MongoDB and Entrust KeyControl

Integration Guide

23 Sep 2022

Contents

1. Introduction

This document describes the integration of MongoDB with the Entrust KeyControl Key Management Solution (KMS). Entrust KeyControl 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 server as a KMS in MondoDB.

To install and configure the Entrust KeyControl server as a KMIP server, see the `Entrust KeyControl nShield HSM Integration Guide`. You can access this in the Entrust Document Library.

Also refer to the MongoDB online documentation.

1.2. Requirements

- Entrust KeyControl version 5.5.1 or later

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

- MongoDB Enterprise Edition 6.0.1 or later

1.3. High-availability considerations

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

1.4. Product configuration

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

Product	Version
MongoDB Enterprise Edition	6.0.1

Product	Version
Entrust KeyControl	5.5.1
Red Hat Enterprise Linux 8.6 (Ootpa)	Kernel: Linux 4.18.0-240.el8.x86_64

2. Procedures

2.1. Installation overview

1. Install the MongoDB Enterprise Edition.
2. Install and configure Entrust KeyControl.
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 Tue 2022-09-13 12:12:40 EDT; 14s ago
     Docs: https://docs.mongodb.org/manual
  Process: 144694 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 144692 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 144690 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 144687 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
 Main PID: 144696 (mongod)
   Memory: 73.6M
   CGroup: /system.slice/mongod.service
           └─144696 /usr/bin/mongod -f /etc/mongod.conf

