    1.1. Documents to read first . . . 2
    1.2. Requirements . . . 2
    1.3. High-availability considerations . . . 3
    1.4. Product configuration . . . 3
2. Procedures . . . 4
    2.1. Install and configure Entrust KeyControl . . . 4
    2.2. Deploy Cohesity DataProtect . . . 4
    2.3. Create Cohesity client certificates in KeyControl . . . 4
    2.4. Configure Cohesity for encryption with an external Key Management System . . . 5
    2.5. Create a Cohesity storage domain that uses KeyControl for encryption . . . 11
    2.6. Check KeyControl for Cohesity keys . . . 13
    2.7. Integrating with an Entrust HSM . . . 13
3. Cohesity DataPlatform CLI . . . 14
    3.1. Log in to the Cohesity server . . . 14
    3.2. Create a KMIP KMS . . . 14
    3.3. List current KMS settings . . . 14
    3.4. Modify Cohesity DataPlatform KMS settings . . . 15
4. Troubleshooting . . . 16
    4.1. KMS validation error with KMS configuration . . . 16
    4.2. KMS unreachable error during storage domain creation . . . 17
5. Additional resources and related products . . . 18
    5.1. nShield HSMs . . . 18
    5.2. nShield as a Service . . . 18
    5.3. KeyControl . . . 18
    5.4. Entrust products . . . 18
    5.5. nShield product documentation . . . 18

Chapter 1. Introduction

Entrust KeyControl has been rebranded as the Entrust Cryptographic Security Platform (CSP) Key Manager.

The Entrust CSP Key Manager continues to provide a comprehensive solution for discovering and managing the lifecycles of cryptographic keys, secrets, certificates, tokens, libraries, protocols, and configurations:

- The KeyControl Compliance Manager is now the Entrust CSP Compliance Manager. It still integrates with Entrust nShield Hardware Security Modules (HSMs) to protect the master keys for the CSP.
- KeyControl Vault is now the Entrust Cryptographic Security Platform Vault. The Cryptographic Security Platform Vaults also still integrate with Entrust nShield HSMs to provide an optional HSM root of trust.

Because the Entrust integrations are tested against specific product versions, this guide is still branded as a "KeyControl" integration. It was tested against a pre-CSP version of KeyControl.

Exercise caution when using an Entrust Integration Guide with a product version that does not match the tested version, because your version might not function in exactly the same way.

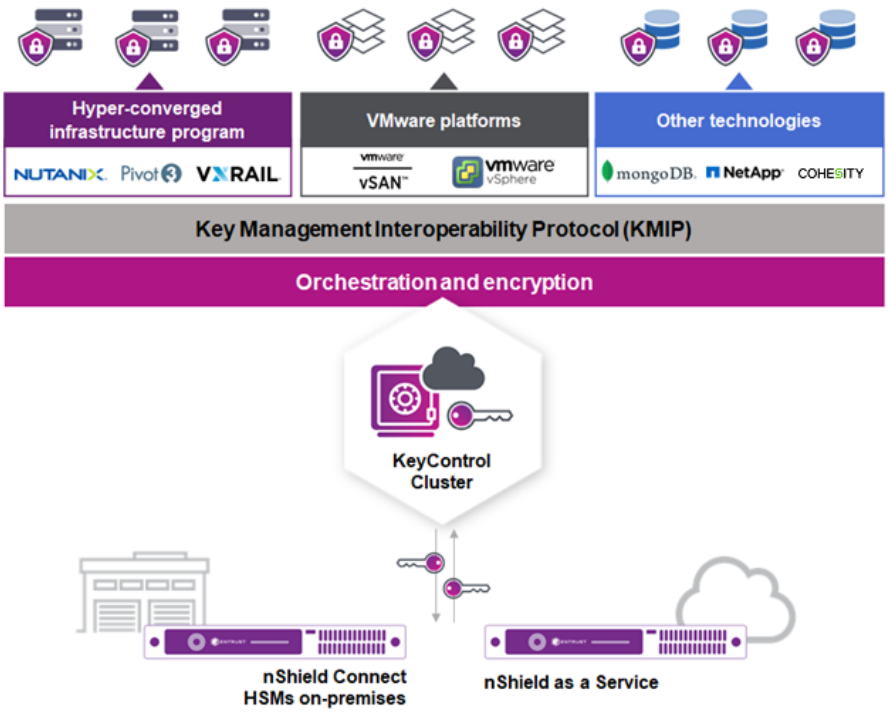
Entrust cannot guarantee the success of integrations in configurations other than those indicated in the guide. This guide remains on the website for customers using pre-CSP versions of KeyControl.

This document describes the integration of a Cohesity DataPlatform with the Entrust KeyControl Key Management Solution (KMS). Entrust KeyControl can serve as a KMS to a Cohesity cluster using the open standard Key Management Interoperability Protocol (KMIP).

Mutual authentication of each entity is performed using X.509 certificates over a Transport Layer Security (TLS) secure channel.

After deploying and configuring Entrust KeyControl, a KMS certificate is automatically generated and signed by the internal Certificate Authority (CA). The internal CA generates the X.509 client certificate that is uploaded to the Cohesity cluster for authentication.

