COHESITY

Version 1.1

July 2024

Use Vormetric DSM to Manage Cohesity Encryption Keys

Manage Cohesity Platform KEKs with Vormetric Data Security Manager

ABSTRACT

To ensure the security of your data, both in flight and at rest, Cohesity supports AES-256 software encryption using an internal Key Management Service (KMS) that automatically generates keys and stores them internally. However, if you manage your encryption keys in a centrally managed external KMS like Vormetric DSM, you can configure Cohesity Platform to use that instead. Use this guide to set up Vormetric DSM as your external KMS in Cohesity.

Table of Contents

Introduction to KEK Management

As organizations work to address a stream of security threats to the data they manage and store, encryption becomes more critical to every aspect of their cyber defense strategies. Among its many security features, Cohesity supports the use of an external Key Management System (KMS). This guide gives you the information and procedures you need to configure and integrate Cohesity Platform™ and Vormetric® Data Security Manager (DSM), the KMS from Thales.

When used with an external key manager like Vormetric DSM, the Cohesity cluster requests that the key manager creates a Key Encryption Key (KEK) for each Storage Domain in the cluster. These KEKs are used to encrypt and decrypt the Data Encryption Keys (DEKs) that are created and stored locally in the cluster. As the name implies, the DEKs are used to encrypt and decrypt the data itself.

The Cohesity cluster reaches out to Vormetric DSM to retrieve the KEKs after the system reboots or the keychain service restarts. If Vormetric DSM is unavailable, the data in the Storage Domains remains encrypted and inaccessible.

KMIP and Certificate Requirements

The Key Management Interoperability Protocol (KMIP) is used to facilitate communication between the Cohesity cluster and the key server on Vormetric DSM. As KMIP requires the use of Transport Layer Security (TLS) to provide a secure connection. X.509 certificates must exist for both the key server and the Cohesity cluster. When you install Vormetric DSM, the key server certificate is automatically generated and signed by the internal Certificate Authority (CA). You will need to create a client certificate for Cohesity using a tool like OpenSSL. This certificate may be self-signed or externally signed, but this characteristic must be a system-wide characteristic within the Vormetric DSM environment. In other words, all KMIP clients connecting to Vormetric DSM must either use self-signed certificates or externally-signed certificates—they cannot use a mixture of both.

Prerequisites

The procedures in this guide assume a basic knowledge of Vormetric DSM concepts and licensing

requirements for KMIP functionality, as well as:

- Cohesity version 6.3.1 or later is installed and operational, and the cluster is configured to use encryption. You can only enable encryption at cluster level when you create the Cohesity cluster. See Setup Guide for your specific cluster model in the online Help.

  NOTE: When encryption is enabled at the cluster level in Cohesity Platform, it cannot be disabled later.

- Vormetric DSM version 6.3.0.
- Vormetric DSM is installed and operational and is accessible by the Cohesity cluster on port 5696. For releases before Cohesity version 6.5, use the linux ‘firewall command’ to enable firewall ports on each node. For 6.5 and later, see Manage Application-Based Firewall Rules in the online Help.
- You know your organization's security policies governing data-at-rest encryption.

- You have access to OpenSSL or some other mechanism for generating a client certificate and private key in the Privacy Enhanced Mail (PEM) format on a Linux machine.

Encryption and Configuration Considerations

The following are some key points to understand regarding this integration:

- Once you enable encryption on a Cohesity cluster, you cannot disable it.
- Once you configure a Cohesity cluster to use an external KMS, you cannot return to using the internal KMS.
- With encryption enabled at the cluster level, all Storage Domains in the cluster are encrypted, by rule.
- The Cohesity cluster supports the configuration of one external KMS, and the IP address of the KMS cannot be altered once configured.
- Once it establishes a TLS connection with Vormetric DSM, a Cohesity cluster never tears down that connection. This results in persistent TLS connections.

Set Up Cohesity-Vormetric Integration

There are several tasks involved to deploy the Cohesity integration with Vormetric DSM:

- Use OpenSSL to create a private key and client certificate for the Cohesity cluster.
- Use OpenSSL to extract the internal root CA certificate from Vormetric DSM.
- Configure Vormetric DSM to support the integration, including the creation of a host for the Cohesity cluster. The client certificate will be imported into the host.
- Configure the Cohesity cluster to use Vormetric DSM as its external Key Management System (KMS).

Create the Client Certificate and Private Key

The first step is to create a client certificate and private key for Cohesity that you will use when you configure Vormetric DSM with the Cohesity cluster. You can use OpenSSL or any similar tool to create them, as long as the certificate and key files are in the PEM format, and they are created on a Linux machine.

There are two different types of client certificates:

- **Self-signed certificates**. If your security policy allows for it, you can generate and sign your client certificate yourself.
- **Externally-signed certificates**. If your security policy calls for an external Certificate Authority (CA), use this option. To use it, you must have:
  - Vormetric DSM configured to support system-wide use of externally signed certificates.
  - Established the external CA as trusted by importing the necessary CA certificates.
  - An external, trusted CA that is available to sign the Client Signing Request (CSR).

NOTE: The certificate files need to be available to the machine where you plan to log in to Cohesity Platform to configure the KMS settings.

Generate a Self-Signed Certificate and Private Key

To generate a self-signed certificate and private key for Cohesity:

1. Log in to the Cohesity Command Line Interface (CLI).
2. Use the `genrsa` command to generate the private key that will be written to the key filename and length you specify.

```
$ openssl genrsa -out <key_file_name> <key_length>
```

NOTE: Key lengths less than 2048 bits are not secure.

3. Enter the following OpenSSL command to create the self-signed certificate per your security policy.

```
$ openssl req -new -x509 -key <key_file_name> -out <cert_file_name> -days
<validity_period>
```

4. Enter the requested information when prompted by OpenSSL:
   - **Country Name**. Your two-letter country code.
   - **State or Province Name**. The full name of your state.
   - **City**. The full name of your city.
   - **Organization**. The name of your organization.
   - **Organizational Unit**. The name of your department.

- **Common Name**. The name that must match the name of the host created on Vormetric DSM.
- **Email**. (*Optional*) Your email address.
- **Challenge password**. (*Optional*) Leave this blank; click **Enter**.
- **Company name**. (*Optional*) Leave this blank; click **Enter**.

5. When the process completes, you can find the generated file in the current directory.

```
[cohesity@virtual-robo-esx ~]$ openssl req -new -x509 -key client_key.key -
out client_cert.crt -days 365
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []: California
Locality Name (eg, city) [Default City]:San Jose
Organization Name (eg, company) [Default Company Ltd]:Cohesity
Organizational Unit Name (eg, section) []:Engineering
Common Name (eg, your name or your server's hostname) []:dsm.cohesity.com
Email Address []:
[cohesity@virtual-robo-esx ~]$
```

6. You must generate the file on a Linux machine, and if you're creating it manually, be careful that the file doesn't contain any extraneous characters.

```
[cohesity@virtual-robo-esx ~]$ cat client_cert.crt
-----BEGIN CERTIFICATE-----
MIIDxTCCAq2gAwIBAgIJAOSurkIpu6d4MA0GCSqGSIb3DQEBCwUAMHkxCzAJBgNVBAYTAIVTMRMw
EQYDVQQIDApDYWxpZm9ybmIhMREwDwYDVQQHDAhTYW4gSm9zZTERMA8GAlUECgwIQ29oZXNpdHkx
FDASBgNVBAsMC0VuZ2IuZWVyaW5nMRkwFwYDVQQDDBBkc20uY29oZXNpdHkuY29tMB4XDTIwMDQx
NzE3MDcwNVoXDTIxMDQxNzE3MDcwNVoweTELMAkGAlUEBhMCVVMxEzARBgNVBAgMCkNhbGImb3Ju
aWExETAPBgNVBAcMCFNhbiBKb3NIMREwDwYDVQQKDAhDb2hIc2I0eTEUMBIGAlUECwwLRW5naW5I
ZXJpbmcxGTAXBgNVBAMMEGRzbS5jb2h!c2I0eS5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQDMwMq7EW+ikiKZTyhvR0G2+/lJTy0HrI6bzyo3PRDXC+Mb67yX3sCZDuAodY5bblXi
yRoZrxFjwHi6xu0+mYedPhMFHSdo9CPQF2gx2GfWZtEBFH0qZFZpXug3evqz2IOKBj8SLtFto4uK
xSE9EfkMM/IezcP8JVsW7cERohuqc38VBuLU7LH52vKH+oY0P5v0vpp8IPMJ2zB5vVSh97NTiF/Y
o4sWBxeLBQfGmQjxLqkpzSKGt7T3cle5d4AgPgITzLs6ia+XGQHvoZyIuYnD797hgoJfJZ2R82b8
zJRms7IGet8TXn!cic6+067It2z8fSveyBfRnDyphyWvKhpnAgMBAAGjUDBOMBOGAlUdDgQWBBS7
4xB8ZVcniPRZgErlx601kO9+8jAfBgNVHSMEGDAWgBS74xB8ZVcniPRZgEr1x601kO9+8jAMBgNV
HRMEBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQByOZndS7Y8tG2a2HAzPZB6p9DBDvpNGtT+kpXS
jh5uoU4PdB2//k3zGr9UIE8TYBFbLlTv!tMn908EVcmlbSYjnK+4k9LWOd0PWsLn4a9mFiTvIiba
tQ3CkjYZAv8jg2y6GI01s6wCxvWun70jkTxjf2S4bT0ioI+jPp/p0d9oafa+AWAcIMieEm2Gh7n7
5dZRWQ6Ne7XIxKk61kASKBiurnzRSvgaCBTTcfTG0fVhZWE7FFAf0C2vAMEFgiI8H65Lgw9f4Lmo
dngrIrD3JQms93QymIeyoF+wwrDsl0nOxONrgJ3oAEH+VdW37YSCR3WeTq46FW2e5wBVA5TJIr9z
-----END CERTIFICATE-----
[cohesity@virtual-robo-esx ~]$
```