Sep 13 12:12:38 mongodb-redhat-8 systemd[1]: Starting MongoDB Database Server...
Sep 13 12:12:38 mongodb-redhat-8 mongod[144694]: about to fork child process, waiting until server is ready for
connections.
Sep 13 12:12:38 mongodb-redhat-8 mongod[144694]: forked process: 144696
Sep 13 12:12:40 mongodb-redhat-8 mongod[144694]: child process started successfully, parent exiting
Sep 13 12:12:40 mongodb-redhat-8 systemd[1]: Started MongoDB Database Server.
```

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: 6320ac843b58494b26132854
Connecting to:
mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.5.4
Using MongoDB:          6.0.1
Using Mongosh:          1.5.4

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
   2022-09-13T12:14:03.745-04:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
   2022-09-13T12:14:03.745-04:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to
'never'
   2022-09-13T12:14:03.745-04:00: 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

To install and configure Entrust KeyControl, follow the installation and setup instructions in the `Entrust KeyControl nShield HSM Integration Guide`. You can access this in the Entrust Document Library.

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

2.3.1. Creating the KMIP Tenant in KeyControl

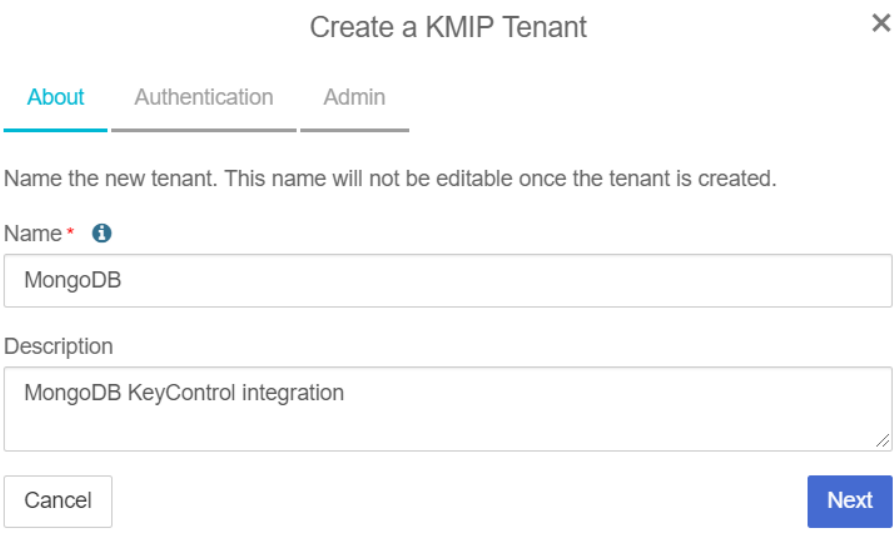
Certificates are required to facilitate the two-way KMIP communications between the KeyControl server and MongoDB. Use the built-in capabilities in the KeyControl server to create and publish the certificates. With KeyControl 5.5.1 Multi Tenancy you will need to first create a tenant before you can create the certificates.

1. Log into the KeyControl web user interface using an account with Security Admin privileges.
2. In the top menu bar, select **KMIP** and then select the **Tenants** tab.
3. Select **Actions** > **Create a KMIP tenant**.

   The **Create a KMIP Tenant** dialog appears.

4. In the **About** tab, enter the **Name** of the tenant and a **Description**.

   > The tenant name cannot be changed after the tenant is created.



Name the new tenant. This name will not be editable once the tenant is created.

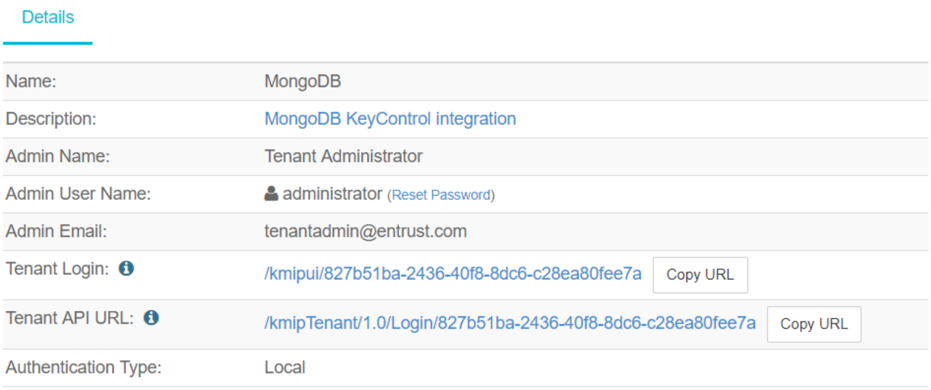
Name*: MongoDB

Description: MongoDB KeyControl integration

5. Select **Next**.
6. In the **Authentication** tab, for **Authentication Type**, select **Local User Authentication**.

   If you want to use **Managed Authentication**, this will require an Active Directory server. For the purpose of this guide, **Local User Authentication** is used. Please refer to the KeyControl Online documentation for more information on how to use **Managed Authentication**. Please refer to KMIP Tenant Authentication for more details.

7. Select **Next**.
8. In the **Admin** tab, enter the Administrator information:
   a. For **User Name**, enter the Administrator's user name.
   b. For **Full Name**, enter the Administrator's full name.
   c. For **Email**, enter the Administrator's email.
   d. For **Password**, set the Administrator's password.
   e. For **Password Expiration**, set the date when you want the password to expire.
9. Select **Create**. This will create the tenant in KeyControl. Once it is created, it will be listed under the **Tenants** tab.
10. Select the newly created tenant. When you select it the information for the tenant is displayed. For example:



Name:	MongoDB
Description:	MongoDB KeyControl integration
Admin Name:	Tenant Administrator
Admin User Name:	👤 administrator (Reset Password)
Admin Email:	tenantadmin@entrust.com