If your organization mandates all certificates to be signed by a specific CA, KeyControl can use your organization's CA to sign its certificate.



Once configured, the Cohesity cluster will request a Key Encryption Key (KEK) from KeyControl for the entire cluster. This KEK securely wraps (encrypt/decrypt) the Data Encryption Keys (DEKs) created and stored locally in the Cohesity cluster. The DEKs are used to encrypt and decrypt the data in the Cohesity cluster. Cohesity retrieves the KEKs from KeyControl after a reboot or a restart of the keychain service. If KeyControl is unavailable, the data in the Cluster and Storage Domains will remain encrypted and inaccessible.

1.1. Documents to read first

This guide describes how to configure the Entrust KeyControl server as a KMS in Cohesity.

To install and configure the Entrust KeyControl server cluster, see KeyControl Installation and Upgrade Guide. For information on configuring Entrust KeyControl as a KMIP server, see KeyControl KMIP Vault Overview.

Also refer to the Cohesity online documentation.

1.2. Requirements

- Entrust KeyControl version 5.4 or later.

  An Entrust KeyControl license is required for the installation. You can obtain this license from your Entrust KeyControl account team or through Entrust KeyControl customer support.

- Cohesity DataProtect version 6.8.2 or later.

  A Cohesity license is required for the installation. You can obtain this license from your Cohesity account team or through Cohesity customer support.

Entrust recommends that you allow only unprivileged connections unless you are performing administrative tasks.

1.3. High-availability considerations

Entrust KeyControl uses an active-active deployment, which provides high-availability capability to manage encryption keys. Entrust recommends this deployment configuration. In an active-active cluster, changes made to any KeyControl node in the cluster are automatically reflected on all nodes in the cluster. For information about Entrust KeyControl, see the Entrust KeyControl Product Overview.

1.4. Product configuration

The integration between the Cohesity DataPlatform, Entrust KeyControl, and nShield HSM has been successfully tested in the following configurations:

Product	Version
Cohesity DataProtect (Virtual Edition for VMware)	6.8.2
Entrust KeyControl	10.2

Chapter 2. Procedures

2.1. Install and configure Entrust KeyControl

To install and configure the Entrust KeyControl server cluster, see KeyControl Installation and Upgrade Guide. For information on configuring Entrust KeyControl as a KMIP server, see KeyControl KMIP Vault Overview.

Make sure the Entrust KeyControl Vault gets created and KMIP certificates are generated for Cohesity. These certificates are used in the configuration of the KMS described below.

2.2. Deploy Cohesity DataProtect

Deploy Cohesity DataProtect as described in the Cohesity user documentation: https://docs.cohesity.com/.

2.3. Create Cohesity client certificates in KeyControl

Before we can enable encryption, Cohesity and the KeyControl server must establish a mutual trust relationship. Client certificates are required to facilitate two-way KMIP communications between the KeyControl server and Cohesity. To perform this operation, create the certificate bundle as described in the Managing KMIP Client Certificates section of the *Entrust KeyControl Admin Guide*.

The configuration was tested using certificates without password protection. This client certificate is used to securely authenticate with the Entrust KeyControl server. After you create and download these certificates, you need to upload or import them into the Cohesity appliance.

1. Log in to the Entrust KeyControl server.
2. Select the **Security** icon, then select **Client Certificates > +**.
3. In the **Create a New Client Certificate** dialog, enter the **Certificate Name** and **Expiration Date**.
4. Leave the **Password** field blank.

   This integration requires a password-less client certificate.

5. Select **Create**.
6. After the certificate has been created, select it, and select **Action > Download Certificate**.

7. This downloads a zip file that contains:
    - A `<cert_name>.pem` file that includes both the client certificate and private key.

      In our scenario this file is called `COHESITY.pem`.

      The client certificate section of the `<cert_name>.pem` file includes the lines "`-----BEGIN CERTIFICATE-----`" and "`-----END CERTIFICATE-----`" and all text between them.

      The private key section of the `<cert_name>.pem` file includes the lines "`-----BEGIN PRIVATE KEY-----`" and "`-----END PRIVATE KEY-----`" and all text in between them.

    - A `cacert.pem` file, which is the root certificate for the KMS cluster. It is always named `cacert.pem`.

You will use these files in the Cohesity configuration.

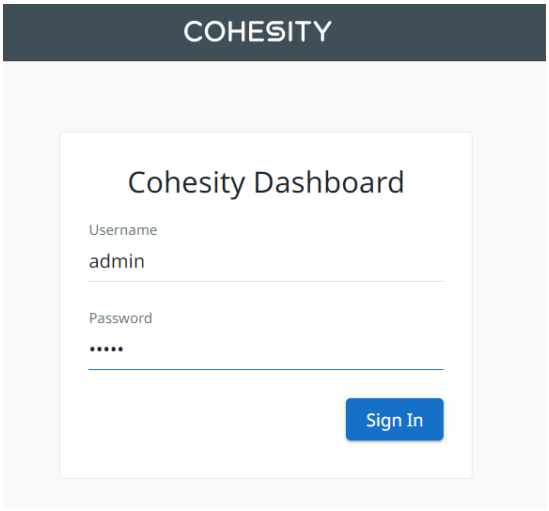
2.4. Configure Cohesity for encryption with an external Key Management System

