IBM DB2 and Entrust KeyControl

Integration Guide

2024-04-19

Table of Contents

Chapter 1. Introduction

This document describes the integration of IBM DB2 with the Entrust KeyControl Key Management Solution (KMS). Entrust KeyControl can serve as a KMS to IBM DB2 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 IBM DB2.

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

Also refer to the IBM DB2 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 IBM DB2 account team or through Entrust KeyControl customer support.

- IBM DB2 Server 11.5.7 or later.

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

1.3. High-availability considerations

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

1.4. Product configuration

The integration between the IBM DB2 Server and Entrust KeyControl has been successfully tested in the following configurations:

Product	Version
Ubuntu	20.04.4 LTS
IBM DB2 Server	11.5.7
Entrust KeyControl	5.5.1

Chapter 2. Procedures

2.1. Installation overview

To integrate IBM DB2 with the Entrust KeyControl KMS:

1. Install the IBM DB2 server.
2. Install and configure Entrust KeyControl.
3. Set up a centralized KMIP keystore.
4. Configure the DB2 instance to use the keystore.
5. Verify that the encryption is working and that IBM DB2 is using KeyControl to manage the keys.
6. Configure the nShield HSM in the KeyControl Server.

2.2. Install the IBM DB2 server

Installing the IBM DB2 depends on the operating system on which you are installing it. See the IBM DB2 online documentation for details on how to install IBM DB2 in your environment.

2.3. Install and configure Entrust KeyControl

Follow the installation and setup instructions in the *Entrust KeyControl nShield HSM Integration Guide*. You can access it from the Entrust Document Library and from the nShield Product Documentation website.

Make sure the Entrust KeyControl tenant gets created and KMIP certificates are generated for IBM DB2. These certificates are used in the configuration of the KMS described below.

The following sections describe how to create the KeyControl tenant and KMIP certificates.

2.3.1. Creating the KMIP Tenant in KeyControl

Certificates are required to facilitate the KMIP communications from the KeyControl server to IBM DB2 and conversely. Use the built-in capabilities in the KeyControl server to create and publish the certificates. With KeyControl 5.5.1

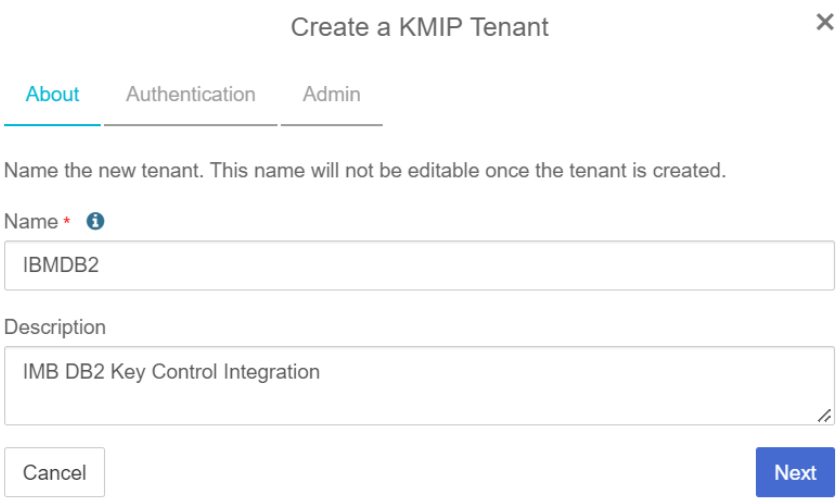
Multi Tenancy you will need to first create a tenant before you can create the certificates.

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.



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.

