MongoDB and Entrust KeyControl Vault

Integration Guide

2024-04-19

Table of Contents

Chapter 1. Introduction

This document describes the integration of MongoDB with the Entrust KeyControl Vault Management Solution (KMS). Entrust KeyControl Vault can serve as a KMS in MongoDB using the open standard Key Management Interoperability Protocol (KMIP).

1.1. Documents to read first

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

To install and configure the Entrust KeyControl Vault server as a KMIP server, see the *Entrust KeyControl Vault nShield HSM Integration Guide*. You can access it from the Entrust Document Library and from the nShield Product Documentation website.

Also refer to the MongoDB online documentation.

1.2. Requirements

- Entrust KeyControl Vault version 10.2 or later

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

- MongoDB Enterprise Edition 7.0.6 or later

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

1.3. High-availability considerations

Entrust KeyControl Vault 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 Vault Product Overview.

1.4. Product configuration

The integration between the MongoDB Enterprise Edition and Entrust KeyControl Vault has been successfully tested in the following configurations:

Product	Version
MongoDB Enterprise Edition	7.0.6
Entrust KeyControl Vault	10.2
Red Hat Enterprise Linux 8.9 (Ootpa)	Kernel: Linux 4.18.0-513.18.1.el8_9.x86_64

Chapter 2. Procedures

2.1. Installation overview

1. Install the MongoDB Enterprise Edition.
2. Install and configure Entrust KeyControl Vault.
3. Configure MongoDB Server Security options to use KMIP.
4. Verify that the encryption is working and that MongoDB is using KeyControl to manage the keys.

2.2. Install the MongoDB Enterprise Edition

Installing the MongoDB depends on the operating system on which you are installing it. See the MongoDB documentation for details on how to install MongoDB in your environment. This guide used a RedHat Linux 8 installation. Follow the installation steps in the Install MongoDB Enterprise Edition on Red Hat or CentOS guide from the mongoDB official site.

1. By default, MongoDB instance stores files in the following locations:
   - Data files: `/var/lib/mongo`
   - Log files: `/var/log/mongodb`
2. MongoDB runs using the `mongodb` user account.

2.2.1. Validate the MongoDB installation

1. Start MongoDB:

```
% sudo systemctl start mongod
```

2. Verify that MongoDB has started successfully:

```
% sudo systemctl status mongod

● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2024-03-14 11:56:15 EDT; 19s ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 7834 (mongod)
   Memory: 75.7M
   CGroup: /system.slice/mongod.service
           └─7834 /usr/bin/mongod -f /etc/mongod.conf
```

```
Mar 14 11:56:15 mongodb-kc102-redhat-8 systemd[1]: Started MongoDB Database Server.
Mar 14 11:56:16 mongodb-kc102-redhat-8 mongod[7834]: {"t":{"$date":"2024-03-14T15:56:16.008Z"},"s":"I",
"c":"CONTROL",  "id":7484500, "ctx":"main","msg":"En>
```

3. Stop MongoDB:

```
% sudo systemctl stop mongod
```

4. Restart MongoDB:

```
% sudo systemctl restart mongod
```

5. View the log file used by MongoDB at `/var/log/mongodb/mongod.log`.

   You can follow the state of the process for errors or important messages by watching the output in the `/var/log/mongodb/mongod.log` file.

6. Begin using MongoDB.

   Start a `mongosh` session on the host machine where `mongod` is running. You can run `mongosh` without any command-line options to connect to a `mongod` that is running on your localhost with default port `27017`. For example:

```
% mongosh

Current Mongosh Log ID: 65f45f4b138c69e482b24d1d
Connecting to:
mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.5
Using MongoDB:          7.0.6
Using Mongosh:          2.1.5

For mongosh info see: https://docs.mongodb.com/mongodb-shell/


To help improve our products, anonymous usage data is collected and sent to MongoDB periodically
(https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting
   2024-03-14T11:56:16.710-04:00: Access control is not enabled for the database. Read and write access to
data and configuration is unrestricted
   2024-03-14T11:56:16.710-04:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest
setting it to 'never'
   2024-03-14T11:56:16.710-04:00: vm.max_map_count is too low
------

Enterprise test>
```

7. Create a test database:

```
% Enterprise test> use mydb1;
```

```
switched to db mydb1

% Enterprise mydb1> db

mydb1
```

8. Insert an entry into the database:

```
% Enterprise mydb1> db.movie.insertOne({"name":"tutorials point"})

{
  acknowledged: true,
  insertedId: ObjectId("60f73d77d932673cb91de215")
}
```

9. List the databases:

```
> Enterprise mydb1> show dbs

admin   40.00 KiB
config  12.00 KiB
local   72.00 KiB
mydb1   40.00 KiB
```

10. Drop the test database:

```
% Enterprise mydb1> db.dropDatabase()

{ ok: 1, dropped: 'mydb1' }
```

If you were able to create the database and perform the commands above, the installation is validated.

2.3. Install and configure Entrust KeyControl Vault

To install and configure Entrust KeyControl Vault, follow the installation and setup instructions in the *Entrust KeyControl Vault nShield HSM Integration Guide*. You can access it from the Entrust Document Library and from the nShield Product Documentation website.

