



**ENTRUST**

# Cohesity and Entrust KeyControl

with nShield® HSM Integration Guide

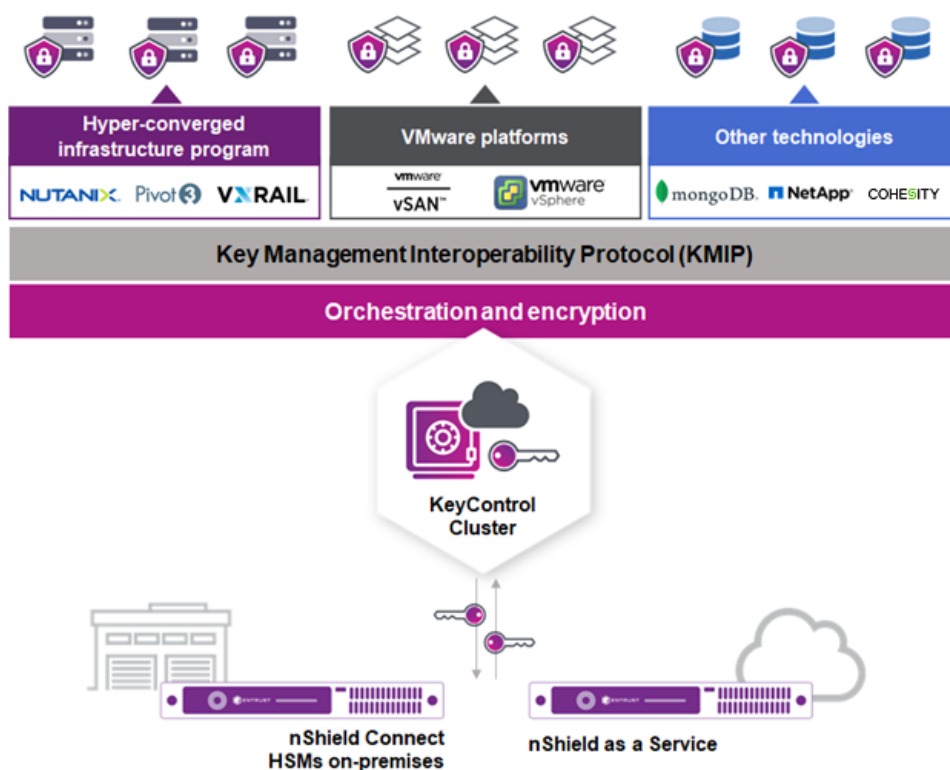
24 Nov 2021

# Contents

1. Introduction	3
1.1. Requirements	3
1.2. High-availability considerations	4
1.3. Product configuration	4
2. Procedures	5
2.1. Install the Entrust KeyControl server	5
2.2. Configure the KeyControl server	5
2.3. Configure the KeyControl server as a KMIP server	5
2.4. Configure the nShield HSM in the KeyControl server	6
2.5. Deploy Cohesity Virtual Edition using VMware vCenter	10
2.6. Create Cohesity client certificates in KeyControl	13
2.7. Configure Cohesity for encryption with an external Key Management System	14
2.8. Create a Cohesity storage domain that uses KeyControl for encryption	19
2.9. Check KeyControl for Cohesity keys	20
3. Cohesity DataPlatform CLI	22
3.1. Log in to the Cohesity server	22
3.2. Create a KMIP KMS	22
3.3. List current KMS settings	22
3.4. Modify Cohesity DataPlatform KMS settings	23
4. Troubleshooting	24
4.1. KMS validation error with KMS configuration	24
4.2. KMS unreachable error during storage domain creation	25

# 1. Introduction

This document describes how to configure and integrate Entrust KeyControl (KMS) with a Cohesity DataPlatform using KMIP. Communication between the Cohesity cluster and the Entrust KeyControl Key Management Server (KMS) cluster is enabled via the 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). It is this CA that 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. 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 Virtual Edition version 6.5.1.

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

## 1.2. 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](#).

## 1.3. Product configuration

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

Product	Version
Cohesity Virtual Edition	6.5.1
Entrust KeyControl	5.4
nShield client software	12.60.11
nShield Connect XC	12.50.11 image version 12.60.10

## 2. Procedures

### 2.1. Install the Entrust KeyControl server

The Entrust KeyControl server is a software solution deployed from an OVA or ISO image. Entrust recommends that you read the [HyTrust KeyControl Installation Overview](#). To configure a KeyControl cluster (active-active configuration is recommended), Entrust recommends the OVA installation method, as described in [HyTrust KeyControl OVA Installation](#).

The KeyControl OVA must be deployed from the vCenter server. Do not deploy from an ESXi host.

After the KeyControl server is deployed, configure the first KeyControl node as described in the [HyTrust Configuring the First KeyControl Node installation guide](#).

After completing this procedure, create the recommended active-active cluster. To do this, add a second node as described in the [HyTrust Adding a New KeyControl Node to an Existing Cluster \(OVA Installation\)](#)

Entrust recommends deploying the solution with a minimum of four nodes for an active-active cluster solution that instantiates a high availability architecture. However, an active-active cluster is not a requirement and a single KeyControl node can be deployed to perform the functions of KMIP.