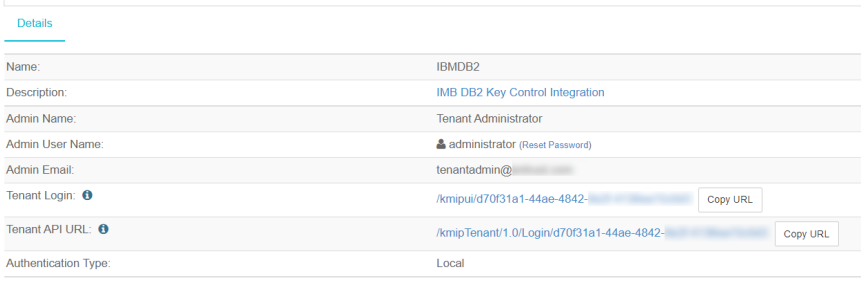
7. Select **Next**.
8. In the **Admin** tab, enter the Administrator information:
   a. For **User Name**, enter the Administrator's user name.
   b. For **Full Name**, enter the Administrator's full name.

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:



11. Test the IBM DB2 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 IBM DB2.

2.3.2. Establishing trust between the KeyControl Server and IBM DB2

Certificates are required to facilitate all KMIP communications between the KeyControl Server and IBM DB2.

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

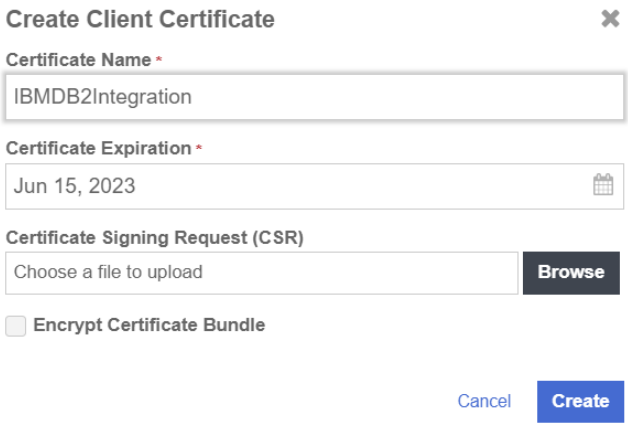
   Use the Administrator login ID and password created during the tenant creation.

   > 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:



   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. It is always named `cacert.pem`.

These files will be used to establish trust between KeyControl and IBM DB2. In this example, the `<cert_name>.pem` file is called `IBMDB2Integration.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. Set up a centralized KMIP keystore

To set up a centralized keystore, with a key manager that is configured for the Key Management Interoperability Protocol (KMIP), for use with DB2 native encryption, you need to create a KMIP keystore configuration file.

After you have created the configuration file, you can enter parameter values to configure DB2 communication between the DB2 instance and the key manager. For more information, see Setting up a centralized KMIP keystore in the IBM documentation site.

2.4.1. Create the keycontrol-kmip.p12 and .sth files

1. Export the libraries for GSKit from the IBM DB2 installation directory.

```
% export LD_LIBRARY_PATH=/opt/ibm/db2/V11.5/lib64/gskit:$LIBPATH
% export PATH=/opt/ibm/db2/V11.5/gskit/bin:$PATH
```

2. Run the utility to create the `.p12` and `.sth` files.

```
% mkdir temp
% cd temp
% gsk8capicmd_64 -keydb -create -db "keycontrol-kmip.p12" -pw "mypassword" -type pkcs12 -stash
% ls -al
-rw------- 1 xxxx xxxx 1392 Jun 15 15:50 keycontrol-kmip.p12
-rw------- 1 xxxx xxxx  193 Jun 15 15:50 keycontrol-kmip.sth
```

3. Add the Client Cert and Key to the SSL Keystore.
   a. Copy the `IBMDB2Integration.pem` file to the `temp` directory. This is one of the certificate files that came in the certificate bundle that you downloaded from Entrust KeyControl. The file is typically located in the user's home directory.

```
% cp ~/IBMDB2Integration.pem .
```

b. Add the client cert and key to the SSL keystore by running the following command:

```
% gsk8capicmd_64 -cert -add -db "keycontrol-kmip.p12" -stashed -label "keycontrol_app_cert" -file
"IBMDB2Integration.pem" -format ascii
```

c. Copy the `cacert.pem` file to the temp directory. This is one of the certificate files that came in the certificate bundle you downloaded from the Entrust KeyControl. The file is typically located in the user's home directory.

```
% cp ~/cacert.pem .
```

d. Import CA Certificate into the SSL keystore by running the following command:

```
% gsk8capicmd_64 -cert -add -db "keycontrol-kmip.p12" -stashed -label "trustedCA" -file cacert.pem
-format ascii -trust enable
```