Tenant Login: ℹ	/kmipui/827b51ba-2436-40f8-8dc6-c28ea80fee7a Copy URL
Tenant API URL: ℹ	/kmipTenant/1.0/Login/827b51ba-2436-40f8-8dc6-c28ea80fee7a Copy URL
Authentication Type:	Local

11. Test the MongoDB tenant by selecting the **Tenant Login** URL. Attempt to log in using the user you provided during the tenant configuration. If successful, the tenant is ready to create the certificate bundle for MongoDB.

2.3.2. Establishing trust between the KeyControl Server and MongoDB

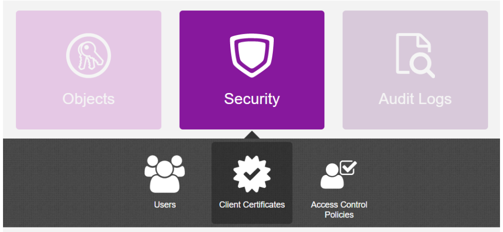
Certificates are required to facilitate all KMIP communications between the KeyControl Server and MongoDB.

1. Log into the KeyControl web user interface using the **Tenant Login** URL.

Use the Administrator login ID and password created during the tenant creation.

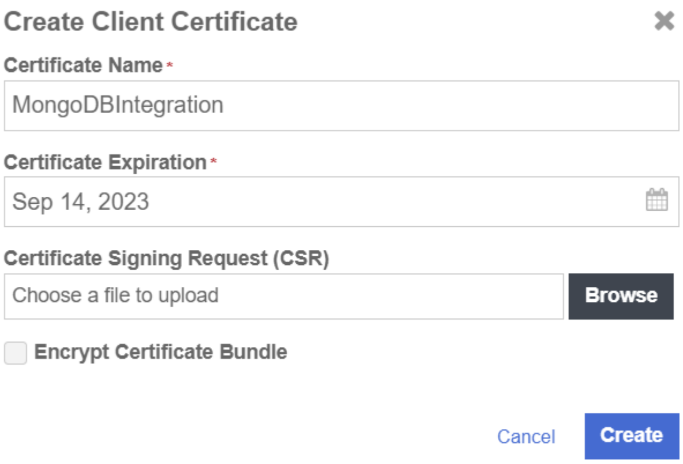
> The **Tenant Login** URL was displayed at the end of the Creating the KMIP Tenant in KeyControl procedure and is different from the standard KeyControl web user interface URL.

2. Select **Security**, then select **Client Certificates**.



The **Manage Client Certificate** tab appears.

3. Select the **+** icon on the right to create a new certificate.
4. In the **Create Client Certificate** dialog:
   a. For **Certificate Name**, enter a name.
   b. For **Certificate Expiration**, set the date on which you want the certificate to expire.
   c. Accept the defaults for remaining properties. For example:



Create Client Certificate

Certificate Name*
MongoDBIntegration

Certificate Expiration*
Sep 14, 2023

Certificate Signing Request (CSR)
Choose a file to upload

□ Encrypt Certificate Bundle

   d. Select **Create**.
5. Select the new certificate once it is created and then select **Download**.

A .zip file downloads, which contains:

- A `<cert_name>.pem` file that includes both the client certificate and private key.

  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. This file is always named cacert.pem.

These files will be used to establish trust between KeyControl and MongoDB. In this example, the <cert_name>.pem file is called MongoDBIntegration.pem and the <cacert>.pem file is called cacert.pem.

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

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 ~/MongoDBIntegration.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: kc-551-1.interop.com,kc-551-2.interop.com
    port: 5696
    clientCertificateFile: /opt/mongodb/security/MongoDBIntegration.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.

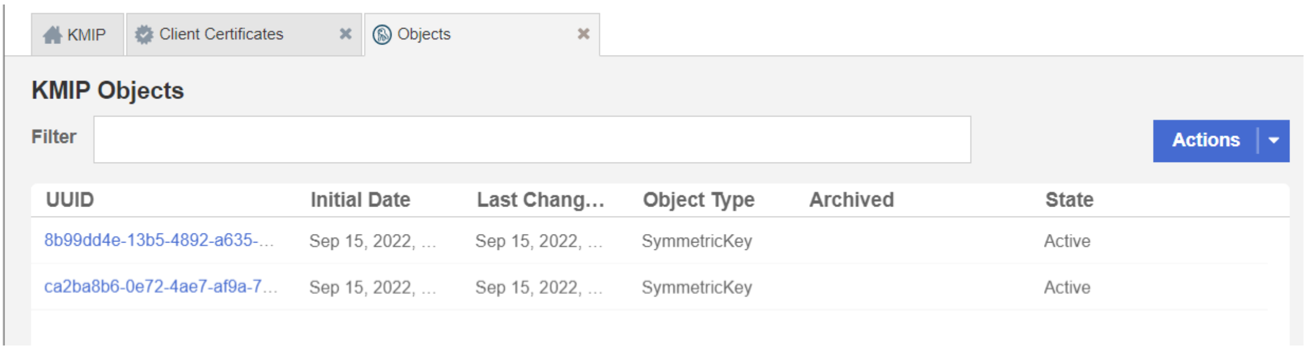
2.4.2. Test configuration

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 Thu 2022-09-15 15:35:07 EDT; 5s ago
     Docs: https://docs.mongodb.org/manual
  Process: 1068257 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 1068255 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1068252 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1068250 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
 Main PID: 1068259 (mongod)
   Memory: 74.7M
   CGroup: /system.slice/mongod.service
           └─1068259 /usr/bin/mongod -f /etc/mongod.conf

Sep 15 15:35:06 mongodb-redhat-8 systemd[1]: Starting MongoDB Database Server...
Sep 15 15:35:06 mongodb-redhat-8 mongod[1068257]: about to fork child process, waiting until server is ready for
connections.
Sep 15 15:35:06 mongodb-redhat-8 mongod[1068257]: forked process: 1068259
Sep 15 15:35:07 mongodb-redhat-8 mongod[1068257]: child process started successfully, parent exiting
Sep 15 15:35:07 mongodb-redhat-8 systemd[1]: Started MongoDB Database Server.
```