2.4.1. Configure the Cohesity cluster

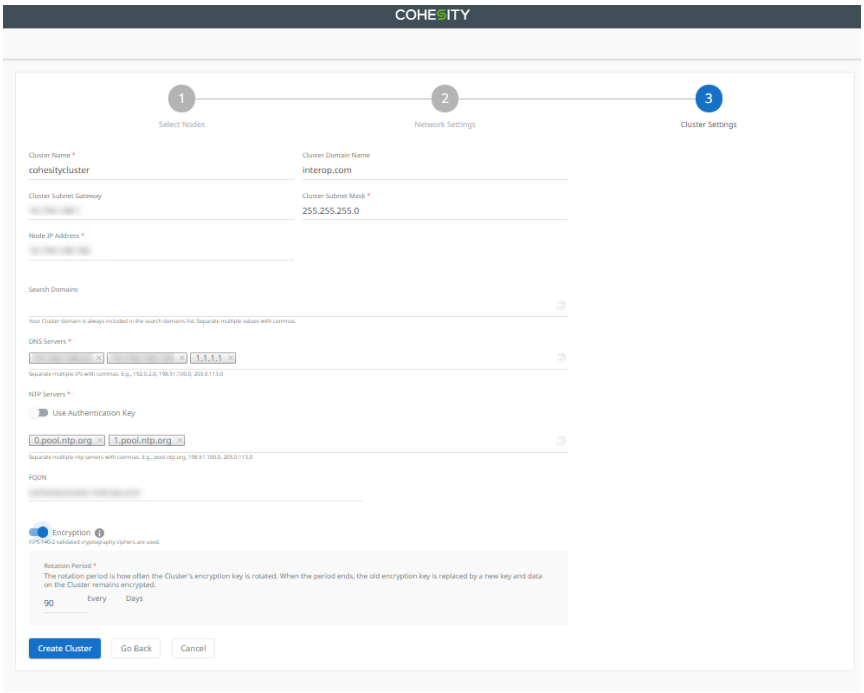
This section assumes that you have not configured the Cohesity DataProtect cluster. If you have already configured this cluster, you can skip to Select the Key Management System.

1. Log in to the Cohesity Web UI:
    a. Point your browser to the Cohesity Appliance IP Address.
    b. Log into the Cohesity Web UI with the default username and password (admin/admin).

```
https://IP_ADDRESS.
```



2. On **Virtual Edition Cluster Setup**, select **Get Started**.
3. Enter cluster information.



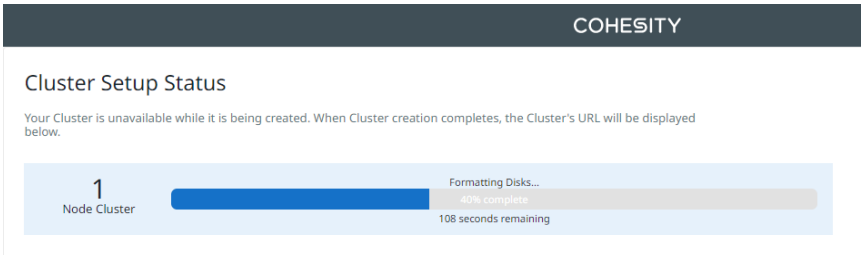
   a. In **Cluster Name**, enter the name of the cluster.
   b. In **Cluster Domain Name**, enter the name of the domain.
   c. In **Cluster Subnet Gateway**, enter the subnet gateway IP address.
   d. In **Cluster Subnet Mask**, enter the subnet mask.
   e. In **Node IP Address**, enter the node IP address.
   f. In **DNS Servers**, enter the IP addresses for all required DNS servers. Separate DNS servers with commas. For example: 192.0.2.0, 198.51.100.0, 203.0.113.0
   g. In **NTP Servers**, enter the IP addresses for all required NTP servers. Separate NTP servers with commas. For example: 0.pool.ntp.org, 1.pool.ntp.org

h. In **FQDN**, enter the full qualified domain name of the cluster.

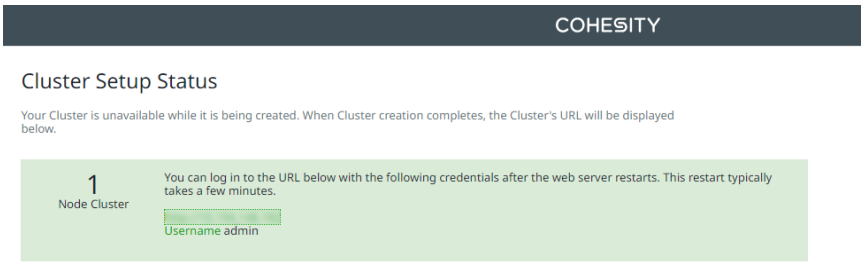
i. Optionally, enable **Encryption** at the cluster level. If you enable encryption at the cluster level, all storage domains created in the cluster will be automatically encrypted with FIPS 140.2 validated cryptography ciphers. You must also set a **Rotation Period** for the cluster's encryption key. At the end of each rotation period, the cluster encryption key is replaced, and all data remains encrypted.