4. List the certificates in the keystore.

```
% gsk8capicmd_64 -cert -list -db keycontrol-kmip.p12 -stashed

Certificates found
* default, - personal, ! trusted, # secret key
!       trustedCA
-       keycontrol_app_cert
```

5. Copy the `keycontrol-kmip.p12` and `keycontrol-kmip.sth` files to the location where they will be used by IBM DB2.

```
% sudo mkdir -p /opt/ibm/db2/security
% sudo cp keycontrol-kmip.p12 /opt/ibm/db2/security/.
% sudo cp keycontrol-kmip.sth /opt/ibm/db2/security/.
% sudo chmod 644 /opt/ibm/db2/security/*
% ls -al /opt/ibm/db2/security

-rw-r--r-- 1 root root 5882 Jun 15 15:51 keycontrol-kmip.p12
-rw-r--r-- 1 root root  193 Jun 15 15:51 keycontrol-kmip.sth
```

2.4.2. Create the KMIP keystore configuration file

To use DB2 native encryption to store your master key or keys in a centralized keystore using KMIP, you need to create a configuration file that lists details about

the keystore.

1. On the DB2 server, create the KMIP keystore configuration file in a text editor. For example:

```
VERSION=1
PRODUCT_NAME=OTHER
ALLOW_NONCRITICAL_BASIC_CONSTRAINT=TRUE
ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP=TRUE
SSL_KEYDB=/opt/ibm/db2/security/keycontrol-kmip.p12
SSL_KEYDB_STASH=/opt/ibm/db2/security/keycontrol-kmip.sth
SSL_KMIP_CLIENT_CERTIFICATE_LABEL=keycontrol_app_cert
MASTER_SERVER_HOST=10.194.148.126
MASTER_SERVER_KMIP_PORT=5696
CLONE_SERVER_HOST=10.194.148.127
CLONE_SERVER_KMIP_PORT=5696
```

   Attention should be given to the following keywords:

   ALLOW_NONCRITICAL_BASIC_CONSTRAINT

   Set it to TRUE, this allows DB2 to use local Certificate Authority within KMIP server that does not have a "critical" keyword set and avoids "414" error that is returned by GSKit.

   SSL_KEYDB

   This is the absolute path and name of the local keystore file that holds the TLS certificates for communication between the DB2 server and the KMIP key manager. (Required)

   SSL_KEYDB_STASH

   Absolute path and name of the stash file for the local keystore that holds the TLS certificates for communication between the DB2 server and the KMIP key manager. Default value: None. (Optional)

   SSL_KMIP_CLIENT_CERTIFICATE_LABEL

   The label of the TLS certificate for authenticating the client during communication with the KMIP key manager. This is the label you used when you created the keystore. (Required)

   MASTER_SERVER_HOST*

   Host name or IP address of the KMIP key manager. (Required)

   MASTER_SERVER_KMIP_PORT

   The KMIP TLS port of the KMIP key manager. (Required)

CLONE_SERVER_HOST

Host name or IP address of secondary KMIP keystore. Default value: None. You can specify up to five clone servers by repeating the `CLONE_SERVER_HOST` and `CLONE_SERVER_KMIP_PORT` parameter pairs in the configuration file, each host with a different value. Clone servers are considered read-only and are only used for retrieving existing master keys from the KMIP keystore. Clone servers are not used when inserting a new key, which occurs when an existing master key label has not been specified for the `CREATE DATABASE ENCRYPT` or `ADMIN_ROTATE_MASTER_KEY` commands, or for the `db2p12tokmip` executable. (Optional)

CLONE_SERVER_KMIP_PORT

The KMIP TLS port of the secondary KMIP keystore. Default value: None. (Optional)