MongoDB requires the KMIP server to support TLS 1.2. Make sure **Restrict TLS** is enabled when enabling KMIP at the Entrust KeyControl Vault Server.

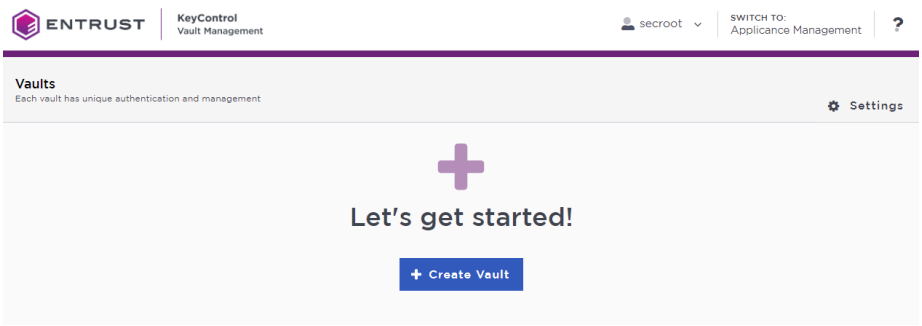
2.3.1. Creating a KMIP Vault on Entrust KeyControl Vault Management Interface

The KeyControl Vault appliance supports different type of vaults that can be used by all type of applications. This section describes how to create a KMIP Vault in the KeyControl Vault Server.

Refer to the Creating a Vault section of the admin guide for more details about it.

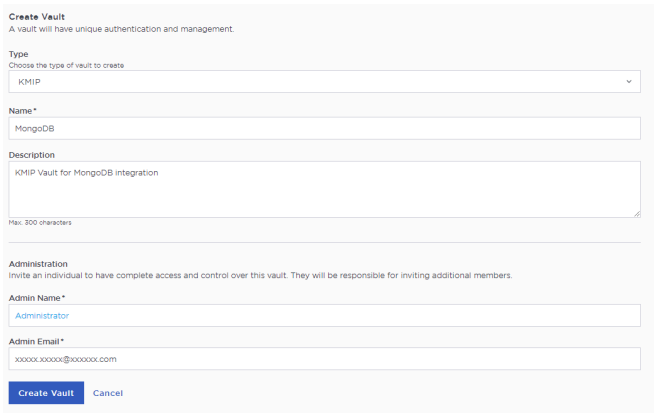
1. Log into the KeyControl Vault Server web user interface:
   a. Use your browser to access the IP address of the server.
   b. Sign in using the **secroot** credentials.
2. If not in the Vault Management interface, select **SWITCH TO: Manage Vaults** in the Menu Header.

   This action will take you to the KeyControl Vault Management interface.
3. In the KeyControl Vault Management interface, select **Create Vault**.



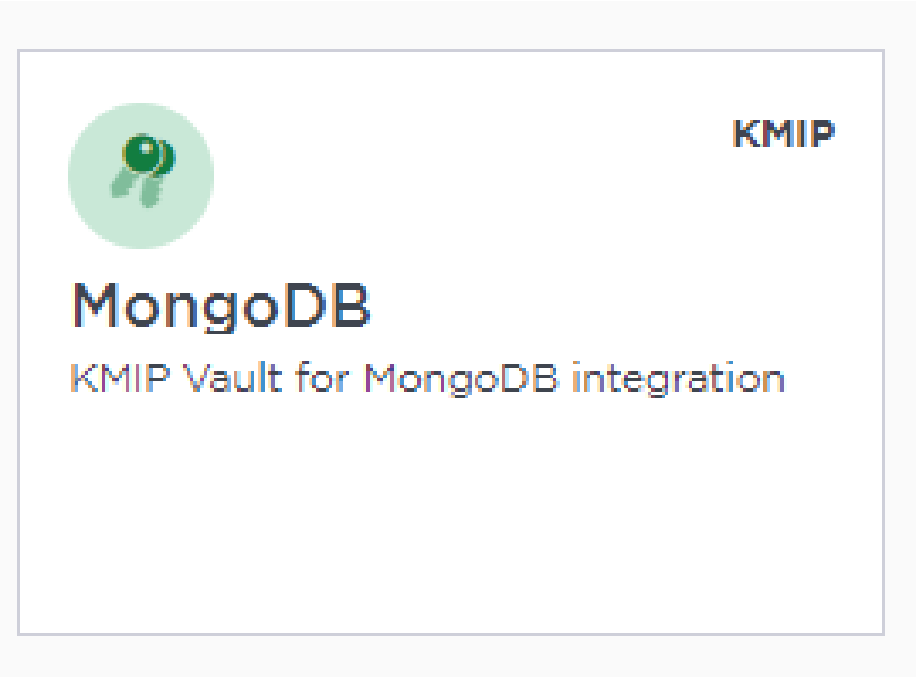
   KeyControl Vault supports the following types of vaults:

   - **Cloud Key Management** - Vault for cloud keys such as BYOK and HYOK.
   - **KMIP** - Vault for KMIP Objects.
   - **PASM** - Vault for objects such as passwords, files, SSH keys, and so on.
   - **Database** - Vault for database keys.
   - **Tokenization** - Vault for tokenization policies.
   - **VM Encryption** - Vault for encrypting VMs.
4. In the **Create Vault** page, create a **KMIP** Vault:
   a. For **Type**, select **KMIP**.
   b. For **Name**, enter the name of the Vault.
   c. For **Description**, enter the description of the Vault.
   d. For **Admin Name**, enter the name of the administrator of the Vault.
   e. For **Admin Email**, enter a valid email for the administrator.