Generate an Externally Signed Certificate and Private Key

If you choose to use an externally signed certificate instead of the self-signed certificate, you need to do so with an external and trusted Certificate Authority.

To generate an externally signed certificate and private key:

1. Log in to the Cohesity Command Line interface (CLI).
2. Use the `genrsa` command to generate the private key that will be written to the key filename and length you specify. Create the key per your organization's security policy.

```
$ openssl genrsa -out <key_file_name> <key_length>
```

For example:

3. Enter the following OpenSSL command to initiate the CSR file generation process.

```
$ openssl req -new -key <key_file-name> -out <csr_file_name>
```

4. Enter the requested information as prompted by OpenSSL:
   - **Country Name**. Your two-letter country code.
   - **State or Province Name**. The full name of your state.
   - **City**. The full name of your city.
   - **Organization**. The name of your organization.
   - **Organizational Unit**. Name of your department
   - **Common Name**. The name that must match the name of the host created on Vormetric DSM.
   - **Email**. (*Optional*) Your email address.
   - **Challenge password**. (*Optional*) Leave this blank; click **Enter**.
   - **Company name**. (*Optional*) Leave this blank; click **Enter**.

5. When the process completes, you can find the generated CSR file in the current directory.

```
[cohesity@virtual-robo-esx ~]$ openssl req -new -key cohesity241.key -out
cohesity241.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []: California
Locality Name (eg, city) [Default City]:San Jose
Organization Name (eg, company) [Default Company Ltd]:Cohesity
Organizational Unit Name (eg, section) []:Engineering
Common Name (eg, your name or your server's hostname) []:dsm.cohesity.com
Email Address []:
[cohesity@virtual-robo-esx ~]$
```

6. Have a trusted CA sign the CSR and download or create a file that contains the certificate.

```
[cohesity@virtual-robo-esx -]$ vi DSM_root.crt
[cohesity@virtual-robo-esx -]$ cat DSM_root.crt
-----BEGIN CERTIFICATE-----
MIIENjCCAx6gAwIBAgIGALOmLIL9MA0GCSqGSIb3DQEBDAUAMIGjMSQwIgYDVQQDExtDRyBDQ
SBTIG9uIGRzbS5jb2hlc2l0eS5jb20xETAPBgNVBAsTCGNvaGVzaXR5MREwDwYDVQQKEwhjb2
hlc2l0eTEQMA4GAlUEBxMHU2FuSm9zZTELMAkGAlUECBMCQ0ExKTAnBgkqhkiG9w0BCQEWGmZ
lcWlhbmcuemhhbmdAY29oZXNpdHkuY29tMQswCQYDVQQGEwJVUzAeFw0yMDA0MTUyMzQ5MTha
Fw0zMDA0MTcyMzQ5MThaMIGjMSQwIgYDVQQDExtDRyBDQSBTIG9uIGRzbS5jb2hlc2l0eS5jb
20xETAPBgNVBAsTCGNvaGVzaXR5MREwDwYDVQQKEwhjb2hlc2l0eTEQMA4GAlUEBxMHU2FuSm
9zZTELMAkGAlUECBMCQ0ExKTAnBgkqhkiG9w0BCQEWGmZlcWlhbmcuemhhbmdAY29oZXNpdHk
uY29tMQswCQYDVQQGEwJVUzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAL5BAxvn
dW/VoHfpMQPFqw9nzvxddT3yfypbl07TZOeby7C2XHhuIneUzPPr7m/vC8cTS0eqj75cj0SVw
VkMzKiU2nHXxX3qAztnkHhlo5uryWsQk2Kls0fB3GecvjU25oDo9NOJqDoU29wKhjgZMAMv/S
PIHYu1IfWT0CV5FfjtsdsDYINHiLdUxZIQJF42r0aPQT0weiTG0V9bJj/ilA2hCUTyVVwXc4e
rEsDbeT9Sc7kKJ3QEGcBMcVtit5zMAK5HhMq8T+oS9dhIyWBZpiAVp/rYE2ZfIU2FeplHJtf6
0zCpIgHr7N12lqviTbsKsvhyLt20teadRitSRlXGweMCAwEAAaNuMGwwEgYDVR0TAQH/BAgwB
gEB/wIBADAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFKgZMHRJBV1MRLi+xHdp0vrjsQSwMC
cGAlUdIwQgMB6AFKgZMHRJBV1MRLi+xHdp0vrjsQSwggYAs6Ysgv0wDQYJKoZIhvcNAQEMBQA
DggEBAEsQ29HfaHpHUFBj4Tn74XW0b9w9Pmk2u5AM13d9CU/oLYcMJn0LtfavpnMgKPuDwQ/3
gqgFDXiq40dq73fD+38A2wlcB/xZumQ6nGGuYGk2p2VVy6CrJlVD0pL4iwidQXMYJ7xDmL+Tw
HM8clwPiwxqI3HRNRQokqwl489VeptNMteInllMIY28TZOn/T89OxUDxoqxIlXEu672qZq7kG
/YpeeGG+0uwH8QZnxjogaIQPulTxJs5tzcsFM7nRjE63Z14GUIsQElpXikSu9G8ljpGBmkfcJ
daVIHSeRTB7NV5ZIj2NvAUXm/RFvjTxh9blR5yTkdPUDEDgFdJWVA6Fw=
-----END CERTIFICATE-----
[cohesity@virtual-robo-esx -]$
```

Extract the Vormetric DSM Internal CA Certificate

Once you have generated the client certificate and private key, you need to extract the Vormetric DSM internal root CA certificate for import into the Cohesity cluster. You can extract it via web browser on a Linux machine or by using OpenSSL, as below.

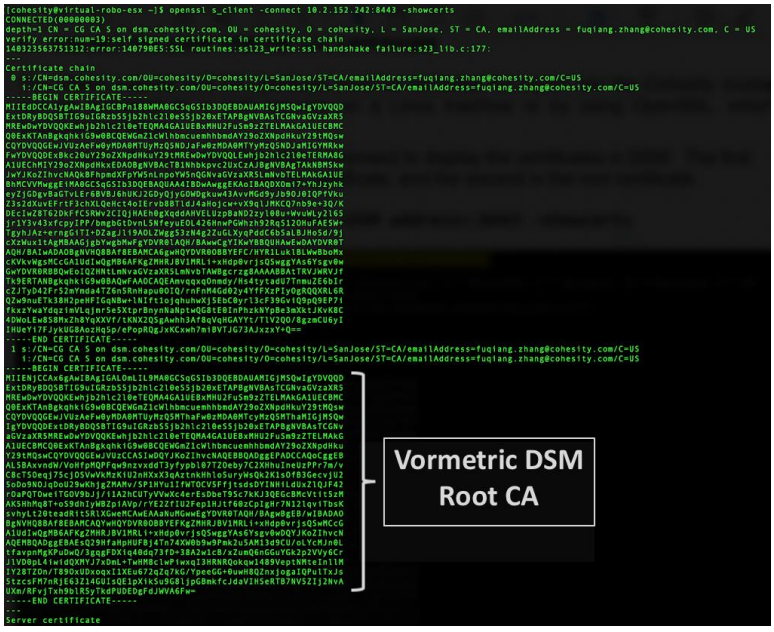
To extract the Vormetric DSM internal root CA certificate:

1. Log in to the Cohesity CLI.

2. Enter the following OpenSSL command to display the certificates in Vormetric DSM. The first certificate is the server certificate and the second is the root certificate.

```
$ openssl s_client -connect <DSM_address>:8443 -showcerts
```

   For example:



3. Carefully copy the second certificate in the command output (that is, the self-signed root CA certificate).

4. Using vi or another Linux utility, create a file and paste the certificate into it. Be careful not to include any extraneous characters.

```
[cohesity@virtual-robo-esx -]$ vi DSM_root.crt
[cohesity@virtual-robo-esx -]$ cat DSM_root.crt
-----BEGIN CERTIFICATE-----
MIIENjCCAx6gAwIBAgIGALOmLIL9MA0GCSqGSIb3DQEBDAUAMIGjMSQwIgYDVQQDExtDRyBDQS
BTIG9uIGRzbS5jb2hlc2l0eS5jb20xETAPBgNVBAsTCGNvaGVzaXR5MREwDwYDVQQKEwhjb2hl
c2l0eTEQMA4GAlUEBxMHU2FuSm9zZTELMAkGAlUECBMCQ0ExKTAnBgkqhkiG9w0BCQEWGmZlcW
lhbmcuemhhbmdAY29oZXNpdHkuY29tMQswCQYDVQQGEwJVUzAeFw0yMDA0MTUyMzQ5MThaFw0z
MDA0MTcyMzQ5MThaMIGjMSQwIgYDVQQDExtDRyBDQSBTIG9uIGRzbS5jb2hlc2l0eS5jb20xET
APBgNVBAsTCGNvaGVzaXR5MREwDwYDVQQKEwhjb2hlc2l0eTEQMA4GAlUEBxMHU2FuSm9zZTEL
MAkGAlUECBMCQ0ExKTAnBgkqhkiG9w0BCQEWGmZlcWlhbmcuemhhbmdAY29oZXNpdHkuY29tMQ
swCQYDVQQGEwJVUzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAL5BAxvndW/VoHfp
MQPFqw9nzvxddT3yfypbl07TZOeby7C2XHhuIneUzPPr7m/vC8cTS0eqj75cj0SVwVkMzKiU2n
HXxX3qAztnkHhlo5uryWsQk2Kls0fB3GecvjU25oDo9NOJqDoU29wKhjgZMAMv/SPIHYu1IfWT
0CV5FfjtsdsDYINHiLdUxZIQJF42r0aPQT0weiTG0V9bJj/ilA2hCUTyVVwXc4erEsDbeT9Sc7
kKJ3QEGcBMcVtit5zMAK5HhMq8T+oS9dhIyWBZpiAVp/rYE2ZfIU2FeplHJtf60zCpIgHr7N12
lqviTbsKsvhyLt20teadRitSRlXGweMCAwEAAaNuMGwwEgYDVR0TAQH/BAgwBgEB/wIBADAOBg
NVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFKgZMHRJBV1MRLi+xHdp0vrjsQSwMCcGAlUdIwQgMB6A
FKgZMHRJBV1MRLi+xHdp0vrjsQSwggYAs6Ysgv0wDQYJKoZIhvcNAQEMBQADggEBAEsQ29HfaH
pHUFBj4Tn74XW0b9w9Pmk2u5AM13d9CU/oLYcMJn0LtfavpnMgKPuDwQ/3gqgFDXiq40dq73fD
+38A2wlcB/xZumQ6nGGuYGk2p2VVy6CrJlVD0pL4iwidQXMYJ7xDmL+TwHM8clwPiwxqI3HRNR
Qokqwl489VeptNMteInllMIY28TZOn/T89OxUDxoqxIlXEu672qZq7kG/YpeeGG+0uwH8QZnxj
ogaIQPulTxJs5tzcsFM7nRjE63Z14GUIsQElpXikSu9G8ljpGBmkfcJdaVIHSeRTB7NV5ZIj2N
vAUXm/RFvjTxh9blR5yTkdPUDEDgFdJWVA6Fw=
-----END CERTIFICATE-----
[cohesity@virtual-robo-esx -]$
```

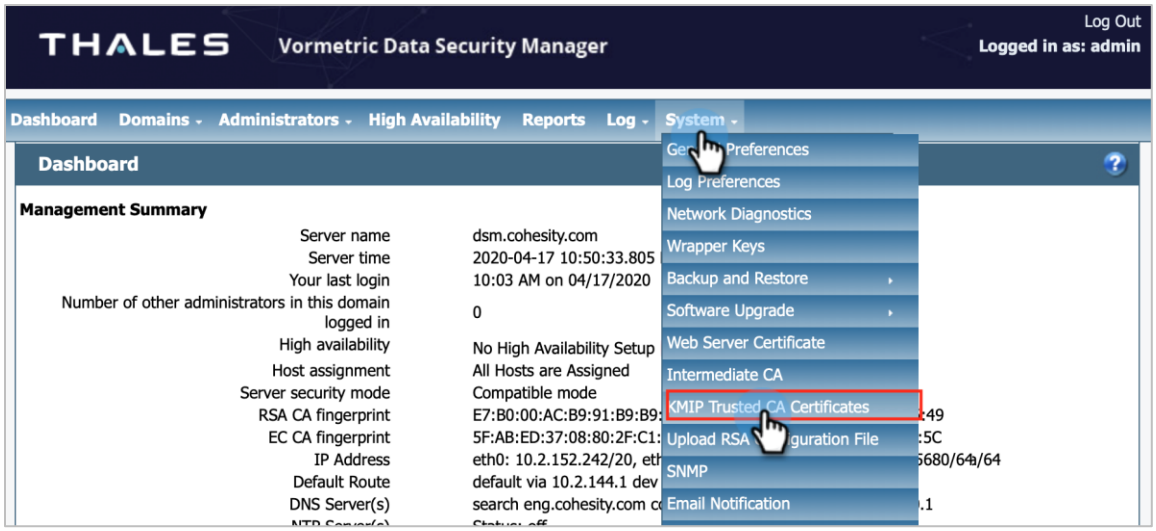
As necessary, transfer the CA certificate file, the client private key file, and the client certificate file to the machine that will be used to authenticate Cohesity Platform and Vormetric DSM to perform the configuration steps.

Configure Vormetric DSM Settings

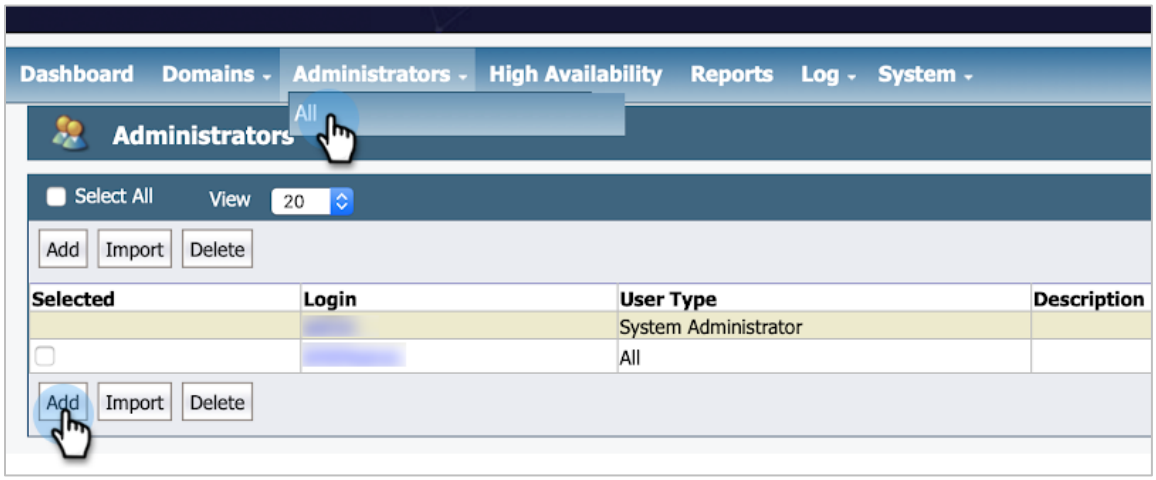
Equipped with your client certificate, private key, and the Vormetric DSM internal CA certificate, you are ready to configure the Vormetric DSM settings to support Cohesity for a Key Management Interoperability Protocol integration.