For a list of the keywords that can be used in this configuration file, see the IBM documentation at Creating a KMIP keystore configuration file

2. Name this file `kmipdb2config.txt` and copy it to where the `.p12` and `.sth` files are.

```
% sudo cp kmipdb2config.txt /opt/ibm/db2/security/.
```

2.5. Configure the DB2 instance to use the keystore

After the keystore is configured, it is ready to be used by DB2. First, add the location of the configuration files and enable the configuration. To configure a DB2 instance to use a keystore for native encryption, you need to set two database manager configuration parameters:

- `KEYSTORE_TYPE`
- `KEYSTORE_LOCATION`

For a centralized keystore, where the key manager product uses the Key Management Interoperability Protocol (KMIP), set `KEYSTORE_TYPE` to `KMIP`, and set `KEYSTORE_LOCATION` to the absolute path and file name of the centralized keystore configuration file.

1. Become the `db2inst1` user:

```
% sudo su - db2inst1
```

2. Update the database parameters:

```
% db2 update dbm cfg using keystore_location /opt/ibm/db2/security/kmipdb2config.txt keystore_type kmip

DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
SQL1362W  One or more of the parameters submitted for immediate modification
were not changed dynamically. Client changes will not be effective until the
next time the application is started or the TERMINATE command has been issued.
Server changes will not be effective until the next DB2START command.
```

3. Restart DB2 again so that the keystore changes take effect:

```
% db2stop

06/15/2022 16:00:23     0   0   SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.

% db2start

06/15/2022 16:00:59     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
```

4. Verify that `dbm cfg` is set correctly by running the following command.

```
% db2 get dbm cfg | grep Keystore

 Keystore type                           (KEYSTORE_TYPE) = KMIP
 Keystore location                   (KEYSTORE_LOCATION) = /opt/ibm/db2/security/kmipdb2config.txt
```

   Look at value of `KEYSTORE_TYPE` and `KEYSTORE_LOCATION`.

2.6. Verify that the encryption is working and that IBM DB2 is using KeyControl to manage the keys

Now that IBM DB2 is configured to use Entrust KeyControl, check that encryption is working and KeyControl is used.

Before starting, become the `db2inst1` user:

```
% sudo su - db2inst1
```

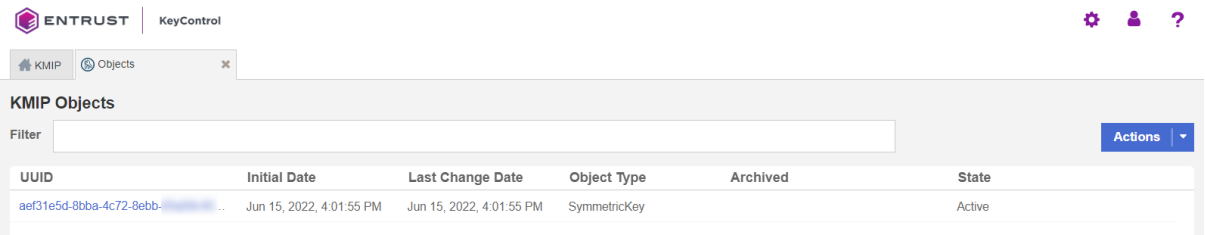
2.6.1. Create an encrypted database

Try to create an encrypted database:

```
% db2 create db mydb1 encrypt
```

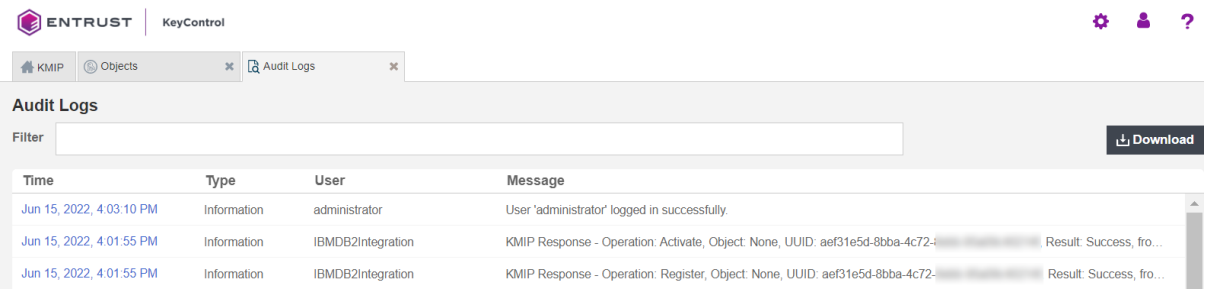
```
DB20000I  The CREATE DATABASE command completed successfully.
```