   If encryption is not enabled at the cluster level, you can enable encryption during the Storage Domain creation process if required.

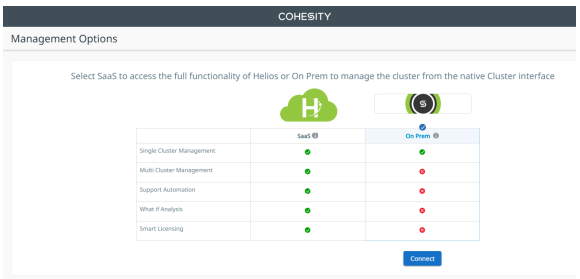
4. Wait until the cluster setup completes.



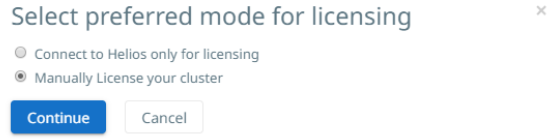
Once the setup is complete, wait a few minutes until the web services are restarted.



5. Log in again to the cluster.
6. Accept the End User License Agreement.
7. In **Management Options**, select either **SaaS** or **On Prem**.



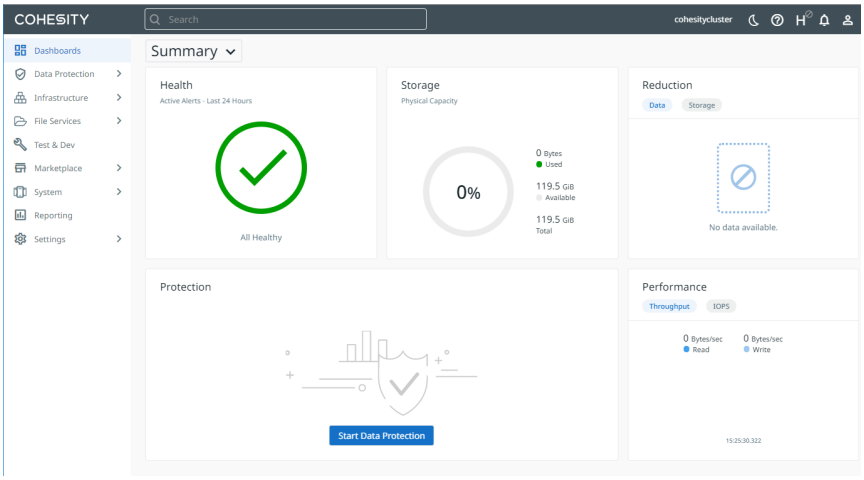
8. On **Select preferred mode for licensing**, select Helios licensing or manual licensing.



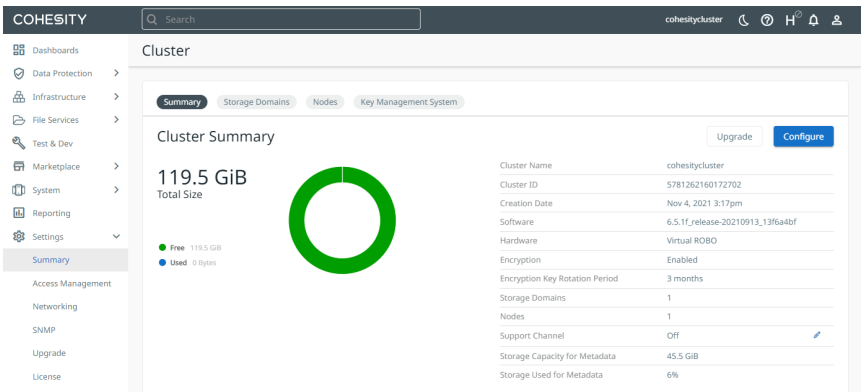
Obtain the license from your Cohesity account team or through Cohesity customer support.

9. Change the **admin** password.

   The Cluster Dashboard appears. For example:

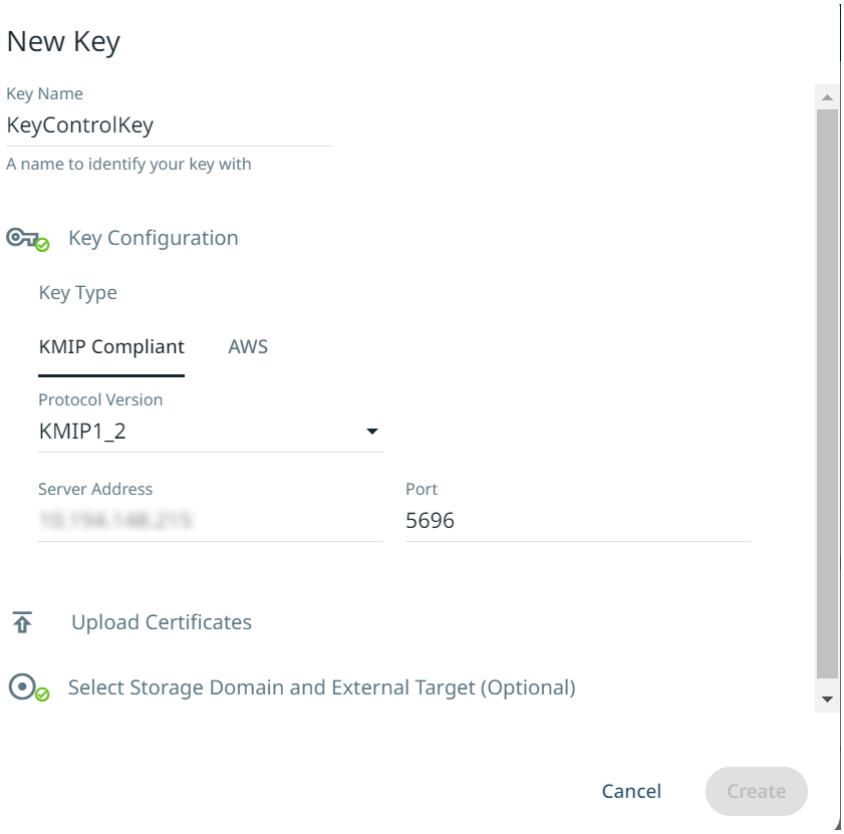


10. Select **Settings > Summary** in the left side bar to view the **Cluster Summary**. For example:

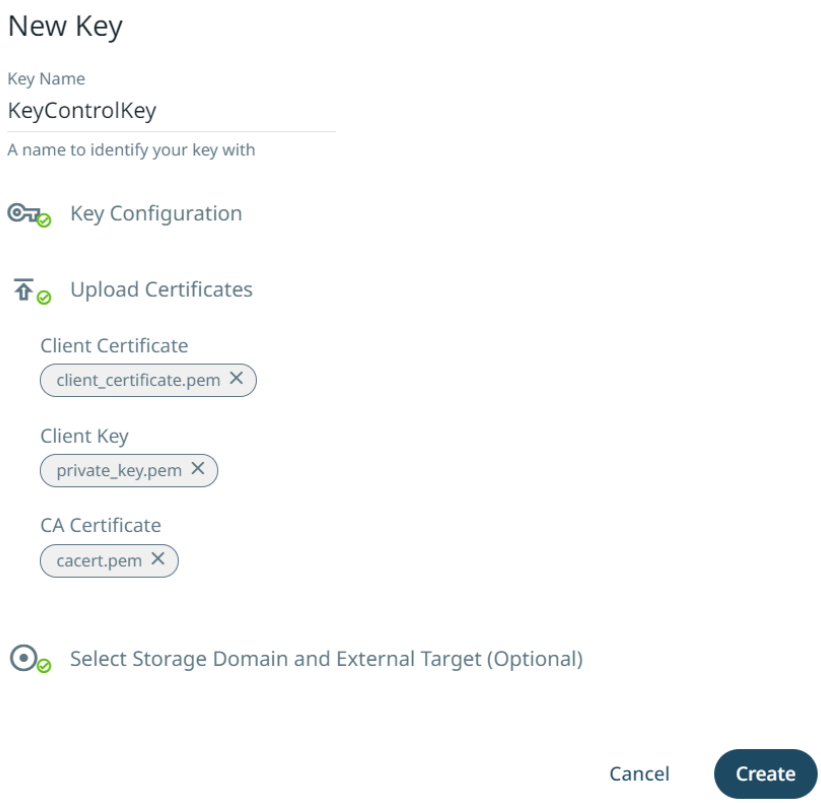


2.4.2. Select the Key Management System

1. Log in to the Cohesity cluster.
2. Select **Key Management System**.
3. In **Key Management System**, create the external Key Management System by selecting **Add New Key**.



4. Expand **Key Configuration**:
   a. In **Key Type**, select **KMIP Compliant**.
   b. In **Protocol Version**, select the protocol version that Entrust KeyControl is configured to use. The KMIP version used in Cohesity must match the version used in Entrust KeyControl. Versions supported by Cohesity and KeyControl are **KMIP1_1**, **KMIP1_2**, and **KMIP1_3**.
   c. In **Server Address**, enter the IP address of the KeyControl server.
   d. In **Port**, enter **5696**.
5. For the Certificates, expand **Upload Certificates**:



- These will be the certificates created in KeyControl that have been downloaded before.
- Certificates must be in PEM format.
- There should be two files: `COHESITY.pem` and `cacert.pem`.
- Break up the `COHESITY.pem` file into two separate files. One file to contain the public key. The other file to contain the private key.
- In **Client Certificate**, select the public key file created from `COHESITY.pem`.
- In **Client Key**, select the private key file created from `COHESITY.pem`.
- In **CA Certificate**, select the `cacert.pem` file.
- For example, the `client_certificate.pem` file contains the public key from inside `COHESITY.pem` file.

```
-----BEGIN CERTIFICATE-----
MIIElTCCA32gAwIBAgIFAJ/aNcYwDQYJKoZIhvcNAQELBQAwVzELMAkGA1UEBhMC
3sYrq6XZjm3aZv8MnK6aroZFww5QWcwUQIEONThwOuQvP7FanSbIejEaqwk3LWlW
.
.
.
8Uy4Xel5zMMjMrR5F1XLRDHaQa9ZSWUDmc9sPmzyvOe99LBz5EL+bCwlxYQ/7Wqn
ugyrDuL7B62OpYmurGeaQ3Z7FfQnhkJmnA==
-----END CERTIFICATE-----
```