To configure Vormetric DSM:

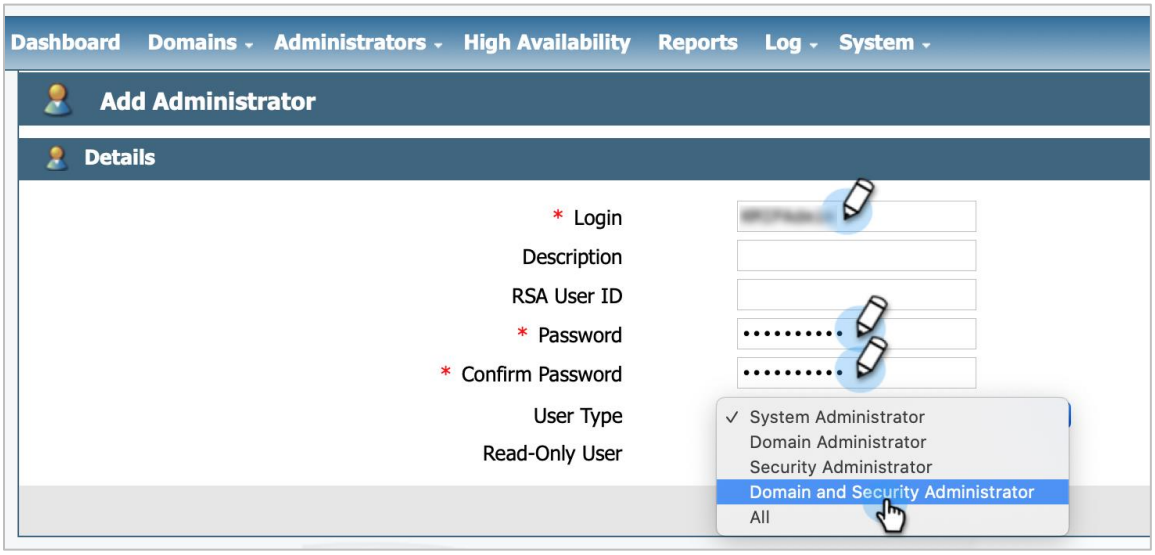
1. Log in to your account on Vormetric.
2. If you are using externally-signed client certificates for KMIP clients, navigate to **System > KMIP Trusted CA Certificates** to import the CA certificate(s) into Vormetric DSM.



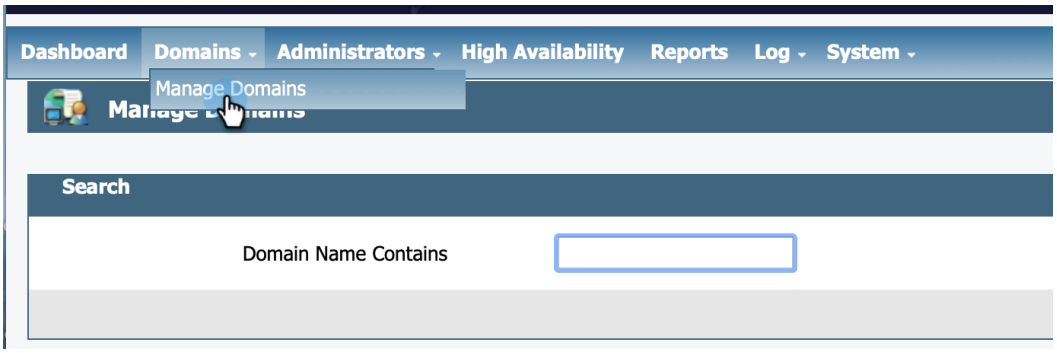
3. Create a “Domain and Security administrator” or “All” if it does not already exist. To do so, navigate to **Administrators >All** and click **Add**.



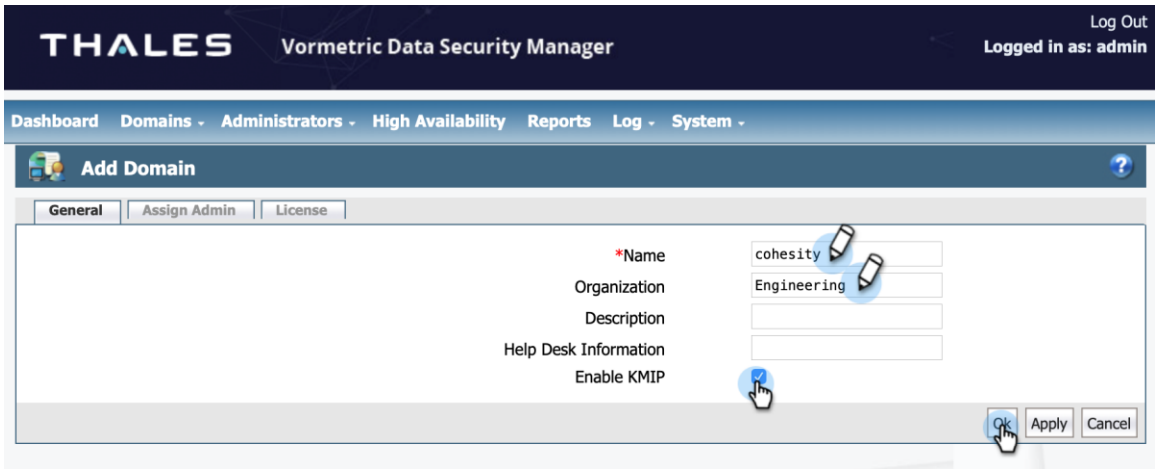
4. Enter the **Login** name, **Password** (twice), and select **User Type** 'Domain and Security Administrator' or 'All.'



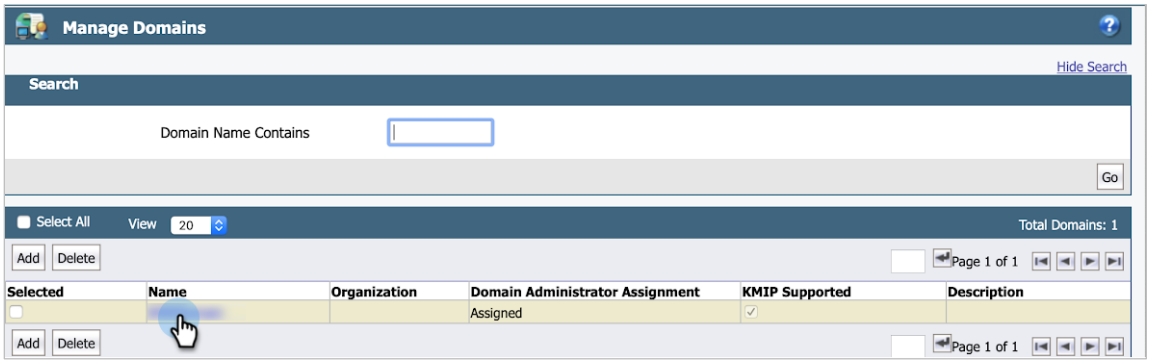
5. If you already have a Cohesity domain in Vormetric DSM, you can use that. If not, you can add a new domain for Cohesity via the **Domains > Manage Domains.**



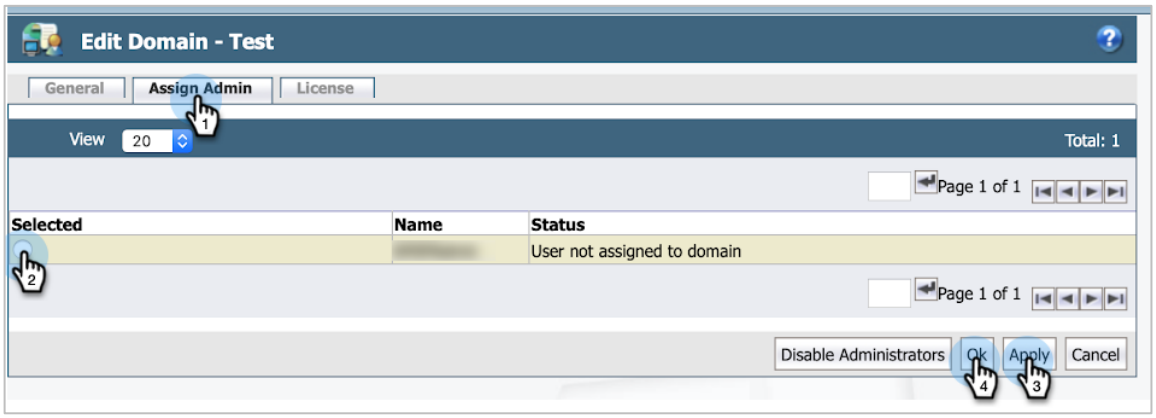
6. Provide **Name** and **Organization.** Check **Enable KMIP** and click **OK**.