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.



KMIP Objects

UUID	Initial Date	Last Chang...	Object Type	Archived	State
8b99dd4e-13b5-4892-a635-...	Sep 15, 2022, ...	Sep 15, 2022, ...	SymmetricKey		Active
ca2ba8b6-0e72-4ae7-af9a-7...	Sep 15, 2022, ...	Sep 15, 2022, ...	SymmetricKey		Active

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 Fri 2022-09-16 14:04:12 EDT; 1min 40s ago
        Docs: https://docs.mongodb.org/manual
     Process: 1083728 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=51)
     Process: 1083726 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
     Process: 1083724 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
     Process: 1083723 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
    Main PID: 1068388 (code=exited, status=0/SUCCESS)

   Sep 16 14:04:09 mongodb-redhat-8 systemd[1]: Stopped MongoDB Database Server.
   Sep 16 14:04:09 mongodb-redhat-8 systemd[1]: Starting MongoDB Database Server...
   Sep 16 14:04:09 mongodb-redhat-8 mongod[1083728]: about to fork child process, waiting until server is ready for
   connections.
   Sep 16 14:04:09 mongodb-redhat-8 mongod[1083728]: forked process: 1083731
   Sep 16 14:04:12 mongodb-redhat-8 systemd-coredump[1083743]: Process 1083731 (mongod) of user 973 dumped core.
   Sep 16 14:04:12 mongodb-redhat-8 systemd[1]: mongod.service: Control process exited, code=exited status=51
   Sep 16 14:04:12 mongodb-redhat-8 mongod[1083728]: ERROR: child process failed, exited with 51
   Sep 16 14:04:12 mongodb-redhat-8 mongod[1083728]: To see additional information in this output, start without the
   "--fork" option.
   Sep 16 14:04:12 mongodb-redhat-8 systemd[1]: mongod.service: Failed with result 'exit-code'.
   Sep 16 14:04:12 mongodb-redhat-8 systemd[1]: Failed to start MongoDB Database Server.
   ```

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

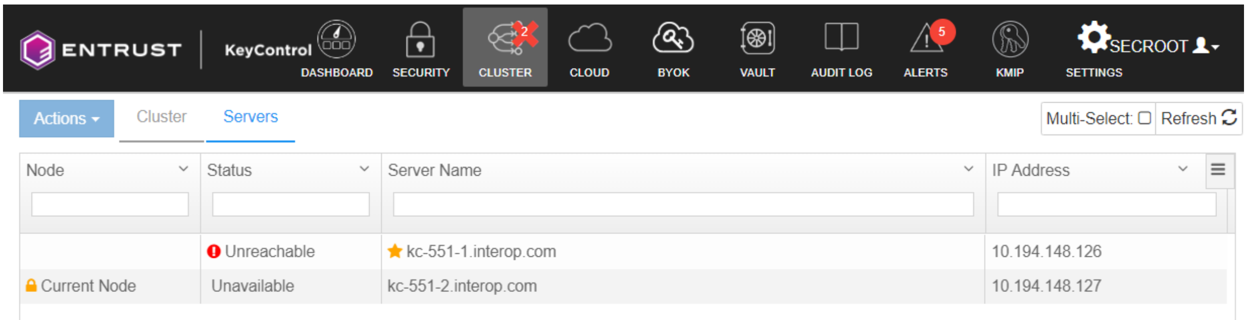
   ```
   ● mongod.service - MongoDB Database Server
      Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
      Active: active (running) since Fri 2022-09-16 14:09:00 EDT; 6s ago
        Docs: https://docs.mongodb.org/manual
     Process: 1084142 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
     Process: 1084141 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
     Process: 1084139 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
     Process: 1084137 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
    Main PID: 1084145 (mongod)
      Memory: 167.4M
      CGroup: /system.slice/mongod.service
              └─1084145 /usr/bin/mongod -f /etc/mongod.conf