Your KeyControl license determines how many KeyControl nodes you can have in a cluster. For full information about the KeyControl licensing, see the [HyTrust Managing the KeyControl License admin page](#).

### 2.2. Configure the KeyControl server

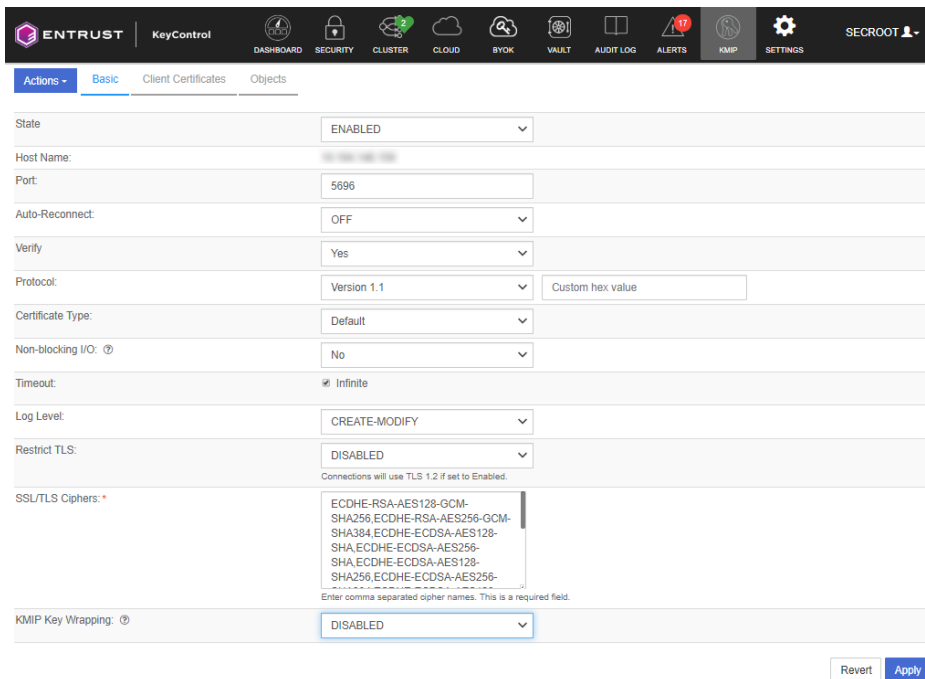
After the Entrust KeyControl server is deployed and the initial installation is complete, you can configure the network settings, e-mail server preferences, and certificate configuration. For these procedures, see the [HyTrust KeyControl System Configuration admin guide](#).

### 2.3. Configure the KeyControl server as a KMIP server

This integration uses external key management. To use external key management, the Cohesity cluster requires an external key management server such as the Entrust KeyControl server. The KeyControl server is the KMIP server and the Cohesity cluster is the KMIP client.

To configure the KeyControl server as a KMIP server, see the [HyTrust Configuring a KeyControl KMIP Server](#) section of the admin guide.

1. Log in to the Entrust KeyControl server.
2. Select the **KMIP** icon on the top bar and configure the KMIP settings according to image below:



The screenshot shows the KeyControl web interface with the KMIP configuration page. The top navigation bar includes icons for Dashboard, Security, Cluster, Cloud, BYOK, Vault, Audit Log, Alerts, KMIP, Settings, and Secroot. The main content area has tabs for Actions, Basic, Client Certificates, and Objects. The configuration form includes the following fields:

- State: ENABLED (dropdown)
- Host Name: [Redacted]
- Port: 5696
- Auto-Reconnect: OFF (dropdown)
- Verify: Yes (dropdown)
- Protocol: Version 1.1 (dropdown) with a Custom hex value field
- Certificate Type: Default (dropdown)
- Non-blocking I/O: No (dropdown)
- Timeout: Infinite (dropdown)
- Log Level: CREATE-MODIFY (dropdown)
- Restrict TLS: DISABLED (dropdown) with a note: "Connections will use TLS 1.2 if set to Enabled."
- SSL/TLS Ciphers: A list of cipher suites including ECDHE-RSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES128-SHA, ECDHE-ECDSA-AES256-SHA, and ECDHE-ECDSA-AES128-SHA256. A note below reads: "Enter comma separated cipher names. This is a required field."
- KMIP Key Wrapping: DISABLED (dropdown)

Buttons for Revert and Apply are located at the bottom right of the form.

3. Select **ENABLED** from the State drop-down menu.
4. Select **Version 1.1** from the Protocol drop-down menu.
5. Select **Apply**.

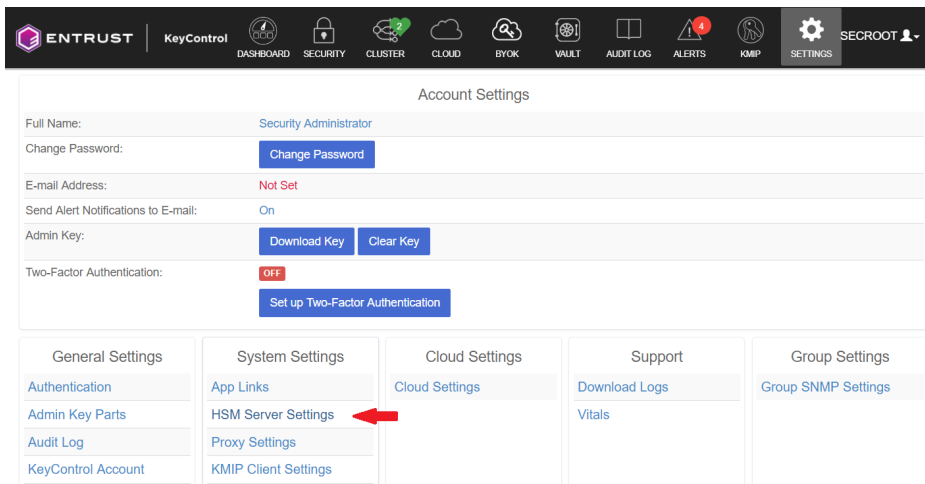
## 2.4. Configure the nShield HSM in the KeyControl server