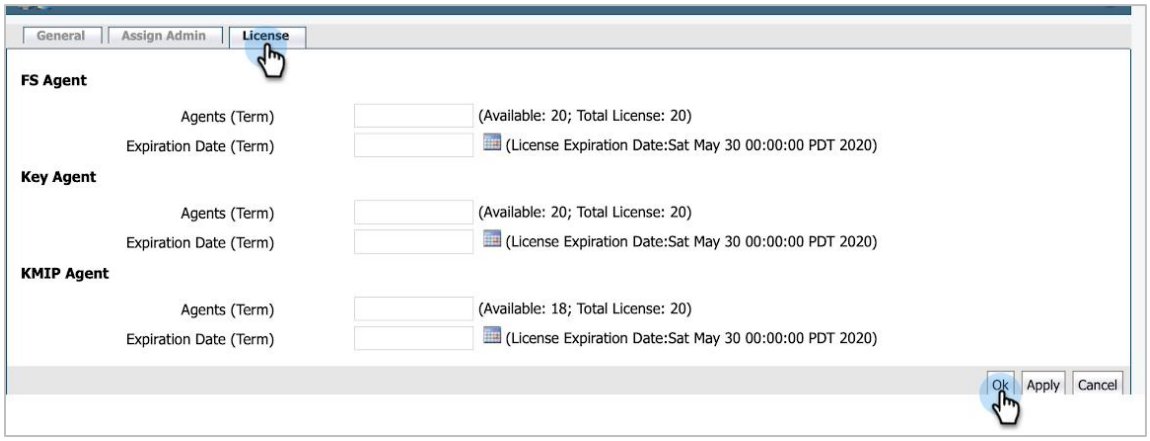
7. Click the domain that you just created.



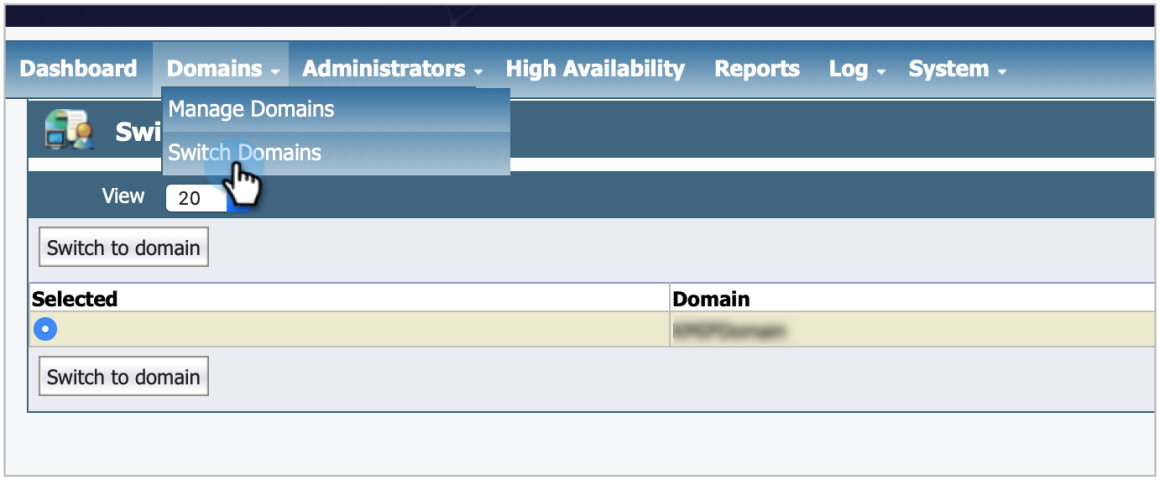
8. Click the **Assign Admin** tab. Click the radio button next to the domain to assign a 'Domain and Security Administrator' user type. Click **Apply** and then **Ok**.



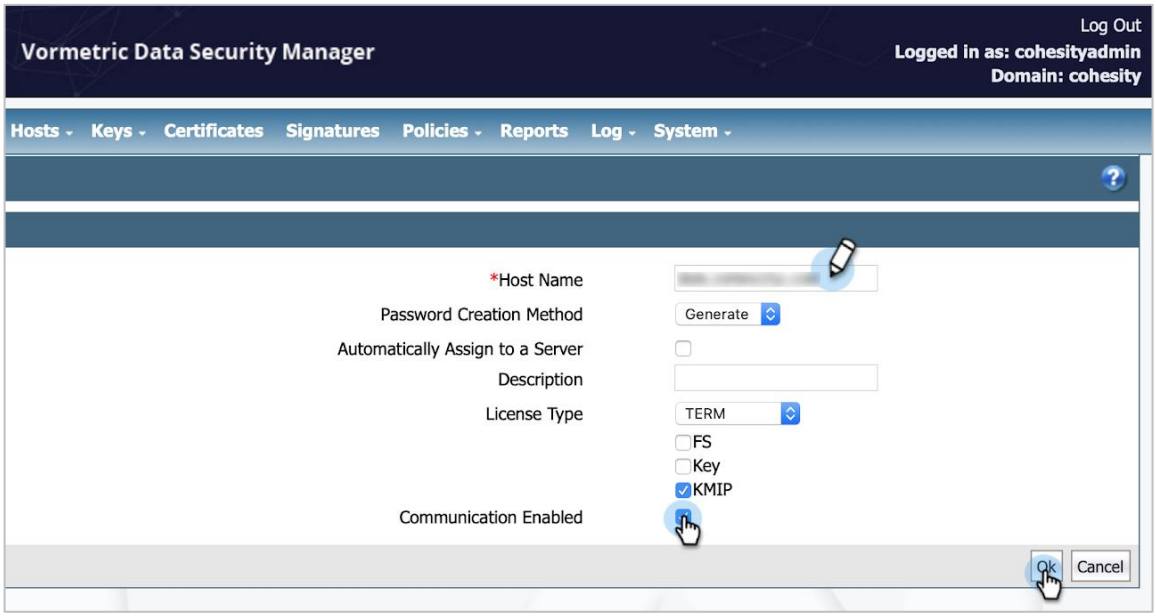
9. Click the **License** tab to make license assignments for KMIP.



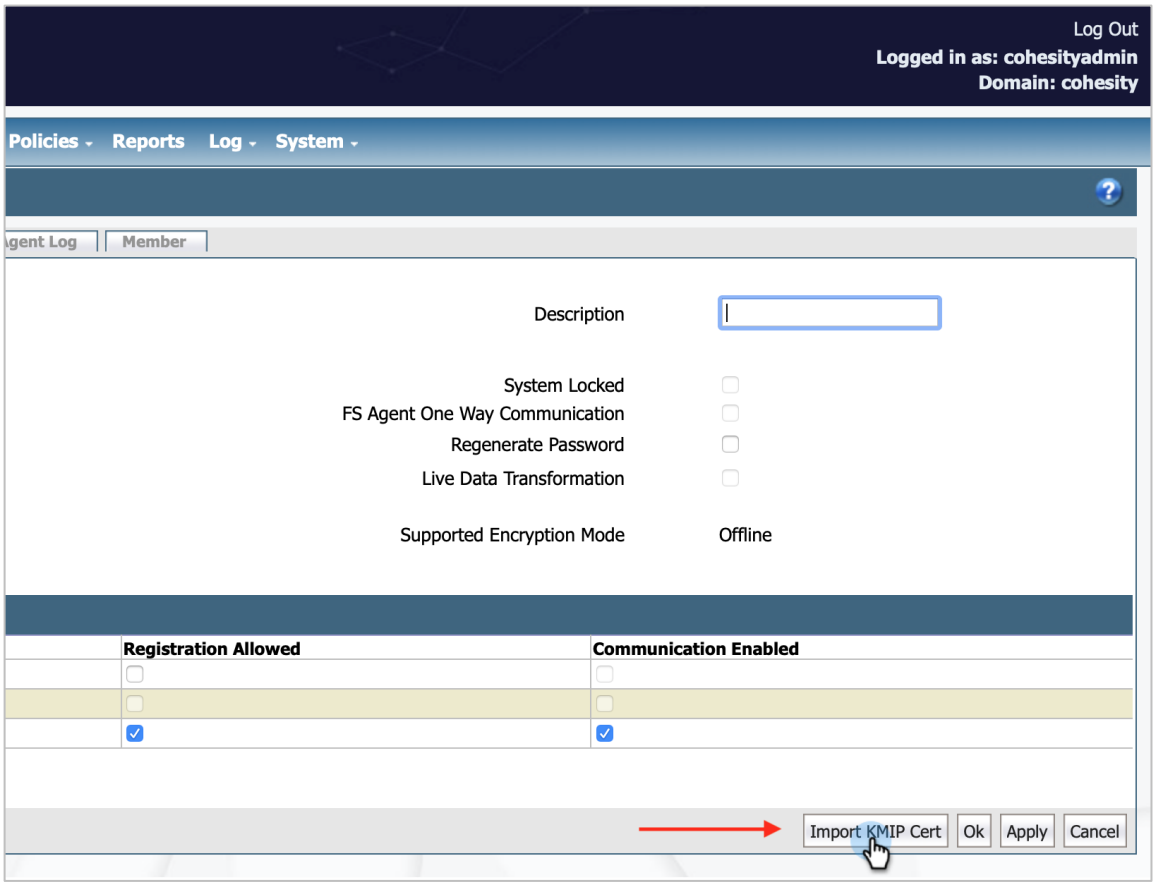
10. Switch into the domain that is to contain the Cohesity cluster.