- For example, the `private_key.pem` file contains the private key from inside

`COHESITY.pem` file.

```
-----BEGIN PRIVATE KEY-----
MIIJQwIBADANBgkqhkiG9w0BAQEFAASCCS0wggkpAgEAAoICAQCjjmh5g0Z9vtWq
lGL9ZMSLjnmRy9kOUsgb5YYyR48d89m32QfN6B5nO37p/OUzUp2b0k3WZ8uOWiRq
.
.
.
yo0CGhGw6Y0LTygoTdyE2mr+h665KEK0ew8lKuHPAGfwlL0cNSy4mgnwYGs8Xadb
kkkVSC4PfxLD5zKJ4tpDapRP7oigv9Q=
-----END PRIVATE KEY-----
```

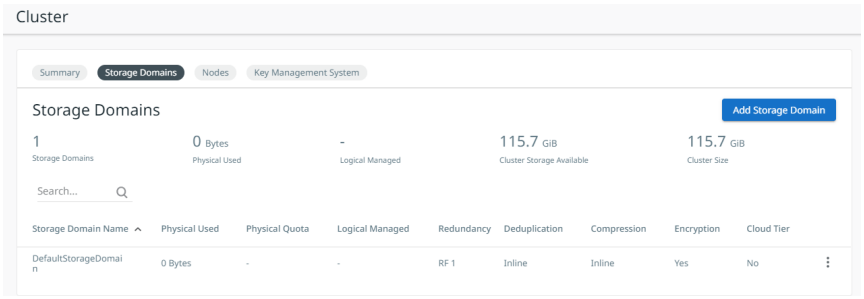
6. Select **Save** to save the settings.

2.5. Create a Cohesity storage domain that uses KeyControl for encryption

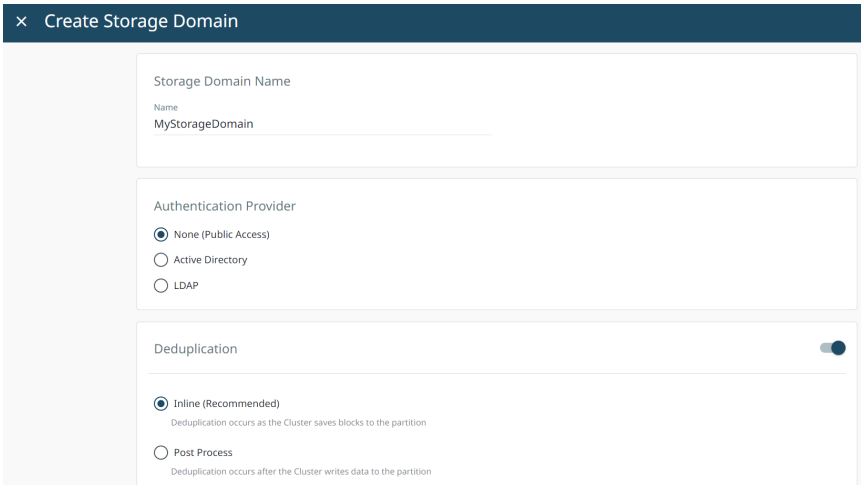
If you already have a suitable storage domain that you want to use, Cohesity supports the migration of data from internal to external KMSs. To migrate a KMS instead of creating a new storage domain, see the Cohesity documentation: https://docs.cohesity.com/7_1_2/Web/UserGuide/Content/Dashboard/Admin/migrate-kms.htm

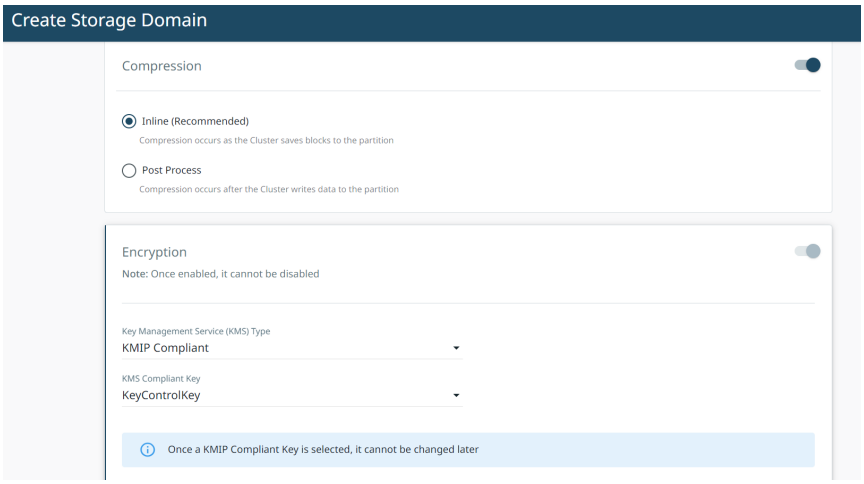
To create a new storage domain:

1. In **Settings > Summary**, select **Storage Domains**.



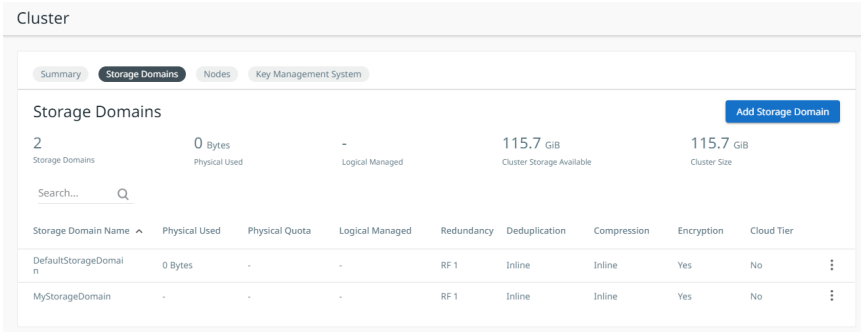
2. Select **Create Storage Domain**.





3. In **Storage Domain Name**, enter the **Storage Domain Name**.
4. Select **Encryption**. This enables encryption at the cluster level.
5. In **Key Management Service (KMS) Type**, select **KMIP Compliant** from the drop down menu.
6. In **KMS Compliant Key**, select your recently created KeyControl Key from the drop down menu.
7. Fill out any other options required.
8. Select **Create**.

   The new storage domain is created and added to the **Storage Domains** list.

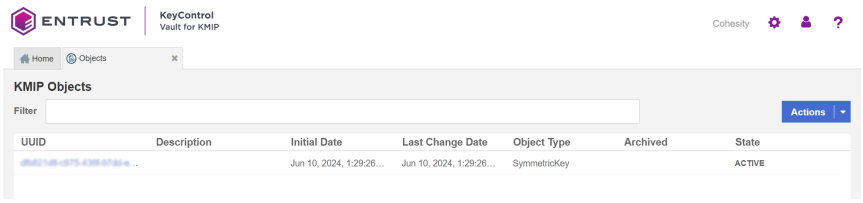


2.6. Check KeyControl for Cohesity keys

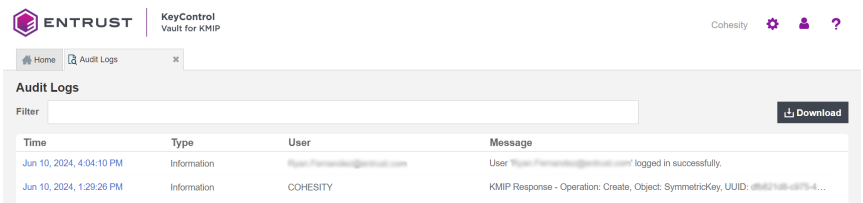
Now that the Cohesity Storage Domain has been created, there should be new keys in KeyControl.

1. Log in to the KeyControl Vault server.
2. Select the **Objects** tab.

   There should be new keys listed that were created when the storage domain was created in the Cohesity cluster. Select one of the keys and validate that it is from Cohesity by selecting the **Custom Attributes** tab. For example:



3. Select the **Audit Logs** page from the home screen on the KeyControl Vault page and validate the creation of keys when you created the storage domain in Cohesity. For example:



2.7. Integrating with an Entrust HSM

For guidance on integrating the Entrust KeyControl with a Hardware Security Module (HSM), consult with your HSM vendor. If you are using an Entrust nShield HSM, refer to the *Entrust KeyControl nShield HSM Integration Guide* available at Entrust documentation library.

Chapter 3. Cohesity DataPlatform CLI

You may also configure Entrust KeyControl KMS using the Cohesity DataPlatform CLI. Here are some examples of CLI commands that can be used to configure the KMS.

3.1. Log in to the Cohesity server

```
% iris_cli -server xx.xxx.xxx.xxx -username=admin -password=xxxxxx

Cohesity Command Line Interface.
Version: 1.0
This command line tool helps to run any cluster management operations.

admin@xx.xxx.xxx.xxx>
```

3.2. Create a KMIP KMS

```
admin@xx.xxx.xxx.xxx> kms create-kmip

  DESCRIPTION
    Create a new kmip KMS.

  PARAMS
    ca-certificate-path          [string]              required  File path to ca-certificate.
    client-certificate           [string]              required  File path to client-certificate.
    client-key                   [string]              required  File path to client-key.
    ip                           [string]              required  IP address of the KMS.
    kmip-protocol-version        [string]              required  kmip-protocol-version
    name                         [string]              optional  Name of the KMS.
    port                         [int]                 required  KMS Port. Default KMIP port is 5696.
```

3.3. List current KMS settings

```
admin@xx.xxx.xxx.xxx> kms list

KMS ID                       : 0
KMS TYPE                     : kInternalKMS
KMS NAME                     : Internal KMS
KMS CONNECTION STATUS        : false


KMS ID                       : 5287
KMS TYPE                     : kCryptsoftKMS
KMS NAME                     : KeyControl
KMS CONNECTION STATUS        : true
KMS IP                       : xx.xxx.xxx.xxx
KMS PORT                     : 5696
KMIP PROTOCOL VERSION        : KMIP1_2
 CLIENT CERTIFICATE EXPIRY DATE: Wednesday, 02-Nov-22 10:13:59 EDT
```

3.4. Modify Cohesity DataPlatform KMS settings

If you update the Key Management settings after initial configuration, the keychain service must be restarted for the new settings to take effect. This restart is done using the CLI using the following steps.