A temporary password will be emailed to the administrator's email address. This is the password that will be used to sign in for the first time to the KMIP Vaults space in KeyControl. In a closed gap environment where email is not available, the password for the user is displayed when you first create the vault. That can be copied and sent to the user.

5. Select **Create Vault**.
6. Select **Close** when the Vault creation completes.
7. The newly-created Vault is added to the Vault dashboard.



8. After the Vault has been created, the KMIP server settings on the appliance are **enabled**.

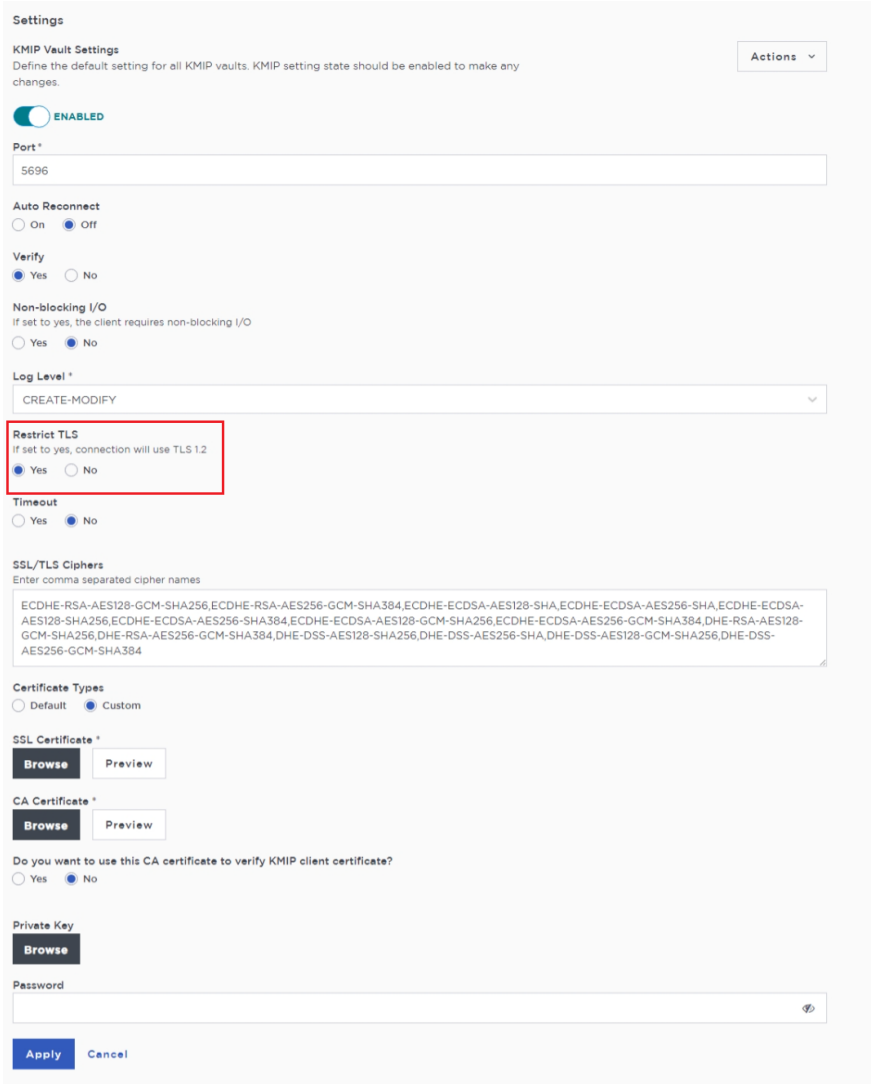
2.3.2. KMIP server settings

The KMIP server settings are set at the KeyControl appliance level and apply to all the KMIP Vaults in the appliance. After a KMIP Vault is created, they are automatically set to **ENABLED**.

To use external key management and configure the KeyControl Vault KMIP settings, refer to the KMIP Client and Server Configuration section of the admin guide.

When using external key management, as is the case in this solution, the KeyControl server is the KMIP server and the MongoDB server is the KMIP client.

1. Select the **Settings** icon on the top right to view/change the KMIP settings.
    a. The defaults settings are appropriate for most applications.
    b. Make sure **Restrict TLS** is enabled.
    c. Make any changes necessary.