For instructions on how to integrate an nShield HSM with KeyControl, see the *Entrust KeyControl nShield HSM Integration Guide*.

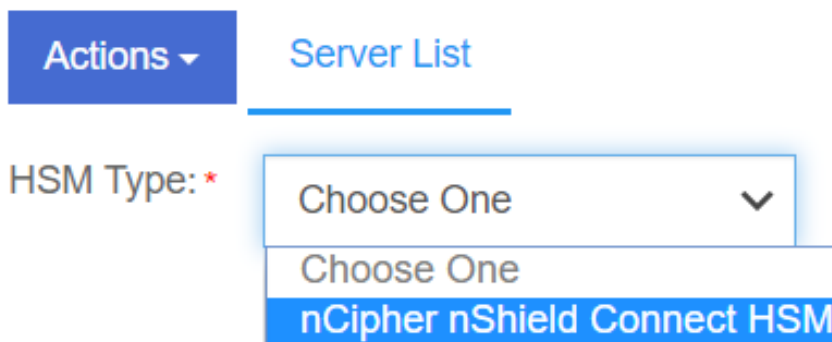
If you want to use an HSM to be used with KeyControl and have not yet configured one, please follow the procedures in this section.

### 2.4.1. Initialize the HSM on KeyControl

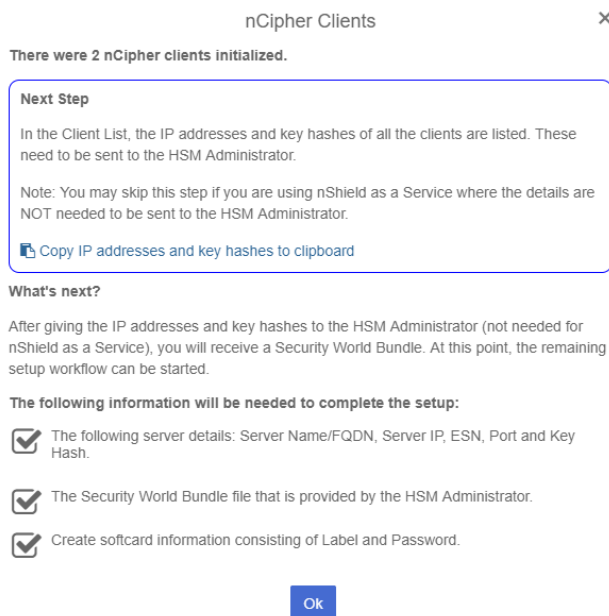
1. Log in to the KeyControl web user interface using Security Admin privileges.
2. In the top menu bar, select **Settings**.
3. In **Account Settings**, select **HSM Server Settings**.



4. In **Actions**, select **nCipher nShield Connect HSM**.



5. In **nCipher Clients**, select **Copy IP address and key hashes to clipboard**.



6. Paste the contents of the clipboard into a file. Your HSM administrator will need the IP addresses and hash pairs to add the KeyControl nodes as HSM clients.

The following is an example data file for a 2-node KeyControl cluster:

```
10.194.148.159, 4dc3acd8763a80ceac412d1c4aadb382e41ab073
10.194.148.161, 4dba62cbbbe2f5a21f121e7132558db8aa592596
```

## 2.4.2. Add the KeyControl node(s) to the HSM

Send the IP address and hash pair for each KeyControl node in the cluster to the HSM administrator. The HSM administrator adds each KeyControl node as a client to the HSM and sends back the following information:

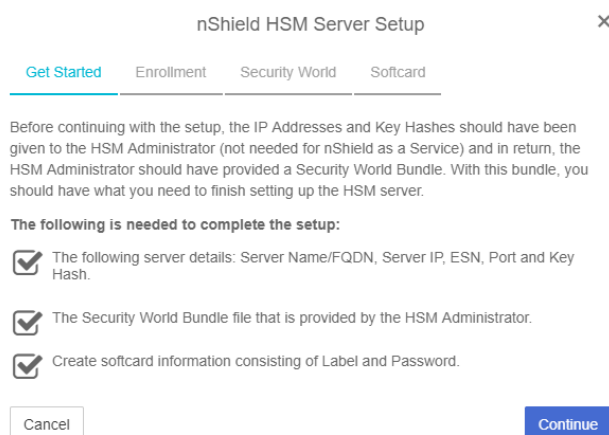
- A zipped file that contains the nShield Security World and HSM module files.
- The FQDN of the HSM.
- The IP address of the HSM.
- The Electronic Serial Number (ESN) and the key hash of the HSM. This can also be obtained by running the following command on the nShield RFS server:

```
% anonkneti <HSM IP address>
```

- The network port number that the HSM uses.