   Sep 16 14:08:58 mongodb-redhat-8 systemd[1]: Starting MongoDB Database Server...
   Sep 16 14:08:59 mongodb-redhat-8 mongod[1084142]: about to fork child process, waiting until server is ready for
   connections.
   Sep 16 14:08:59 mongodb-redhat-8 mongod[1084142]: forked process: 1084145
   Sep 16 14:09:00 mongodb-redhat-8 mongod[1084142]: child process started successfully, parent exiting
   Sep 16 14:09:00 mongodb-redhat-8 systemd[1]: Started MongoDB Database Server.
   ```

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:



Node	Status	Server Name	IP Address
	Unreachable	★ kc-551-1.interop.com	10.194.148.126
Current Node	Unavailable	kc-551-2.interop.com	10.194.148.127

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 Fri 2022-09-16 14:13:33 EDT; 6s ago
     Docs: https://docs.mongodb.org/manual
  Process: 1084310 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 1084308 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1084306 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1084302 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
 Main PID: 1084312 (mongod)
   Memory: 168.9M
   CGroup: /system.slice/mongod.service
           └─1084312 /usr/bin/mongod -f /etc/mongod.conf

Sep 16 14:13:26 mongodb-redhat-8 systemd[1]: mongod.service: Succeeded.
Sep 16 14:13:26 mongodb-redhat-8 systemd[1]: Stopped MongoDB Database Server.
Sep 16 14:13:26 mongodb-redhat-8 systemd[1]: Starting MongoDB Database Server...
Sep 16 14:13:26 mongodb-redhat-8 mongod[1084310]: about to fork child process, waiting until server is ready for
connections.
Sep 16 14:13:26 mongodb-redhat-8 mongod[1084310]: forked process: 1084312
Sep 16 14:13:33 mongodb-redhat-8 mongod[1084310]: child process started successfully, parent exiting
Sep 16 14:13:33 mongodb-redhat-8 systemd[1]: Started MongoDB Database Server.
```

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. When the master key has been rotated and the database keystore has been re-encrypted, `mongod` exits. This is visible as you will see a failure when attempting to start.