11. Add a host in the selected domain using the Common Name of the Cohesity cluster that will be put in the client certificate.



12. After adding the host, click **Import KMIP Cert** to import the client certificate you created for the Cohesity cluster.



13. Vormetric DSM is now configured to support KMIP communication with the Cohesity cluster. You are ready to configure the KMS settings in Cohesity Platform.

Configure Cohesity Key Management Settings

The final step is to configure the KMS settings on the Cohesity cluster. Once the configuration has been saved, the cluster immediately establishes a TLS session with Vormetric DSM. After a few minutes of internal processing, it then requests that keys be created for internal use and for the existing Default Storage Domain.

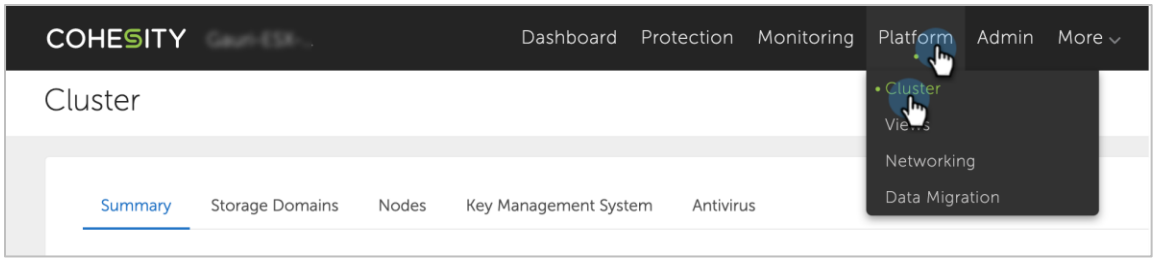
> **NOTE**: The Vormetric DSM IP address cannot be changed once it is configured.

You can configure Vormetric DSM in the Cohesity Platform browser UI or in the Cohesity Command Line Interface (CLI).

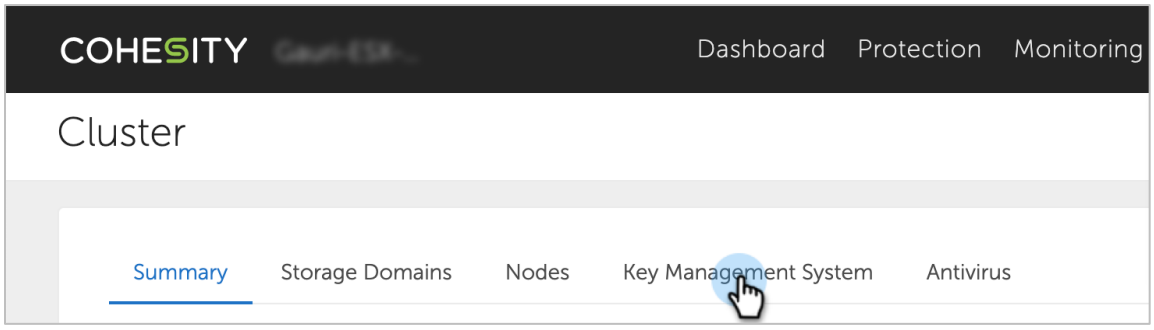
Configure Vormetric DSM in Cohesity Platform UI

To configure Vormetric DSM in the Cohesity UI:

1. Log in to Cohesity Platform.
2. Navigate to **Platform > Cluster**.



3. In the **Cluster Summary** page, click the **Key Management System** tab.

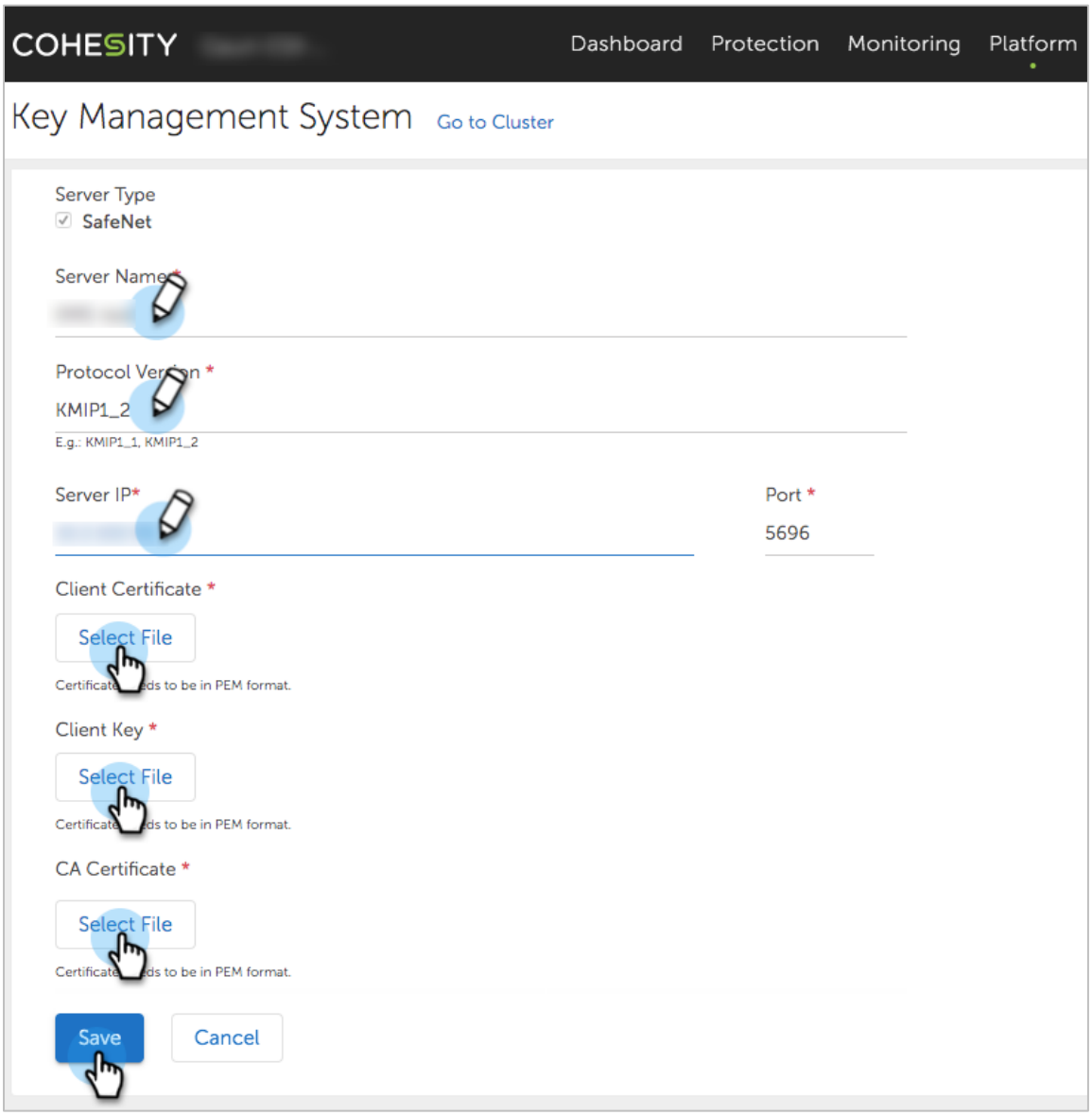


4. In the **Key Management System** form, select or provide:

   - **Server Type**. This setting cannot be modified and is OK as set.
   - **Server Name**. A name of your choosing for identifying Vormetric DSM.
   - **Protocol Version**. The version of KMIP to be used.

     NOTE: Acceptable options are: KMIP1_1, KMIP1_2, or KMIP1_3. In our testing, Cohesity used KMIP1_2.

   - **Server IP**. Vormetric DSM's IP address.
   - **Client Certificate**. Select the client certificate file that you created above.
   - **Client Key**. Select the private key file that you created.
   - **CA Certificate**. Select the CA certificate file that you extracted.