## 2.4.3. Set up the nShield HSM Server

1. After you have copied the IP address and key hashes and added the KeyControl nodes as clients to the HSM, select **OK** in the **nCipher Clients** dialog.
2. In the **nShield HSM Server Setup** dialog, select **Continue**.



3. In the **Enrollment** step of the configuration:



nShield HSM Server Setup ×

Get Started **Enrollment** Security World Softcard

---

**Enroll with Server Settings**

Server Name \*

Server IP \*

ESN \*

Port \*

Key Hash \*

- a. in **Server Name**, enter the server FQDN of the HSM.
- b. In **Server IP**, enter the IP address of the HSM.
- c. In **ESN**, enter the ESN of the HSM.
- d. In **Port**, enter the port number if it is different from **9004**.
- e. In **Key Hash**, enter the key hash of the HSM.

4. Select **Enroll and Continue**.

5. In the **Security World** step of the configuration, select **Browse** and browse to the zipped file that you received from the HSM administrator.

nShield HSM Server Setup ×

Get Started Enrollment **Security World** Softcard

---

**Upload Security World Bundle**

A security world bundle file needs to be provided from the HSM Administrator. Upload this file in order to enroll the KeyControl nodes.

6. Select **Upload and Continue**.

7. In the **Softcard** step of the configuration:

nShield HSM Server Setup ×

Get Started   Enrollment   Security World   **Softcard**

**Create Softcard**

Create a label and passphrase to link to the HSM Server.

**⚠** Keep a record of the softcard label and password. These will both be needed during a Master Key Recovery (MKR).

Softcard Label \*

Softcard Password \*

Confirm Softcard Password \*

- a. In **Softcard Label**, enter a unique name. This value is user-defined.
- b. In **Softcard Password**, enter a password. This value is user-defined. Then, confirm the password.

8. Select **Complete Setup**.

9. The nShield Connect HSM is now configured to work with Entrust KeyControl.

## 2.4.4. Enable KMIP service and KMIP key wrapping

1. In the top menu bar, select **KMIP** and then select **Basic**.
2. In **KMIP Key Wrapping**, select **System HSM (nShield Connect HSM: xxx.xxx.xxx.xxx)**.  
For example:

3. In **HSM Root Key Label**, enter a unique name for the HSM Root Key.
4. In **KEK Cache Timeout**, enter how long you want KeyControl to cache the HSM-derived Key Encryption Keys (KEKs). The maximum length is 24 hours. During integration, this was set to **0** so that no cache was used, and KeyControl had to use the HSM every time.
5. Select **Apply**.

## 2.5. Deploy Cohesity Virtual Edition using VMware vCenter

1. Obtain the single node Virtual Edition for VMware OVA file from the Cohesity Download Site.

2. Using the VMware vSphere Web Client, log in to the vCenter Server that will host the Virtual Edition Virtual Machine.
3. In the inventory located in the left panel, navigate to your vCenter server, right-click on the vCenter root and select **Deploy OVF Template**.
4. Enter the URL or a local file location for the Cohesity Virtual Edition OVA file and select **Next**.
5. In **Virtual Machine Name**, enter a unique name for the Virtual Machine.
6. In **Select a computer resource**, select the ESXi to host the Cohesity Virtual Machine. Then, select **Next**.
7. Review the details and select **Next**.
8. In **Deployment Configuration**, select an appropriate deployment configuration provided by Cohesity:
  - a. The **SMALL** configuration supports a Virtual Machine with a minimum of 4 vCPUs, 32 GB of memory and a 64 GB virtual disk to store the operating system.
  - b. The **LARGE** configuration supports a Virtual Machine with a minimum of 8 vCPUs, 64 GB of memory and a 64 GB virtual disk to store the operating system.
9. Select **Next**.
10. Select the storage location for the deployed template.
11. Select a VM storage policy.
12. In **Virtual Disk Format**, select **Thick Provision Lazy Zeroed**.
13. Select **Next**.
14. Select a destination network for the **Data Network** and for the **Secondary Network**.
15. Select the IP address allocation type, either dynamic DHCP or static (manual).
16. Select **Next**.
17. If you are using static (manual) networking, specify the following Data Network properties:
  - **Network IP Address**
  - **Network Netmask**
  - **Default Gateway**
18. Leave the Secondary Network properties blank.

If a Secondary Network interface is configured, the Secondary Network is used as the default gateway for the Cohesity cluster. For more information, see **Default Gateway for Virtual Edition** in the *Cohesity Setup Guide (Cohesity Virtual Edition for VMware)*.

19. If you are using DHCP networking, leave the **Network IP Address**, **Network Netmask**, and **Default Gateway** properties blank.

20. Select **Next**.
21. Review all the settings.
22. Select **Finish**.

The process to deploy the VM starts. The **Recent Tasks** panel displays the status of the deployment of the Cohesity template. Wait until the VM is deployed before continuing to the next procedure. Do not power on the VM as you still need to add disks to it.