   ```
   Job for mongod.service failed because the service did not take the steps required by its unit configuration.
   See "systemctl status mongod.service" and "journalctl -xe" for details.
   ```

5. Remove the `kmip rotateMasterKey` setting from the `/etc/mongod.conf` file by commenting it out:

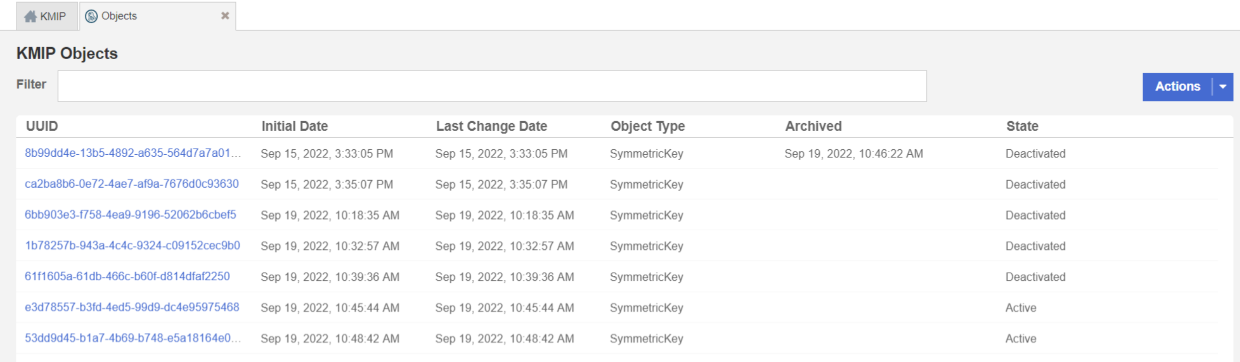
   ```
     kmip:
   #    rotateMasterKey: true
   ```

6. Start the `mongod` service:

```
> sudo systemctl start 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 2022-09-19 10:49:02 EDT; 3min 58s ago
     Docs: https://docs.mongodb.org/manual
  Process: 1154363 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 1154361 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1154359 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1154357 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
 Main PID: 1154365 (mongod)
   Memory: 178.6M
   CGroup: /system.slice/mongod.service
           └──1154365 /usr/bin/mongod -f /etc/mongod.conf

Sep 19 10:49:01 mongodb-redhat-8 systemd[1]: Starting MongoDB Database Server...
Sep 19 10:49:01 mongodb-redhat-8 mongod[1154363]: about to fork child process, waiting until server is ready for
connections.
Sep 19 10:49:01 mongodb-redhat-8 mongod[1154363]: forked process: 1154365
Sep 19 10:49:02 mongodb-redhat-8 mongod[1154363]: child process started successfully, parent exiting
Sep 19 10:49:02 mongodb-redhat-8 systemd[1]: Started MongoDB Database Server.
```

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

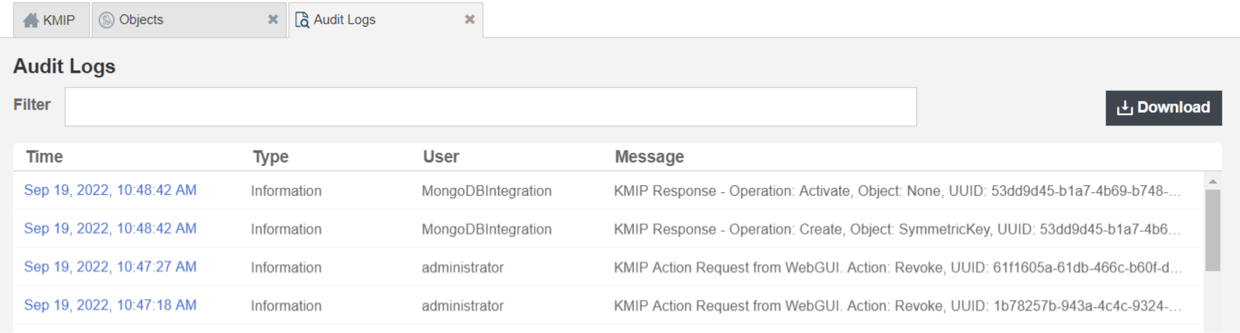
   To confirm that the new master key was successfully created, log in to KeyControl Tenant URL, look under **Objects**. For example:



KMIP Objects

UUID	Initial Date	Last Change Date	Object Type	Archived	State
8b99dd4e-13b5-4892-a635-564d7a7a01...	Sep 15, 2022, 3:33:05 PM	Sep 15, 2022, 3:33:05 PM	SymmetricKey	Sep 19, 2022, 10:46:22 AM	Deactivated
ca2ba8b6-0e72-4ae7-af9a-7676d0c93630	Sep 15, 2022, 3:35:07 PM	Sep 15, 2022, 3:35:07 PM	SymmetricKey		Deactivated
6bb903e3-f758-4ea9-9196-52062b6cbef5	Sep 19, 2022, 10:18:35 AM	Sep 19, 2022, 10:18:35 AM	SymmetricKey		Deactivated
1b78257b-943a-4c4c-9324-c09152cec9b0	Sep 19, 2022, 10:32:57 AM	Sep 19, 2022, 10:32:57 AM	SymmetricKey		Deactivated
61f1605a-61db-466c-b60f-d814dfaf2250	Sep 19, 2022, 10:39:36 AM	Sep 19, 2022, 10:39:36 AM	SymmetricKey		Deactivated
e3d78557-b3fd-4ed5-99d9-dc4e95975468	Sep 19, 2022, 10:45:44 AM	Sep 19, 2022, 10:45:44 AM	SymmetricKey		Active
53dd9d45-b1a7-4b69-b748-e5a18164e0...	Sep 19, 2022, 10:48:42 AM	Sep 19, 2022, 10:48:42 AM	SymmetricKey		Active

   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:



Audit Logs

Time	Type	User	Message
Sep 19, 2022, 10:48:42 AM	Information	MongoDBIntegration	KMIP Response - Operation: Activate, Object: None, UUID: 53dd9d45-b1a7-4b69-b748-...
Sep 19, 2022, 10:48:42 AM	Information	MongoDBIntegration	KMIP Response - Operation: Create, Object: SymmetricKey, UUID: 53dd9d45-b1a7-4b6...
Sep 19, 2022, 10:47:27 AM	Information	administrator	KMIP Action Request from WebGUI. Action: Revoke, UUID: 61f1605a-61db-466c-b60f-d...
Sep 19, 2022, 10:47:18 AM	Information	administrator	KMIP Action Request from WebGUI. Action: Revoke, UUID: 1b78257b-943a-4c4c-9324-...

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 2022-09-19 11:16:10 EDT; 6s ago
     Docs: https://docs.mongodb.org/manual
  Process: 1155054 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=14)
  Process: 1155052 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1155048 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1155047 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
 Main PID: 1154958 (code=exited, status=0/SUCCESS)

Sep 19 11:16:10 mongodb-redhat-8 systemd[1]: mongod.service: Succeeded.
Sep 19 11:16:10 mongodb-redhat-8 systemd[1]: Stopped MongoDB Database Server.
Sep 19 11:16:10 mongodb-redhat-8 systemd[1]: Starting MongoDB Database Server...
Sep 19 11:16:10 mongodb-redhat-8 mongod[1155054]: about to fork child process, waiting until server is ready for
connections.
Sep 19 11:16:10 mongodb-redhat-8 mongod[1155054]: forked process: 1155057
Sep 19 11:16:10 mongodb-redhat-8 mongod[1155054]: ERROR: child process failed, exited with 14
Sep 19 11:16:10 mongodb-redhat-8 mongod[1155054]: To see additional information in this output, start without the "--fork"
option.
Sep 19 11:16:10 mongodb-redhat-8 systemd[1]: mongod.service: Control process exited, code=exited status=14
Sep 19 11:16:10 mongodb-redhat-8 systemd[1]: mongod.service: Failed with result 'exit-code'.
Sep 19 11:16:10 mongodb-redhat-8 systemd[1]: Failed to start MongoDB Database Server.
```

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