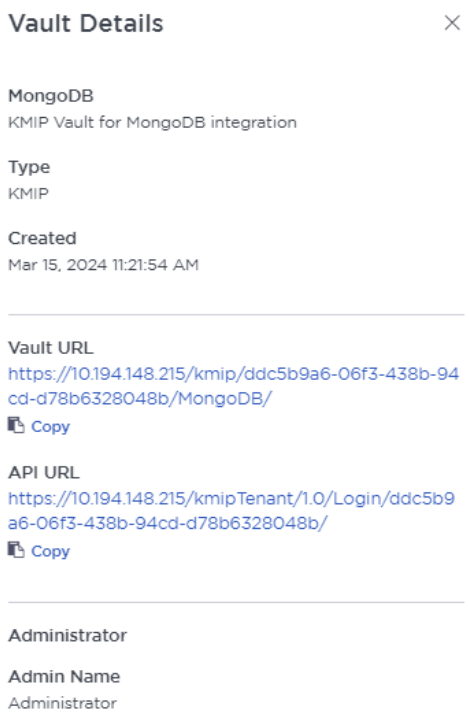


MongoDB requires the KMIP server to support TLS 1.2. Make sure **Restrict TLS** is enabled as shown above.

2. Select **Apply**.

2.3.3. View details for the Vault

To view the details on the Vault, select **View Details** when you hover over the Vault.



2.3.4. Establishing trust between the KeyControl Vault Server and MongoDB

Certificates are required to facilitate the KMIP communications from the KeyControl KMIP Vault and the MongoDB application and conversely. The built-in capabilities in the KeyControl KMIP Vault are used to create and publish the certificates.

For more information on how to create a certificate bundle, refer to Establishing a Trusted Connection with a KeyControl Vault-Generated CSR.

The process below will show how to integrate MongoDB encryption with KeyControl KMIP Vault.

1. Sign in to the KMIP Vault created earlier. Use the login URL and credentials provided to the administrator of the Vault.
2. Select **Security**, then **Client Certificates**.

3. In the **Manage Client Certificate** page, select the **+** icon on the right to create a new certificate.
4. In the **Create Client Certificate** dialog box:
   a. Enter a name in the **Certificate Name** field.
   b. Set the date on which you want the certificate to expire in the **Certificate Expiration** field.
   c. Select **Create**.

   The new certificates are added to the **Manage Client Certificate** pane.
5. Select the certificate and select the **Download** icon to download the certificate.

   The webGUI downloads `certname_datetimestamp.zip`, which contains a user certification/key file called `certname.pem` and a server certification file called `cacert.pem`.
6. Unzip the file so that you have the `certname.pem` file available to upload.
7. The download zip file contains the following:

   - A `certname.pem` file that includes both the client certificate and private key. In this example, this file is called `mongodb.pem`.

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

     The private key section of the `certname.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`.

   These files will be used in the MongoDB configuration next.

2.4. Configure MongoDB Server Security options to use KMIP

Add Key Management Security Options to `/etc/mongod.conf`. The MongoDB configuration file contains a security section. In this section the server name, certificate details, KMIP port number and encryption cipher information must be

updated to use KeyControl as a key management service.

1. Copy the certificate files to a location on the server that they can be used by MongoDB. For example, in your home directory:

```
% sudo mkdir -p /opt/mongodb/security
% sudo cp ~/mongodb.pem /opt/mongodb/security
% sudo cp ~/cacert.pem /opt/mongodb/security
% sudo chmod 644 /opt/mongodb/security/*
```

2. Open the `/etc/mongod.conf` file and add the key management configuration options described below:

```
security:
  enableEncryption: true
  encryptionCipherMode: AES256-GCM
  kmip:
    serverName: kcv-10-2-node-1.interop.local,kcv-10-2-node-2.interop.local
    port: 5696
    clientCertificateFile: /opt/mongodb/security/mongodb.pem
    serverCAFile: /opt/mongodb/security/cacert.pem
```

In this example:

enableEncryption

Sets the flag to enable encryption to the database.

encryptionCipherMode

Sets the cipher used. If you are using Linux, select either AES256-GCM or AES256-CBC. For other OSs, only AES256-CBC is available.

serverName

Hostname of KeyControl operating as the KMS KMIP server. This hostname must match the hostname used in the `/etc/hosts` file and cannot be an IP address. The hostname must also contain the subdomain. You can specify multiple KMIP servers as a comma-separated list: `server1.example.com,server2.example.com`. On startup, the `mongod` will attempt to establish a connection to each server in the order listed, and will select the first server to which it can successfully establish a connection. KMIP server selection occurs only at startup.

port

Contains the KMIP port number. Port 5696 is the default port assignment for KMIP communication and is the KMIP port used by KeyControl.

clientCertificateFile

Contains the directory/file location of the client certificate used for authenticating MongoDB to the KMIP server.

serverCAFile

Contains the directory/file location of the KeyControl root CA certificate.

2.4.1. Add information to the /etc/hosts file

When you added the security section to the `mongod.conf` file, add the KeyControl `hostname.domainname` to the `/etc/hosts` file to resolve the IP address of the KMIP server to the hostname. The hostname must match the hostname used in the `mongod.conf` and cannot be an IP address. The hostname must also contain the subdomain.