### 2.5.1. Attach the Metadata Disk and the Data Tier Disk to the VM

You will need to attach two disks to the Cohesity VM. These disks have specific requirements in a production environment. Please refer to the *Cohesity Setup Guide (Cohesity Virtual Edition for VMware)* for more details.

Use the following configuration:

**Metadata Disk**    50GB

**Data Tier Disk**    100GB

Use the procedure below to add the disks. For the first disk:

1. Attach the disk to the Virtual Machine using the VMware vSphere Web Client.
2. In the left panel, browse for the new Virtual Machine. Right-click the new Virtual Machine and select **Edit Settings**.
3. Select **ADD NEW DEVICE**.
4. Under **Disk, Drives and Storage**, Select **Hard Drive**.

A new hard disk is created.

5. Specify an appropriate disk size, either 50GB or 100GB. The Metadata drive size must be smaller than the Data Tier drive size.
6. To view and edit the rest of the hard disk settings, expand **New Hard disk**.
7. In **Disk Provisioning**, select **Thick Provision Lazy Zeroed**.
8. In **Disk Mode**, select **Independent - Persistent**.
9. Select **OK** to create the disk.

Repeat the process for the second disk.

### 2.5.2. Start the new Cohesity Virtual Machine

1. In the left pane, find the new Virtual Machine.

2. Right-click the Virtual Machine and select **Power On**.

Wait until the VM is powered on. The process of bringing up all of the services and getting the IP address may take several minutes. Once the VM has an IP address, try to open up a browser and access it. For example:

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

The web server can take some time to be available. If the web server does not respond, keep trying.

## 2.6. 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 [Creating KMIP Client Certificate Bundles](#) 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 **KMIP** icon on the top bar, then select **Client Certificates > Actions > Create Certificate**.
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.

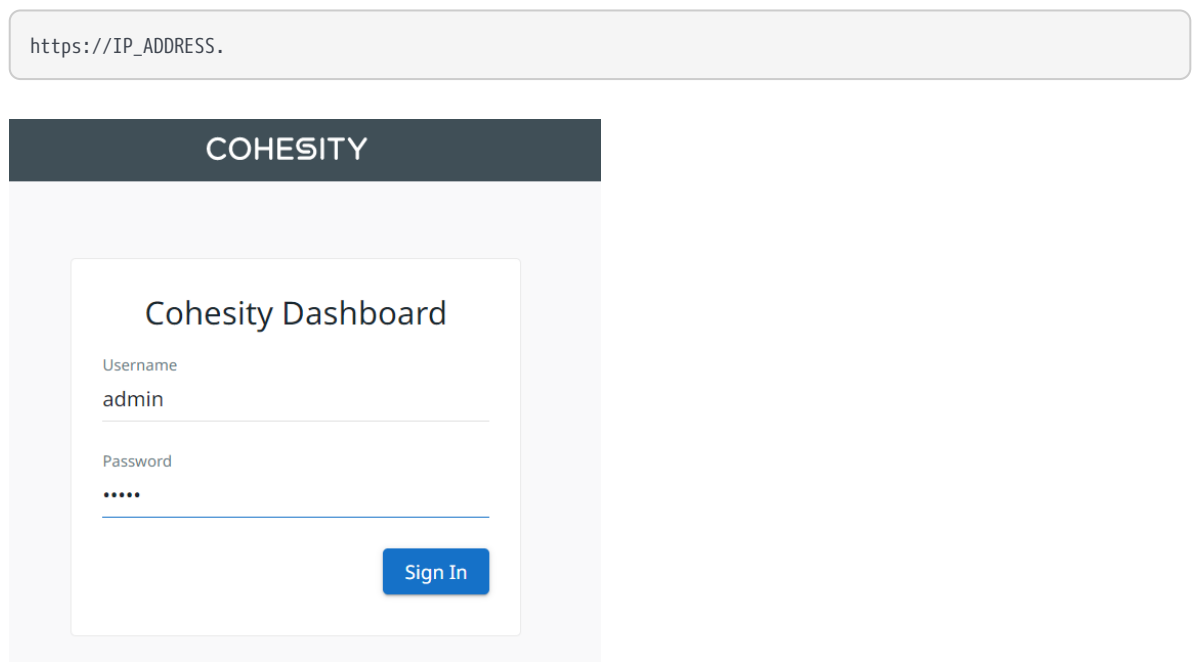
The private key section of the `<cert_name>.pem` file includes the lines "`-----BEGIN PRIVATE KEY-----`" and "`-----END PRIVATE KEY-----`" and all text in between them.

- 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.