5. Click **Save**.

6. The Cohesity cluster immediately attempts to establish a TLS session with Vormetric DSM and initiate KMIP communication. If you get a **KMS Validation Error** at this point, see Troubleshooting below.

Configure Vormetric DSM in Cohesity Platform CLI

You can also configure Vormetric DSM in the Cohesity Command Line Interface. See Using the Cohesity Platform CLI in the online Help for the full list of commands.

To configure Vormetric DSM in the Cohesity Platform CLI:

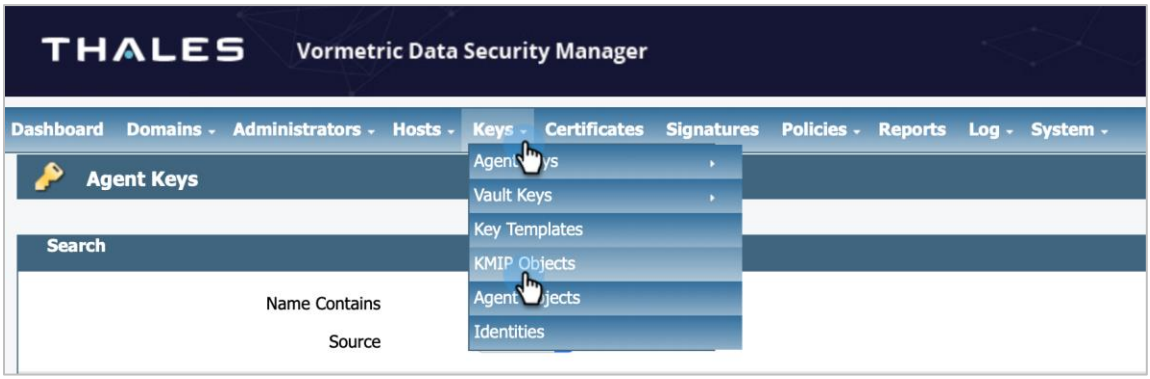
1. SSH to the cluster using following command:

```
ssh cohesity@<ip address of node>
```

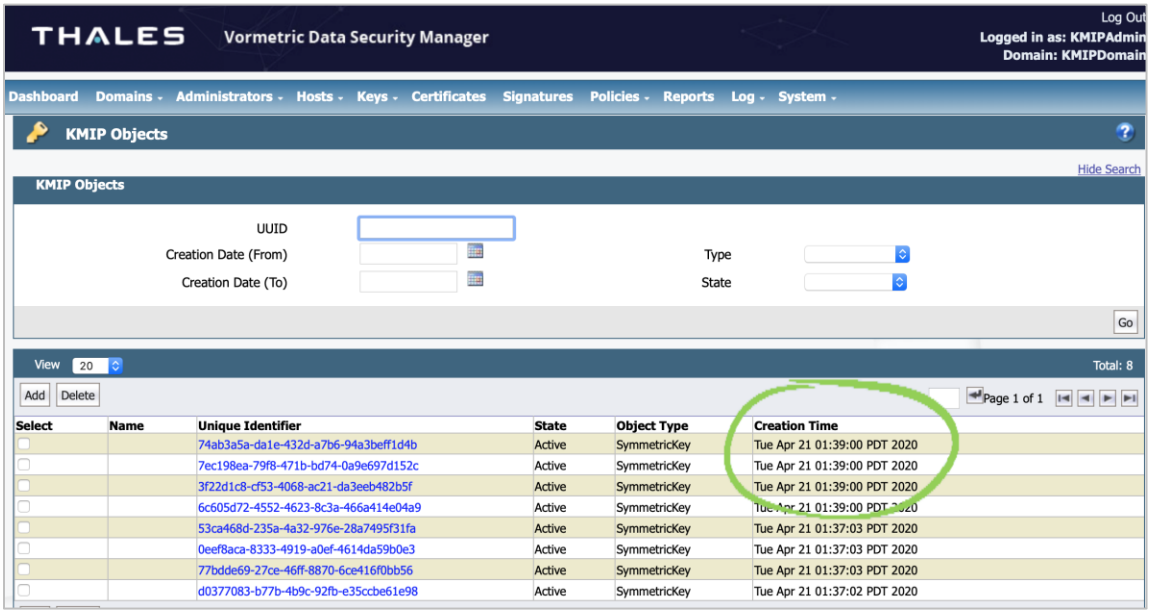
2. In the CLI, use the **kms create** command.

```
admin@x.x.x.x> kms create help
  DESCRIPTION
    To create a new KMS.

Example: kms create ca-certificate="<absolute path to CA certificate>"
client-certificate="<absolute path to client certificate>" dien
t-key="<absolute path to client-key>" kmip-protocol-version="KMIPl_2"
kms-name="KmipServerl" kms-port=5696 kms-ip="IP address of KMS"

  PARAMS
    ca-certificate            [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.
    kmip-protocol-version     [string]  required      kmip-protocol-
version
    kms-ip              [string]  required      IP address of the KMS.
    kms-name                  [string]  required      Name of the KMS.
    kms-port                  [int]    required       KMS Port. Default
KMIP port is 5696.

admin@x.x.x.x>
```

3. After successfully connecting with Vormetric DSM, Cohesity automatically creates one key for a default Storage Domain and other keys for internal use. Allow a couple of minutes for processing, and then check the **Keys > KMIP Objects** page in Vormetric DSM to find signs of keys having just been created to validate the integration.



If you see the below keys, it means that the KEKs have migrated from your Cohesity cluster to Vormetric DSM. This completes the integration of Vormetric DSM with the Cohesity cluster.



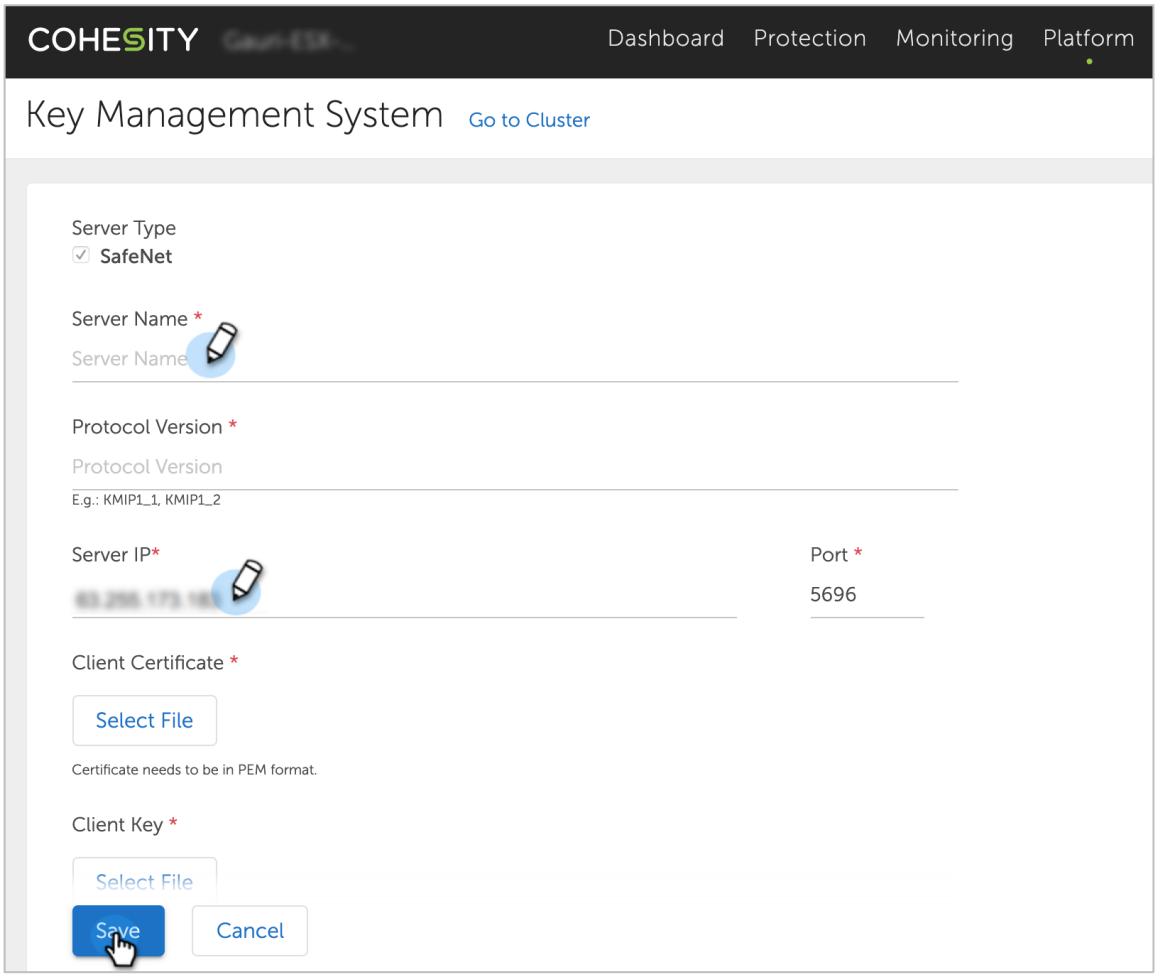
Update Cohesity Key Management Configuration

If your KMS configuration has changed because of expired certificates or a change in KMIP protocol version, you can update the Key Management System settings in Cohesity Platform. As with the initial Cohesity KMS configuration, you can update these in the Cohesity Platform browser UI or in the Cohesity Platform CLI.