2.4.2. Allowing MongoDB to talk to KeyControl on port 5696

Port 5696 needs to be allowed in selinux configuration so mondobd can reach the KeyControl vault server. If you don't do this, mongodb will fail to start with messages like this:

```
{"t":{"$date":"2024-03-18T12:26:46.836-04:00"},"s":"W",  "c":"STORAGE",  "id":24240,
"ctx":"initandlisten","msg":"Connection to KMIP server failed. Trying next server","attr":{"failedHost":"kcv-10-
2-node-1.interop.local:5696","nextHost":"kcv-10-2-node-2.interop.local:5696"}}
{"t":{"$date":"2024-03-18T12:26:46.837-04:00"},"s":"W",  "c":"NETWORK",  "id":23190,
"ctx":"initandlisten","msg":"Failed to connect to remote
host","attr":{"remoteSocketAddress":"10.194.148.216","remoteSocketAddressPort":5696,"call":"connect","error":"Per
mission denied"}}
{"t":{"$date":"2024-03-18T12:26:46.837-04:00"},"s":"E",  "c":"STORAGE",  "id":24248,
"ctx":"initandlisten","msg":"Unable to retrieve
key","attr":{"keyId":".system","error":{"code":2,"codeName":"BadValue","errmsg":"Could not connect to KMIP server
10.194.148.216:5696"}}}
{"t":{"$date":"2024-03-18T12:26:46.847-04:00"},"s":"W",  "c":"NETWORK",  "id":23190,
"ctx":"initandlisten","msg":"Failed to connect to remote
host","attr":{"remoteSocketAddress":"10.194.148.215","remoteSocketAddressPort":5696,"call":"connect","error":"Per
mission denied"}}
{"t":{"$date":"2024-03-18T12:26:46.847-04:00"},"s":"W",  "c":"STORAGE",  "id":24240,
"ctx":"initandlisten","msg":"Connection to KMIP server failed. Trying next server","attr":{"failedHost":"kcv-10-
2-node-1.interop.local:5696","nextHost":"kcv-10-2-node-2.interop.local:5696"}}
{"t":{"$date":"2024-03-18T12:26:46.847-04:00"},"s":"W",  "c":"NETWORK",  "id":23190,
"ctx":"initandlisten","msg":"Failed to connect to remote
host","attr":{"remoteSocketAddress":"10.194.148.216","remoteSocketAddressPort":5696,"call":"connect","error":"Per
mission denied"}}
{"t":{"$date":"2024-03-18T12:26:46.847-04:00"},"s":"E",  "c":"STORAGE",  "id":24248,
"ctx":"initandlisten","msg":"Unable to retrieve
key","attr":{"keyId":".system","error":{"code":2,"codeName":"BadValue","errmsg":"Could not connect to KMIP server
10.194.148.216:5696"}}}
```

Open up port 5696 on selinux using the following command:

```
% sudo semanage port -a -t http_port_t -p tcp 5696
```

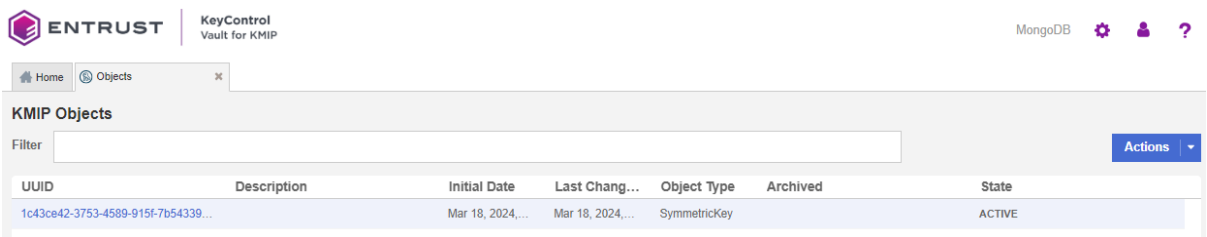
2.4.3. Test configuration

Once you have finished configuring the `mongod.conf` and hosts files, start MongoDB. When started, MongoDB will attempt to retrieve an encryption key stored on KeyControl. If MongoDB does not find a key, it creates one and stores it in KeyControl.

```
% sudo systemctl restart mongod
% sudo systemctl status mongod

● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-03-18 12:36:56 EDT; 8s ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 1811904 (mongod)
   Memory: 82.2M
   CGroup: /system.slice/mongod.service
           └─1811904 /usr/bin/mongod -f /etc/mongod.conf

Mar 18 12:36:56 mongodb-kc102-redhat-8 systemd[1]: Started MongoDB Database Server.
Mar 18 12:36:56 mongodb-kc102-redhat-8 mongod[1811904]: {"t":{"$date":"2024-03-18T16:36:56.841Z"},"s":"I",
"c":"CONTROL",  "id":7484500, "ctx":"main","msg":"Environment variable MONGODB>
```

If you try enable encryption on an existing instance of MongoDB and data already exists in the MongoDB data directory `/var/lib/mongo`, you will get an error when you start the `mongod` service. You can only have encryption enabled in a blank data directory. See Enable encryption on an existing MongoDB instance for instructions on how to get around the issue.

To confirm that the master key was successfully created, log in to KeyControl Tenant URL (KMIP Login) and look under **Objects** as shown below.



You should see the master key that you have just created.

2.5. Verify that the encryption is working and that MongoDB is using KeyControl to manage the keys

You can use the following procedures to validate that encryption is working and that KeyControl is being used by MongoDB.