## 2.7. Configure Cohesity for encryption with an external 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).

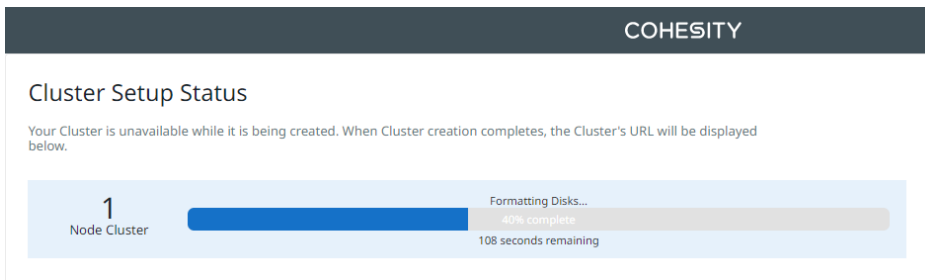


2. On **Virtual Edition Cluster Setup**, select **Get Started**.
3. Enter cluster information.

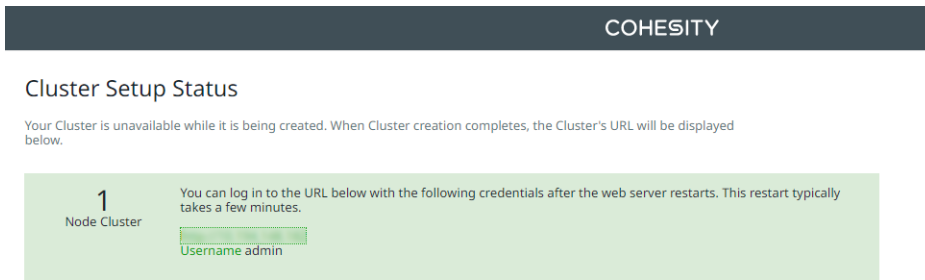
- a. In **Cluster Name**, enter the name of the cluster.
- b. In **Cluster Domain Name**, enter the name of the domain.
- c. In **Cluster Subnet Gateway**, enter the subnet gateway IP address.
- d. In **Cluster Subnet Mask**, enter the subnet mask.
- e. In **Node IP Address**, enter the node IP address.
- 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.

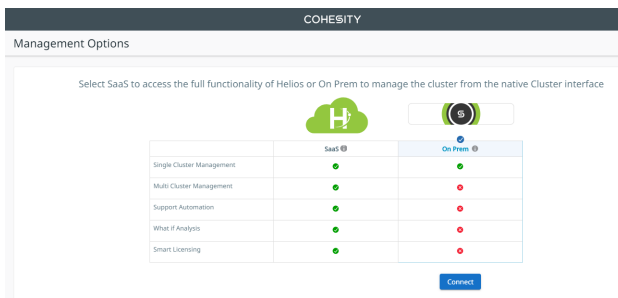
4. Wait until the cluster setup completes.



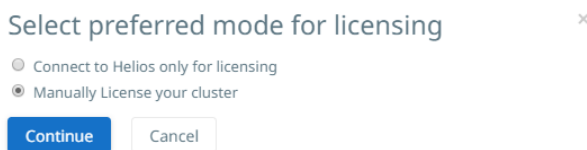
Once the setup is complete, wait a few minutes until the web services are restarted.



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.

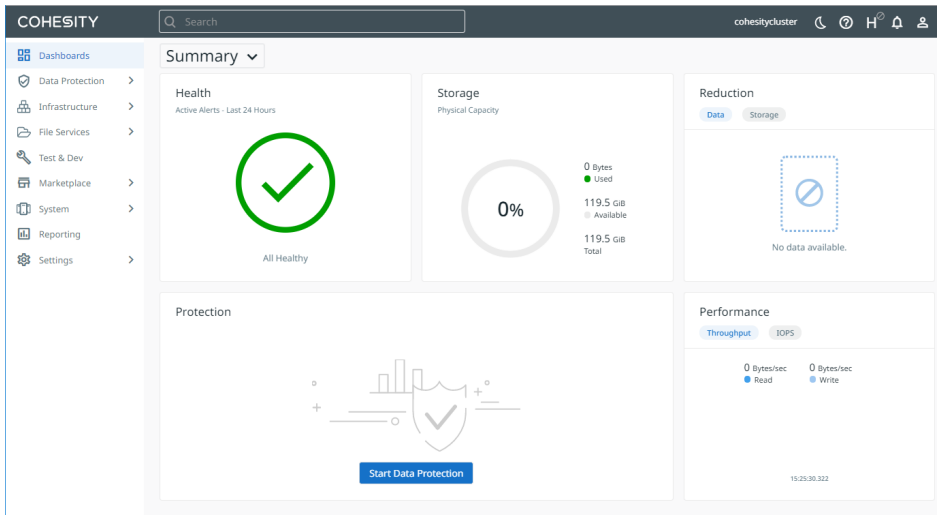


Obtain the license from your Cohesity account team or through Cohesity customer support.

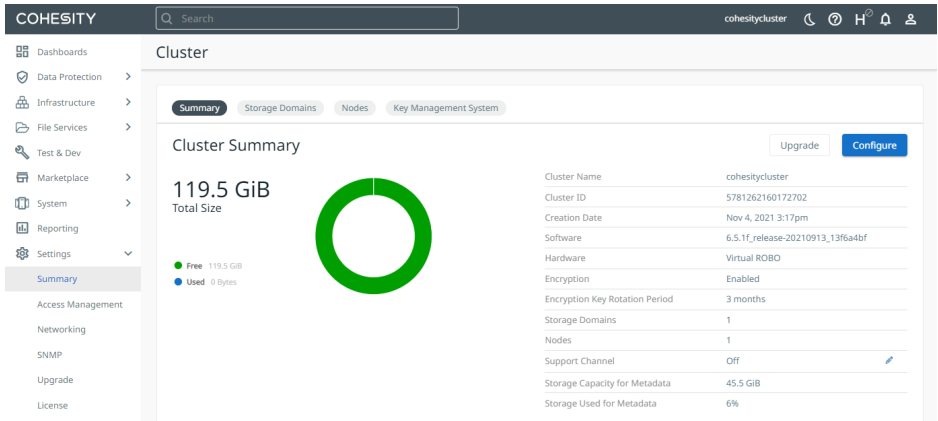
9. Change the **admin** password.