```
{"t":{"$date":"2022-09-19T11:16:10.874-04:00"},"s":"E",  "c":"STORAGE",  "id":24248,   "ctx":"initandlisten","msg":"Unable
to retrieve key","attr":{"keyId":".system","error":{"code":2,"codeName":"BadValue","errmsg":"There are existing data
files, but no valid keystore could be located."}}}
{"t":{"$date":"2022-09-19T11:16:10.882-04:00"},"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":"2022-09-19T11:16:10.890-04:00"},"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: kc-551-1.interop.com,kc-551-2.interop.com
#    port: 5696
#    clientCertificateFile: /opt/mongodb/security/MongoDBIntegration.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 2022-09-19 11:22:27 EDT; 10s ago
     Docs: https://docs.mongodb.org/manual
  Process: 1155178 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 1155176 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1155174 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1155171 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
 Main PID: 1155180 (mongod)
   Memory: 167.7M
   CGroup: /system.slice/mongod.service
           └─1155180 /usr/bin/mongod -f /etc/mongod.conf

Sep 19 11:22:26 mongodb-redhat-8 systemd[1]: Starting MongoDB Database Server...
Sep 19 11:22:26 mongodb-redhat-8 mongod[1155178]: about to fork child process, waiting until server is ready for
connections.
Sep 19 11:22:26 mongodb-redhat-8 mongod[1155178]: forked process: 1155180
Sep 19 11:22:27 mongodb-redhat-8 mongod[1155178]: child process started successfully, parent exiting
Sep 19 11:22:27 mongodb-redhat-8 systemd[1]: Started MongoDB Database Server.
```

3. Make a backup of the data:

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

2022-09-19T11:33:59.150-0400    writing admin.system.version to /data/backup/admin/system.version.bson
2022-09-19T11:33:59.150-0400    done dumping admin.system.version (1 document)
2022-09-19T11:33:59.151-0400    writing mydb1.movie to /data/backup/mydb1/movie.bson
2022-09-19T11:33:59.151-0400    done dumping mydb1.movie (1 document)
```

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

```
security:
  enableEncryption: true
  encryptionCipherMode: AES256-GCM
  kmip:
    serverName: kc-551-1.interop.com,kc-551-2.interop.com
    port: 5696
    clientCertificateFile: /opt/mongodb/security/MongoDBIntegration.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 2022-09-19 11:30:03 EDT; 11s ago
     Docs: https://docs.mongodb.org/manual
  Process: 1155466 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 1155464 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1155462 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 1155460 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
 Main PID: 1155468 (mongod)
   Memory: 79.3M
   CGroup: /system.slice/mongod.service
           └─1155468 /usr/bin/mongod -f /etc/mongod.conf

Sep 19 11:30:01 mongodb-redhat-8 systemd[1]: Starting MongoDB Database Server...
Sep 19 11:30:02 mongodb-redhat-8 mongod[1155466]: about to fork child process, waiting until server is ready for
connections.
Sep 19 11:30:02 mongodb-redhat-8 mongod[1155466]: forked process: 1155468
Sep 19 11:30:03 mongodb-redhat-8 mongod[1155466]: child process started successfully, parent exiting
Sep 19 11:30:03 mongodb-redhat-8 systemd[1]: Started MongoDB Database Server.
```

7. Restore the backup:

```
% sudo mongorestore /data/backup

2022-09-19T11:35:50.110-0400    preparing collections to restore from
2022-09-19T11:35:50.110-0400    reading metadata for mydb1.movie from /data/backup/mydb1/movie.metadata.json
2022-09-19T11:35:50.115-0400    restoring mydb1.movie from /data/backup/mydb1/movie.bson
2022-09-19T11:35:50.127-0400    finished restoring mydb1.movie (1 document, 0 failures)
2022-09-19T11:35:50.127-0400    no indexes to restore for collection mydb1.movie
2022-09-19T11:35:50.127-0400    1 document(s) restored successfully. 0 document(s) failed to restore.
```

The existing data should be now protected by encryption.