2.5.1. Test access when KeyControl is available

To test access:

1. Stop the network services on the MongoDB server and try to restart the `mongod` service.

   The `mongod` service will not start because it cannot connect to one of the KeyControl servers. For example:

```
% sudo systemctl restart mongod
% sudo systemctl status mongod

● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since Mon 2024-03-18 13:39:23 EDT; 19s ago
     Docs: https://docs.mongodb.org/manual
  Process: 1832247 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=14)
 Main PID: 1832247 (code=exited, status=14)

Mar 18 13:39:22 mongodb-kc102-redhat-8 systemd[1]: Started MongoDB Database Server.
Mar 18 13:39:22 mongodb-kc102-redhat-8 mongod[1832247]: {"t":{"$date":"2024-03-18T17:39:22.514Z"},"s":"I",
"c":"CONTROL",  "id":7484500, "ctx":"main","msg":"Environment variable MONGODB_CONFIG_OVERRIDE_NOFORK == 1,
overriding \"processManagement.fork\" to false"}
Mar 18 13:39:23 mongodb-kc102-redhat-8 systemd[1]: mongod.service: Main process exited, code=exited,
status=14/n/a
Mar 18 13:39:23 mongodb-kc102-redhat-8 systemd[1]: mongod.service: Failed with result 'exit-code'.
```

2. Restart the network services and try again. The `mongod` service starts successfully.

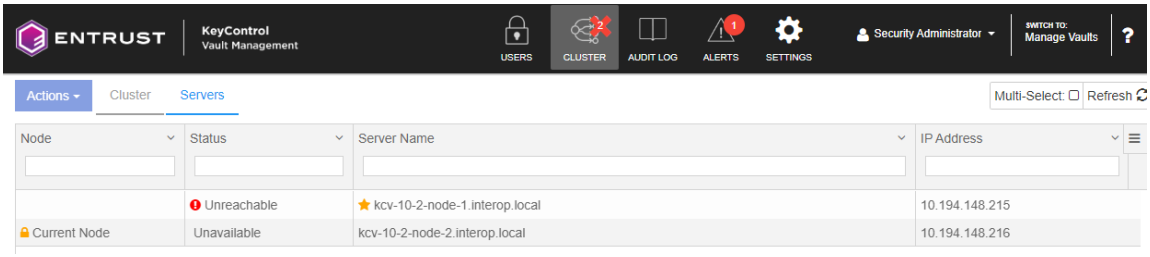
```
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-03-18 13:42:27 EDT; 5s ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 1832593 (mongod)
   Memory: 178.8M
   CGroup: /system.slice/mongod.service
           └─1832593 /usr/bin/mongod -f /etc/mongod.conf

Mar 18 13:42:27 mongodb-kc102-redhat-8 systemd[1]: Started MongoDB Database Server.
Mar 18 13:42:27 mongodb-kc102-redhat-8 mongod[1832593]: {"t":{"$date":"2024-03-18T17:42:27.171Z"},"s":"I",
"c":"CONTROL",  "id":7484500, "ctx":"main","msg":"Environment variable MONGODB_CONFIG_OVERRIDE_NO
```

When MongoDB is encrypted, it is only accessible when KeyControl is available and master key is found.

2.5.2. Validate access when a KeyControl node in the cluster is not available

To validate access:

1. Bring down one of the KeyControl nodes and check that you can access MongoDB. For example:



2. Attempt to start up `mongod` when one of the KeyControl nodes in the cluster is down:

```
% sudo systemctl restart mongod
% sudo systemctl status mongod

● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-03-18 13:56:50 EDT; 15s ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 1837551 (mongod)
   Memory: 181.0M
   CGroup: /system.slice/mongod.service
           └─1837551 /usr/bin/mongod -f /etc/mongod.conf

Mar 18 13:56:50 mongodb-kc102-redhat-8 systemd[1]: mongod.service: Succeeded.
Mar 18 13:56:50 mongodb-kc102-redhat-8 systemd[1]: Stopped MongoDB Database Server.
Mar 18 13:56:50 mongodb-kc102-redhat-8 systemd[1]: Started MongoDB Database Server.
Mar 18 13:56:51 mongodb-kc102-redhat-8 mongod[1837551]: {"t":{"$date":"2024-03-18T17:56:51.027Z"},"s":"I",
"c":"CONTROL",  "id":7484500, "ctx":"main","msg":"Environment variable MONGODB_CONFIG_OVERRIDE_NO>
```

When one of its nodes is down, the KeyControl cluster goes out of `Healthy` status. New keys can only be created when the cluster is in `Healthy` status. Therefore, rotating keys should not be attempted when one of the nodes in the cluster is down.

2.5.3. Rotate the master key in KeyControl with MongoDB

You can rotate the master key, the only externally managed key. With the new master key, the internal keystore will be re-encrypted but the database keys will be otherwise left unchanged. This obviates the need to re-encrypt the entire data set.