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



Additionally, you can use the `db2diag` program on the DB2 server to see the operational status.

You can find the Activity logs about the key creation on the **Audit Logs** page in KeyControl. For example:



With KeyControl you will see a complete audit trail every time the key is retrieved. You will also have complete control on these keys and you can revoke access to a key or disable it, in case you want to lock down your data at rest.

The **Objects** tab in the KeyControl UI as described in the Managing KMIP Objects section of the HyTrust KeyControl admin guide.

If you try to create the encrypted database and it fails with return code 414, the certificate is not valid:

```
SQL1782N  The command or operation failed because an error was encountered
accessing the centralized key manager. Reason code "5:414".
```

Either the local certificate or the peer certificate is not valid.

Use the following command to validate the certificates. Do this as the user who created the SSL Store and in the same directory where the SSL store files are located:

```
% gsk8capicmd_64 -cert -validate -db keycontrol-kmip.p12 -stashed

trustedCA : OK
keycontrol_app_cert : CTGSK2052W An invalid basic constraint extension was found.
Additional untranslated info: GSKKM_VALIDATIONFAIL_SUBJECT: GSKNativeValidator:: [IssuerName=]CN=HyTrust
KeyControl Certificate Authority,O=HyTrust Inc.,C=US[Serial#=]60da35b2[SubjectName=]CN=HyTrust KeyControl
Certificate Authority,O=HyTrust Inc.,C=US[Class=]GSKVALMethod::PKIX[Issuer=]CN=HyTrust KeyControl Certificate
Authority,O=HyTrust Inc.,C=US[#=]60da35b2[Subject=]CN=HyTrust KeyControl Certificate Authority,O=HyTrust
Inc.,C=US
CTGSK2052W An invalid basic constraint extension was found.
```

To address this issue, add the following option to the KMIP keystore configuration file:

```
ALLOW_NONCRITICAL_BASIC_CONSTRAINT=TRUE
```

The encrypted database can then be created.

2.6.2. Rotate the Master Key in KeyControl with IBM DB2

1. Become the `db2inst1` user:

   ```
   % sudo su - db2inst1
   ```

2. List your DB directory:

   ```
   % db2 list db directory

   System Database Directory

    Number of entries in the directory = 1

   Database 1 entry:

    Database alias                       = MYDB1
    Database name                        = MYDB1
    Local database directory             = /home/db2inst1
    Database release level               = 15.00
    Comment                              =
    Directory entry type                 = Indirect
    Catalog database partition number    = 0
    Alternate server hostname            =
    Alternate server port number         =
   ```

3. Connect the DB to the same database:

   ```
   % db2 connect to MYDB1


      Database Connection Information

    Database server        = DB2/LINUXX8664 11.5.7.0
   ```

```
 SQL authorization ID   = DB2INST1
 Local database alias   = MYDB1
```