Update Cohesity KMS in Cohesity Platform UI

To update the Cohesity KMS settings using the browser UI:

1. Navigate to **Platform > Cluster** and click the **Key Management System** tab.
2. In the **Key Management System** page, you can edit the configured KMS details like **Server Name**, **Protocol Version**, **Server IP**, and the associated certificates . When you're done, click **Save**.



Update Cohesity KMS in Cohesity Platform CLI

You can also update the Cohesity KMS settings in the Cohesity Platform CLI. See Using the Cohesity Platform CLI in the online Help for the full list of commands.

To update the Cohesity KMS settings using the Cohesity Platform CLI:

1. SSH to the cluster using following command:

```
ssh cohesity@<ip address of node>
```

2. In the CLI, use the **kms update** command.

```
admin@x.x.x.x> kms update help

  DESCRIPTION
    To update an existing KMS.

  PARAMS

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

admin@x.x.x.x>
```

NOTES:

- If at any point after initial configuration you modify the Key Management System settings on the Cohesity cluster, you must manually restart the keychain service. See Keychain Service below.
- The KMS IP address cannot be modified once configured.

Troubleshooting

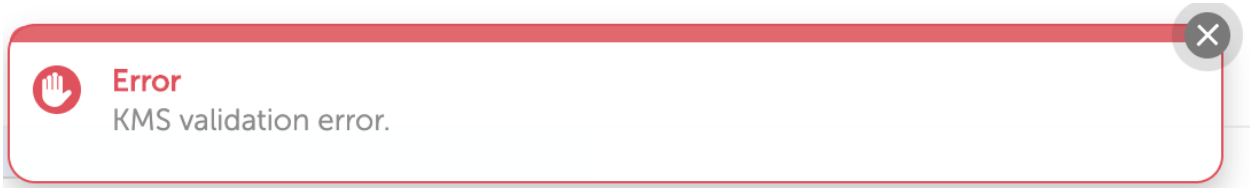
You might encounter errors while configuring KMS or Storage Domain settings in Cohesity Platform. The error might be caused by invalid input parameters or communications errors.

The most common errors are:

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

KMS Validation Error with KMS Configuration

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



If it does, take the following steps:

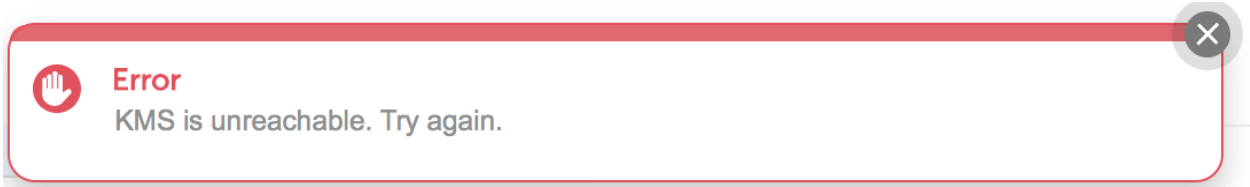
- Verify correct addressing and basic network connectivity between Vormetric DSM and the Cohesity cluster.
- Verify port 5696 is configured on the Cohesity Platform KMS settings page and that firewalls are open for that port.
- If any of the uploaded certificate files or private key file on the Cohesity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

  IMPORTANT: The Cohesity KMS client only accepts an SSL certificate 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.
- Verify that the CA certificate uploaded on the Cohesity Platform KMS settings page is the internal root CA certificate from Vormetric DSM. It should *not* be Vormetric DSM's server certificate. The Cohesity cluster needs the root CA certificate to validate the server certificate that is delivered to it while establishing a TLS session.
- If you are using an externally signed client certificate, make sure all certificates in the CA chain have been imported into Vormetric DSM via the **System > KMIP Trusted CA Certificates** page.
- If you are using a self-signed client, make sure there are no KMIP Trusted CA Certificates that have been imported into Vormetric DSM. If there are, you'll have to create an externally-signed client certificate that has been signed by one of those trusted CAs.
- Verify the host configuration on Vormetric DSM for the Cohesity cluster:

  - The **Host Name** must match the **Common Name** in the client certificate.
  - The client certificate must be uploaded into the host configuration on Vormetric DSM, and its **Certificate Fingerprint** should be visible on the host configuration page.
  - The KMIP **Registration Allowed** and **Communication Enabled** settings should be checked.
- Proper licensing must be in place.

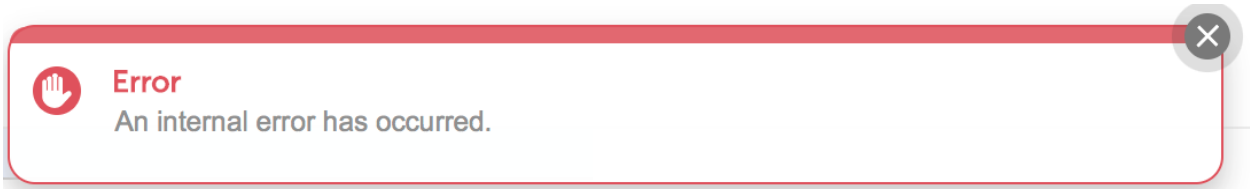
KMS Unreachable Error with Storage Domain Creation

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



A possible cause of this error is that the TLS session with Vormetric DSM has been dropped due to inactivity. The Cohesity cluster will immediately take action to re-establish the connection, but not before you see the ***KMS is unreachable*** error message. To remedy this, simply click the **Create Storage Domain** button to try again. If the problem was indeed just a dropped TLS session, it should work the second time.

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.



To resolve this, try the steps in KMS Validation Error above.

Keychain Service

The keychain service on Cohesity Platform is responsible for communicating with Vormetric DSM.

An error log for the keychain service exists in the `/logs` directory of the Cohesity instance. It is accessible by SSH'ing into the instance to access the Linux OS. The filenames are `keychain_exec.ERROR`, `keychain_exec.INFO`, and `keychain_exec.FATAL`.

Restart the Keychain Service after Key Management Settings Update

If you update the Key Management settings after initially configuring them, you have to restart the keychain service for the new settings to take effect.

To restart the Cohesity keychain service in the Cohesity Platform CLI:

1. Start the Cohesity Platform CLI remotely or locally using the following command;

```
$ iris_cli -server x.x.x.x -username=<username> -password=<password>
```

2. Issue the following command:

```
$ cluster restart service-names="keychain"
```

See Restart the Cohesity Keychain Service in the online Help for more.

Your Feedback

Was this document helpful? Send us your feedback!

About the Authors

Gauri Gokhale is Member of Technical Staff, Product Security, at Cohesity. In her role, Gauri focuses on enterprise data protection and software usability.

Other essential contributors included:

- Bart Abicht, Senior Technology Writer and Editor at Cohesity
- Fuqiang Zhang, Member of Technical Staff, Core Infrastructure Platform, at Cohesity

Document Version History

VERSION	DATE	DOCUMENT HISTORY
1.1	July 2024	Republishing
1.0	May 2020	First release

ABOUT COHESITY

Cohesity is a leader in AI-powered data security and management. Aided by an extensive ecosystem of partners, Cohesity makes it easier to protect, manage, and get value from data – across the data center, edge, and cloud. Cohesity helps organizations defend against cybersecurity threats with comprehensive data security and management capabilities, including immutable backup snapshots, AI-based threat detection, monitoring for malicious behavior, and rapid recovery at scale. Cohesity solutions are delivered as a service, self-managed, or provided by a Cohesity-powered partner. Cohesity is headquartered in San Jose, CA, and is trusted by the world's largest enterprises, including six of the Fortune 10 and 44 of the Fortune 100.

Visit our website and blog, follow us on Twitter and LinkedIn and like us on Facebook.

.