1. Stop the `mongod` service:

```
% sudo systemctl stop mongod
```

2. Add the following to `/etc/mongod.conf` under the `kmip` settings:

```
  kmip:
    rotateMasterKey: true
```

3. Start the `mongod` service:

```
% sudo systemctl start mongod
```

4. Upon successful completion of the master key rotation and re-encryption of the database keystore, a new master key will be created in KeyControl.
5. Remove the `kmip rotateMasterKey` setting from the `/etc/mongod.conf` file by commenting it out:

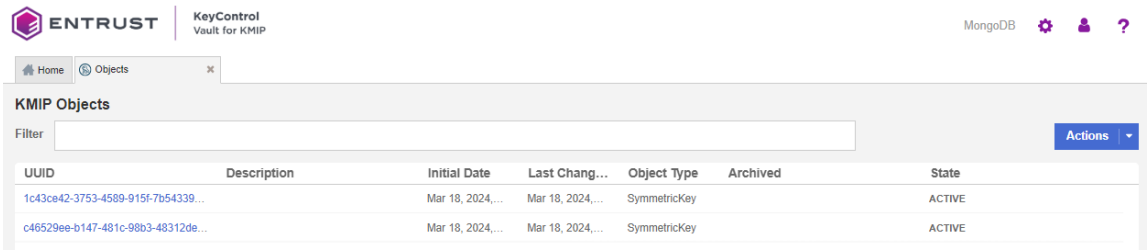
```
  kmip:
#    rotateMasterKey: true
```

6. Restart the `mongod` service:

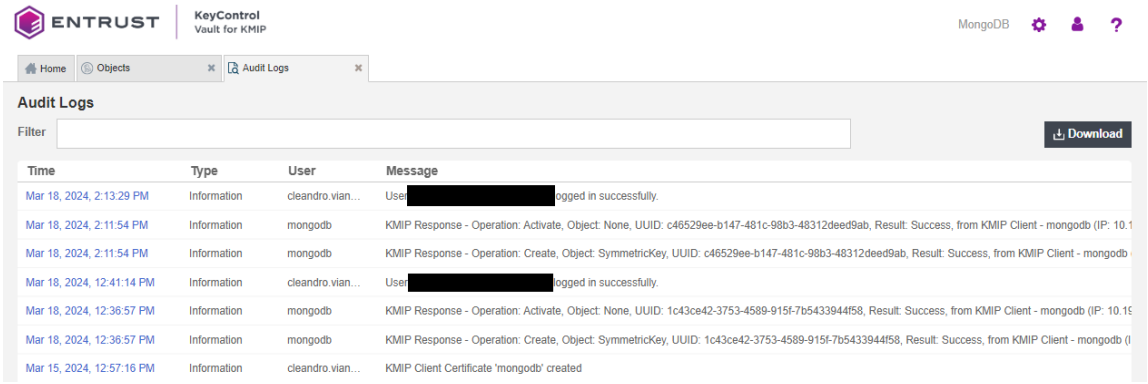
```
> sudo systemctl restart mongod

> sudo systemctl status mongod

● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-03-18 14:15:44 EDT; 6s ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 1842217 (mongod)
   Memory: 177.1M
   CGroup: /system.slice/mongod.service
           └─1842217 /usr/bin/mongod -f /etc/mongod.conf

Mar 18 14:15:44 mongodb-kc102-redhat-8 systemd[1]: Started MongoDB Database Server.
Mar 18 14:15:44 mongodb-kc102-redhat-8 mongod[1842217]: {"t":{"$date":"2024-03-18T18:15:44.466Z"},"s":"I",
"c":"CONTROL",  "id":7484500, "ctx":"main","msg":"Environment variable MONGODB_CONFIG_OVERRIDE_NO>
```

7. When the key is rotated, check on KeyControl if the master key is rotated.

   To confirm that the new master key was successfully created, log in to KeyControl KMIP Vault URL, look under **Objects**. For example:



You will find a new key created in KeyControl. You should also be able to see the creation of the key in the KeyControl Audit Logs. For example:



2.5.4. Enable encryption on an existing MongoDB instance

If you try to enable encryption on an existing instance of MongoDB and data already exists in the MongoDB data directory `/var/lib/mongo`, you should see an error when you start the `mongod` service. For example:

```
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since Mon 2024-03-18 14:29:03 EDT; 5s ago
     Docs: https://docs.mongodb.org/manual
  Process: 1846502 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=14)
 Main PID: 1846502 (code=exited, status=14)

Mar 18 14:29:02 mongodb-kc102-redhat-8 systemd[1]: mongod.service: Succeeded.
Mar 18 14:29:02 mongodb-kc102-redhat-8 systemd[1]: Stopped MongoDB Database Server.
Mar 18 14:29:02 mongodb-kc102-redhat-8 systemd[1]: Started MongoDB Database Server.
Mar 18 14:29:02 mongodb-kc102-redhat-8 mongod[1846502]: {"t":{"$date":"2024-03-18T18:29:02.572Z"},"s":"I",
"c":"CONTROL",  "id":7484500, "ctx":"main","msg":"Environment variable MONGODB_CONFIG_OVERRIDE_NO>
Mar 18 14:29:03 mongodb-kc102-redhat-8 systemd[1]: mongod.service: Main process exited, code=exited,
status=14/n/a
Mar 18 14:29:03 mongodb-kc102-redhat-8 systemd[1]: mongod.service: Failed with result 'exit-code'.
```