4. Check the encryption information:

```
% db2 "select * from table(sysproc.admin_get_encryption_info())"


OBJECT_NAME
OBJECT_TYPE          ALGORITHM        ALGORITHM_MODE   KEY_LENGTH MASTER_KEY_LABEL
KEYSTORE_NAME
KEYSTORE_TYPE        KEYSTORE_HOST
KEYSTORE_PORT KEYSTORE_IP                               KEYSTORE_IP_TYPE PREVIOUS_MASTER_KEY_LABEL
AUTH_ID
APPL_ID
ROTATION_TIME
------------------------------------------------------------------------------------------------------------
-------------------- -------------------- ---------------- ---------------- ----------
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
----------------------------------------
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
---------------------------------------- --------------------
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
---------------------------------------- -------------- ------------------------------------
---------------
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
----------------------------------------
------------------------------------------------------------------------------------------------------------
--------------------
------------------------------------------------------------------------------------------------------------
-------------------- -------------------------
MYDB1
DATABASE             AES              CBC                     256 DB2_SYSGEN_db2inst1_MYDB1_2022-06-15-
16.01.54_1DB10CAA
/opt/ibm/db2/security/kmipdb2config.txt
KMIP                 10.194.148.126
5696                                         Unknown IP Type  DB2_SYSGEN_db2inst1_MYDB1_2022-06-15-
16.01.54_1DB10CAA
DB2INST1
*LOCAL.db2inst1.220615200154
2022-06-15-16.01.55.000000

  1 record(s) selected.
```

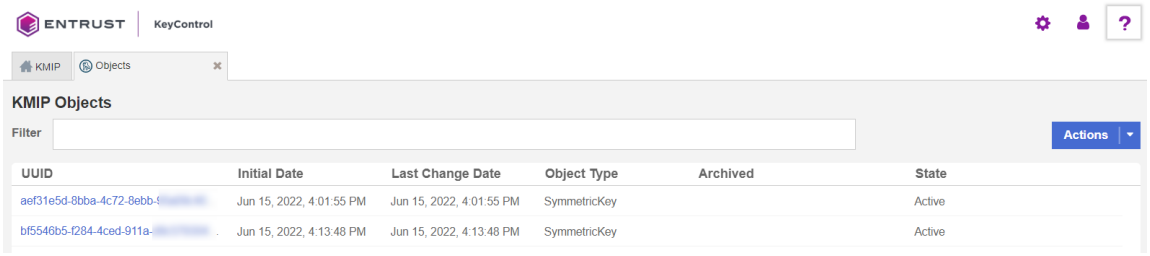
5. Rotate the master key from DB2.

```
% db2 "CALL SYSPROC.ADMIN_ROTATE_MASTER_KEY (NULL)"

 Value of output parameters
  --------------------------
  Parameter Name  : LABEL
  Parameter Value : DB2_SYSGEN_db2inst1_MYDB1_2022-06-15-16.13.48_E4915B6E

  Return Status = 0
```

6. When the key is rotated, check on KeyControl if the Master Key is rotated.



You will find a new key created in KeyControl.

7. For the new key to take effect, stop and start db2.

```
% db2stop
% db2start
```

2.6.3. Test access only when KeyControl is available

1. Stop the network services on the IBM DB2 server and try to connect to the database:

```
% db2 connect to MYDB1
```

If KeyControl is not available, the database fails to connect and you see the following error:

```
$ db2 connect to MYDB1
SQL1782N  The command or operation failed because an error was encountered
accessing the centralized key manager. Reason code "5:101".
$
```

2. Restart Network services on the IBM DB2 server and try to connect to the database:

```
% db2 connect to MYDB1

 Database Connection Information

 Database server        = DB2/LINUXX8664 11.5.7.0
 SQL authorization ID   = DB2INST1
 Local database alias   = MYDB1
```

All databases that are encrypted using KeyControl are only accessible when KeyControl is available and Master Key is found.

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

1. Bring down one of the KeyControl nodes and validate you can access the encrypted database.
2. Attempt to connect to the database when one of the KeyControl nodes in the cluster is down:

```
% db2 connect to MYDB1

 Database Connection Information

 Database server        = DB2/LINUXX8664 11.5.7.0
 SQL authorization ID   = DB2INST1
 Local database alias   = MYDB1
```

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.

2.7. Configure the nShield HSM in the KeyControl Server

It is important to note that if you want to use an HSM to further protect the keys using KeyControl, you can configure the HSM in KeyControl. Follow the installation and setup instructions in the `Entrust KeyControl nShield 5.5.1 HSM Integration Guide`.

Chapter 3. Additional resources and related products