The Cluster Dashboard appears. For example:





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



11. Select **Key Management System**.

12. In **Key Management System**, create the external Key Management System:
  - a. In **Server Type**, select **KMIP Compliant**.
  - b. In **Server Name**, enter **KeyControl**.
  - c. In **Protocol Version**, enter the protocol version set when Entrust KeyControl was configured. Versions supported by Cohesity and KeyControl are **KMIP1\_1**, **KMIP1\_2**, and **KMIP1\_3**.
  - d. In **Server IP**, enter the IP address of the server.
  - e. In **Port**, enter **5696**.
  - f. For the Certificates do the following:
    - 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-----
MIIE1TCCA32gAwIBAgIFAJ/aNcYwDQYJKoZIhvcNAQELBQAwVzELMAkGA1UEBhMC
3sYrQ6XZjm3aZv8MnK6aroZFww5QWcwUQIEONThwOuQvP7FanSbIejEaqwk3LWlW
.
.
.
8Uy4Xe15zMMjMrR5F1XLRDHQa9ZSWUDmc9sPmzyv0e99LBz5EL+bCwLxYQ/7Wqn
ugyrDuL7B62OpYmurGeaQ3Z7FfQnhkJmnA==
-----END CERTIFICATE-----

```

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

```

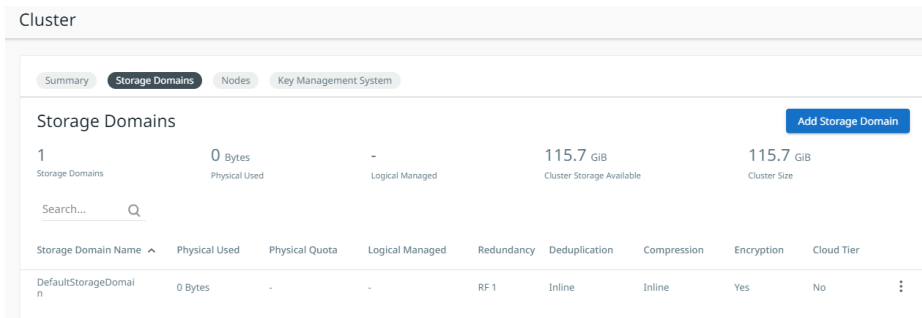
-----BEGIN PRIVATE KEY-----
MIIJQwIBADANBgkqhkiG9w0BAQEFAASCCS0wggkpAgEAAoICAQCj jmh5g0Z9vtWq
1GL9ZMSL jnmRy9k0Usgb5YYyR48d89m32QfN6B5n037p/OUzUp2b0k3WZ8u0WiRq
.
.
.
yo0CGhGw6Y0LTygoTdyE2mr+h665KEK0ew81KuHPAGfwLL0cNSy4mgnwY6s8Xadb
kkkVSC4PfxLD5zKJ4tpDapRP7oigv9Q=
-----END PRIVATE KEY-----

```

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

## 2.8. Create a Cohesity storage domain that uses KeyControl for encryption

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



- Select **Add Storage Domain**.

### Add Storage Domain

Storage Domain Name\*  
MyStorageDomain

Deduplication

Inline Deduplication  
If on, deduplication occurs as the Cluster saves blocks to the Partition. If off, deduplication occurs after the Cluster writes data to the Partition.

Compression

Inline Compression  
If on, compression occurs as the Cluster saves blocks to the Partition. If off, compression occurs after the Cluster writes data to the Partition.

Encryption  
Encryption is on at the Cluster level and therefore the Storage Domain is automatically encrypted.

[Show Advanced Settings](#)

[Create Storage Domain](#) [Cancel](#)

3. In the **Add Storage Domain** dialog, enter the **Storage Domain Name**.
4. Select **Encryption**. This enables encryption at the cluster level.
5. Select **Create Storage Domain**.

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

Cluster

Summary **Storage Domains** Nodes Key Management System

Storage Domains [Add Storage Domain](#)

2 Storage Domains 0 Bytes Physical Used - Logical Managed 115.7 GiB Cluster Storage Available 115.7 GiB Cluster Size

Search... Q

Storage Domain Name	Physical Used	Physical Quota	Logical Managed	Redundancy	Deduplication	Compression	Encryption	Cloud Tier
DefaultStorageDomain	0 Bytes	-	-	RF 1	Inline	Inline	Yes	No
MyStorageDomain	-	-	-	RF 1	Inline	Inline	Yes	No

## 2.9. 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 server.
2. Go to the **KMIP** page and 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:

The screenshot shows the KeyControl interface with the 'Objects' tab selected. The table below lists 15 objects, all of type 'SymmetricKey'. Below the table, the 'Custom Attrs' section shows a single attribute: 'Cohesity 5781262160172702: val'.

UUID	State	Archived	Initial Date	Last Change Date	Object Type	Identifier	Description
f6cba7fa-d2e2-4741-b0ef-b...	Active	No	11/4/2021, 3:47:53 PM	11/4/2021, 3:47:53 PM	SymmetricKey		
4fa8d89c-c22c-4d3c-bd8d-c...	Active	No	11/4/2021, 3:44:55 PM	11/4/2021, 3:44:55 PM	SymmetricKey		