When you look in `/var/log/mongodb/mongod.log` for what may be causing the failure, you should see errors. For example:

```
{"t":{"$date":"2024-03-18T14:29:03.157-04:00"},"s":"E",  "c":"STORAGE",  "id":24248,
"ctx":"initandlisten","msg":"Unable to retrieve
key","attr":{"keyId":".system","error":{"code":2,"codeName":"BadValue","errmsg":"There are existing data files,
but no valid keystore could be located."}}}
```

```
{"t":{"$date":"2024-03-18T14:29:03.167-04:00"},"s":"E",  "c":"STORAGE",  "id":24248,
"ctx":"initandlisten","msg":"Unable to retrieve
key","attr":{"keyId":".system","error":{"code":2,"codeName":"BadValue","errmsg":"There are existing data files,
but no valid keystore could be located."}}}
{"t":{"$date":"2024-03-18T14:29:03.176-04:00"},"s":"E",  "c":"STORAGE",  "id":24248,
"ctx":"initandlisten","msg":"Unable to retrieve
key","attr":{"keyId":".system","error":{"code":2,"codeName":"BadValue","errmsg":"There are existing data files,
but no valid keystore could be located."}}}
```

The error clearly states that the data files are present in the db path without any key store. This translates to how it works with mongodb - you can only have encryption enabled in a blank data directory. You can achieve this by taking the backup, stopping the `mongod` instance, clearing the data directory, restarting by enabling encryption and then restore from backup.

1. If you already added the security section to `/etc/mongod.conf`, comment it out:

```
#security:
#  enableEncryption: true
#  encryptionCipherMode: AES256-GCM
#  kmip:
#    serverName: kcv-10-2-node-1.interop.local,kcv-10-2-node-2.interop.local
#    port: 5696
#    clientCertificateFile: /opt/mongodb/security/mongodb.pem
#    serverCAFile: /opt/mongodb/security/cacert.pem
```

2. Restart the `mongod` service:

```
% sudo systemctl restart mongod
% sudo systemctl status mongod

● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-03-18 14:33:12 EDT; 8s ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 1846611 (mongod)
   Memory: 170.5M
   CGroup: /system.slice/mongod.service
           └─1846611 /usr/bin/mongod -f /etc/mongod.conf

Mar 18 14:33:12 mongodb-kc102-redhat-8 systemd[1]: Started MongoDB Database Server.
Mar 18 14:33:12 mongodb-kc102-redhat-8 mongod[1846611]: {"t":{"$date":"2024-03-18T18:33:12.565Z"},"s":"I",
"c":"CONTROL",  "id":7484500, "ctx":"main","msg":"Environment variable MONGODB_CONFIG_OVERRIDE_NO>
```

3. Make a backup of the data:

```
% sudo mkdir -p /data/backup
% sudo mongodump --out=/data/backup

2024-03-18T14:34:07.023-0400    writing admin.system.version to /data/backup/admin/system.version.bson
2024-03-18T14:34:07.024-0400    done dumping admin.system.version (1 document)
2024-03-18T14:34:07.025-0400    writing mydb1.movie to /data/backup/mydb1/movie.bson
2024-03-18T14:34:07.027-0400    done dumping mydb1.movie (1 document)
```

4. Put the security section back in `/etc/mongod.conf`:

```
security:
  enableEncryption: true
  encryptionCipherMode: AES256-GCM
  kmip:
    serverName: kcv-10-2-node-1.interop.local,kcv-10-2-node-2.interop.local
    port: 5696
    clientCertificateFile: /opt/mongodb/security/mongodb.pem
    serverCAFile: /opt/mongodb/security/cacert.pem
```

5. Clean up the data directory `/var/lib/mongo` and remove all files and directories:

```
% cd /var/lib
% cp -rp mongo mongo.backup
% rm -rf mongo/*
```

6. Restart the `mongod` service:

```
% sudo systemctl restart mongod
% sudo systemctl status mongod

● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-03-18 14:36:07 EDT; 6s ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 1847541 (mongod)
   Memory: 72.9M
   CGroup: /system.slice/mongod.service
           └─1847541 /usr/bin/mongod -f /etc/mongod.conf

Mar 18 14:36:07 mongodb-kc102-redhat-8 systemd[1]: Started MongoDB Database Server.
Mar 18 14:36:07 mongodb-kc102-redhat-8 mongod[1847541]: {"t":{"$date":"2024-03-18T18:36:07.973Z"},"s":"I",
"c":"CONTROL",  "id":7484500, "ctx":"main","msg":"Environment variable MONGODB_CONFIG_OVERRIDE_NO>
```

7. Restore the backup:

```
% sudo mongorestore /data/backup

2024-03-18T14:37:22.551-0400    preparing collections to restore from
2024-03-18T14:37:22.552-0400    reading metadata for mydb1.movie from
/data/backup/mydb1/movie.metadata.json
2024-03-18T14:37:22.569-0400    restoring mydb1.movie from /data/backup/mydb1/movie.bson
2024-03-18T14:37:22.580-0400    finished restoring mydb1.movie (1 document, 0 failures)
2024-03-18T14:37:22.580-0400    no indexes to restore for collection mydb1.movie
2024-03-18T14:37:22.580-0400    1 document(s) restored successfully. 0 document(s) failed to restore.
```

The existing data should be now protected by encryption.

Chapter 3. Additional resources and related products

3.1. Video

3.2. nShield Connect

3.3. nShield as a Service

3.4. KeyControl

3.5. Entrust digital security solutions

3.6. nShield product documentation