For instructions on accessing and general use of the Cohesity CLI, please see the **Cohesity CLI** section of the *Cohesity Virtual Edition Setup Guide*.

```
admin@xx.xxx.xxx.xxx> cluster restart service-names="keychain"
Success: Restarting the cluster services [keychain] ...

admin@xx.xxx.xxx.xxx> cluster status
CLUSTER ID                    : 5781262160172702
CLUSTER NAME                  : cohesitycluster
CLUSTER INCARNATION ID        : 1636053457920
SERVICE STATE SYNC            : DONE
CLUSTER ACTIVE OPERATION      : RESTARTING SERVICES
CLUSTER HEAL STATUS           : NORMAL
CLUSTER IP Preference         : 1

NODE ID                       : 2639329736857246
NODE IPS                      : xx.xxx.xxx.xxx
SOFTWARE VERSION              : 6.5.1f_release-20210913_13f6a4bf
ACTIVE OPERATION              : kClusterRestart
SERVICE NAME                  :
    alerts                    : 29301, 29322
    apollo                    : 29378, 29395
    athena                    : 34581, 34610
    atom                      : 34580, 34596
    bifrost_broker            : 23858, 23865
    bridge                    : 30906, 38313
    bridge_proxy              : 34731, 34870
    eagle_agent               : 23790, 41368
    gandalf                   : 60546, 60549
    groot                     : 42065, 42068
    iris                      : 7240, 7262
    iris_proxy                : 540, 22376
    keychain                  : 17784, 17844
    librarian                 : 25926, 25944
    logwatcher                : 63390
    magneto                   : 40109, 40165
    newscribe                 : 23755, 23777
    nexus                     : 54968
    nexus_proxy               : 61200, 61203
    patch                     : 17875, 18107
    rtclient                  : 17874, 17895
    smb2_proxy                : 17782, 17852
    smb_proxy                 : 17877, 17924
    stats                     : 29337, 29345
    statscollector            : 63389
    storage_proxy             : 17873, 18215
    tricorder                 : 23694
    vault_proxy               : 17876, 17909
    yoda                      : 37198, 37226
```

Chapter 4. Troubleshooting

You might encounter errors while configuring Entrust KeyControl KMS or Storage Domain settings in Cohesity DataPlatform. The error might be caused by invalid input parameters or communications errors.

The most common errors are:

1. A KMS validation error while configuring the KMS.
2. A KMS unreachable error while creating a Storage Domain.

4.1. KMS validation error with KMS configuration

If the Cohesity cluster cannot communicate with Entrust KeyControl when configuring the Key Management settings, the following generic KMS validation error appears:

```
KMS Validation error.
```

If it does, take the following steps:

1. Verify correct addressing and basic network connectivity between Entrust KeyControl and the Cohesity cluster.
2. Verify port 5696 is configured on the Cohesity DataPlatform KMS settings page and that firewalls are open for that port.
3. If any of the uploaded certificate files or private key file on the Cohesity DataPlatform KMS settings page were created on a Windows system, recreate them on a Linux system.

The Cohesity KMS client only accepts an SSL certificate in PEM format that contains a Unix-style newline character, which is '\n'. Format your certificates accordingly — in Windows, replace '\r\n' with '\n' and on Mac OS, replace '\r' with '\n' — and then load the certificates.

4. Verify that the CA certificate uploaded on the Cohesity DataPlatform KMS settings page is the internal root CA certificate from Entrust KeyControl. The Cohesity cluster needs the root CA certificate to validate the server certificate that is delivered to it while establishing a TLS session.
5. Proper licensing must be in place.

4.2. KMS unreachable error during storage domain creation

When you create a new Storage Domain, the Cohesity cluster immediately sends a key generation request to Entrust KeyControl. If a TLS session is not established or if Entrust KeyControl is unreachable, the Storage Domain will not be created, and you will see the following error:

```
KMS is unreachable. Try again.
```

A possible cause of this error is that the TLS session with Entrust KeyControl has been dropped due to inactivity. The Cohesity cluster will immediately take action to re-establish the connection. You may see an error message indicating that the KMS is unreachable before the connection is re-established. In this case, select **Create Storage Domain** to try again. If the problem was a dropped TLS session, the connection should then re-establish.

If the problem was not just the lack of a TLS session, and there is indeed a connectivity issue of some type, you will either continue to see the KMS is unreachable error or possibly the internal error message below. To resolve this, try the steps in KMS Validation Error above.

Chapter 5. Additional resources and related products

5.1. nShield HSMs

5.2. nShield as a Service

5.3. KeyControl

5.4. Entrust products

5.5. nShield product documentation