b9639c96-3cd5-4d7f-82ee-f...	Active	No	11/4/2021, 1:48:39 PM	11/4/2021, 1:48:39 PM	SymmetricKey		
6407fb89-8359-4e2a-9582-...	Active	No	11/4/2021, 1:48:39 PM	11/4/2021, 1:48:39 PM	SymmetricKey		
289a2a3b-efad-4d34-b927-...	Active	No	11/4/2021, 1:48:38 PM	11/4/2021, 1:48:38 PM	SymmetricKey		
7b91d322-29bc-4678-8f80-f...	Active	No	11/4/2021, 1:48:38 PM	11/4/2021, 1:48:38 PM	SymmetricKey		
2eb18a6a-46c9-45fa-a42f-b...	Active	No	11/4/2021, 1:48:36 PM	11/4/2021, 1:48:36 PM	SymmetricKey		
755d88cc-449f-410f-a174-b...	Active	No	11/4/2021, 1:48:35 PM	11/4/2021, 1:48:35 PM	SymmetricKey		
d7103cca-48b3-42f3-a350-8...	Active	No	7/22/2021, 10:26:49 AM	7/22/2021, 10:26:50 AM	SymmetricKey		
ca917454-489e-419a-8358-...	Active	No	7/22/2021, 10:04:37 AM	7/22/2021, 10:04:38 AM	SymmetricKey		

Vendor:  
Product:  
Product version:  
Component:  
Identifier:  
Name:  
Policy:  
Cohesity 5781262160172702: val

- Go to the **Alerts** page and validate the keys that were created when you created the storage domain in Cohesity. For example:

The screenshot shows the KeyControl interface with the 'Alerts' page selected. The table below lists 18 messages, all related to the creation of SymmetricKey objects.

Date	Message
11/4/2021, 3:47:54 PM	KMIP Response: Create SymmetricKey f6cba7fa-d2e2-4741-b0ef-beb469bea1f5 Success
11/4/2021, 3:47:53 PM	KMIP Request: Create SymmetricKey
11/4/2021, 3:44:58 PM	KMIP Response: Create SymmetricKey 5f15bac9-e410-4c0d-a32d-fbd7839e79e6 Success
11/4/2021, 3:44:58 PM	KMIP Request: Create SymmetricKey
11/4/2021, 3:44:58 PM	KMIP Response: Create SymmetricKey 8464d07a-4d86-459a-b28b-8f22c356367d Success
11/4/2021, 3:44:57 PM	KMIP Request: Create SymmetricKey
11/4/2021, 3:44:57 PM	KMIP Response: Create SymmetricKey 4fa8d89c-c22c-4d3c-bd8d-c0a05065e74e Success
11/4/2021, 3:44:56 PM	KMIP Request: Create SymmetricKey
11/4/2021, 3:44:55 PM	KMIP Response: Create SymmetricKey 2389c35b-330a-481f-8af2-d0b925d5d72c Success
11/4/2021, 3:44:55 PM	KMIP Request: Create SymmetricKey
11/4/2021, 1:48:43 PM	KMIP Response: Create SymmetricKey b9639c96-3cd5-4d7f-82ee-f30741388d2 Success
11/4/2021, 1:48:43 PM	KMIP Request: Create SymmetricKey
11/4/2021, 1:48:42 PM	KMIP Response: Create SymmetricKey 6407fb89-8359-4e2a-9582-61e639e9a8d2 Success
11/4/2021, 1:48:42 PM	KMIP Request: Create SymmetricKey
11/4/2021, 1:48:42 PM	KMIP Response: Create SymmetricKey 289a2a3b-efad-4d34-b927-2c0604f57464 Success

- Go to the **Audit Log** page in KeyControl and validate the keys that were created when you created the storage domain in Cohesity. For example:

The screenshot shows the KeyControl interface with the 'Audit Log' page selected. The table below lists 10 records, all showing 'System' as the user performing actions related to SymmetricKey creation.

Date	User	Message
11/4/2021, 3:47:54 PM	System	KMIP Response: Create SymmetricKey f6cba7fa-d2e2-4741-b0ef-beb469bea1f5 Success
11/4/2021, 3:47:54 PM	System	KMIP Request: Create SymmetricKey
11/4/2021, 3:44:58 PM	System	KMIP Response: Create SymmetricKey 5f15bac9-e410-4c0d-a32d-fbd7839e79e6 Success
11/4/2021, 3:44:58 PM	System	KMIP Request: Create SymmetricKey
11/4/2021, 3:44:58 PM	System	KMIP Response: Create SymmetricKey 8464d07a-4d86-459a-b28b-8f22c356367d Success
11/4/2021, 3:44:57 PM	System	KMIP Request: Create SymmetricKey
11/4/2021, 3:44:57 PM	System	KMIP Response: Create SymmetricKey 4fa8d89c-c22c-4d3c-bd8d-c0a05065e74e Success
11/4/2021, 3:44:56 PM	System	KMIP Request: Create SymmetricKey

## 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_1  
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
  newscrite         : 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
```

## 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.