IBM DB2 Transparent Data Encryption

nShield® HSM Integration Guide

2026-05-05

Table of Contents

Chapter 1. Introduction

This guide describes how to integrate and use Entrust Security World software and Entrust Security nShield Hardware Security Modules (HSMs) with an IBM Db2 database.

IBM Db2 encrypts databases and backup images using Db2 native encryption. Native encryption provides transparent and secure key management without requiring changes to existing hardware, software, applications, or database schemas.

A key advantage of using a PKCS #11 keystore is the strong protection it provides for encryption keys. This protection is achieved by enforcing the principle that keys never leave the secure boundary of the keystore. Database data at rest is encrypted using a Data Encryption Key (DEK), which is stored with the database.

The DEK itself is encrypted by a master key (MK) that is stored externally to the database. When access is required, the DEK is sent to the PKCS #11 keystore, where it is decrypted using the MK. The only exception to the rule that keys never leave the keystore occurs during the migration of keys from a local keystore file to a PKCS #11 keystore. In this scenario, migrated keys are marked as external. Performing an immediate key rotation after migration transitions usage to internally generated keys within the PKCS #11 keystore.

Using a PKCS #11 keystore is a more secure alternative, particularly in environments with multiple databases where maintaining individual keystores would be complex and error-prone. Using Entrust HSMs to protect the IBM Db2 master key provides the following benefits:

- Secure generation, storage, and protection of encryption keys on FIPS 140-2 Level 3 validated hardware.
- Full lifecycle management of cryptographic keys.
- Comprehensive HSM audit logging.
- Confidence when adopting cloud-based services.
- Improved performance by offloading cryptographic operations from application servers.

1.1. Documents to read first

To install and configure the Entrust HSM, refer to the relevant *User Guide* and *Installation Guide*. You can access them from the Entrust Document Library and from the nShield Product Documentation website.

Also refer to the IBM Db2 online documentation.

1.2. Requirements

Ensure that you are using supported versions of Entrust nShield products, IBM Db2, and third-party products. See Product configurations.

To perform the integration tasks, you must have:

- `root` access on the operating system.
- Access to `nfast` accounts.

Before starting the integration process, familiarize yourself with:

- The documentation for the HSM.
- The documentation and setup process for the IBM Db2 server.

Before using the nShield software, you need to know:

- The number and quorum of Administrator Cards in the Administrator Card Set (ACS), and the policy for managing these cards.
- Whether the application keys are protected by the module or an Operator Card Set (OCS) with or without a pass phrase.
- The number and quorum of Operator Cards in the OCS, and the policy for managing these cards.
- Whether the security world should be compliant with FIPS 140 Level 3.

If using FIPS 140 Level 3, it is advisable to create an OCS for FIPS authorization.

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

For more information, refer to the *User Guide* and *Installation Guide* for the HSM.

1.3. Product configurations

The integration between the IBM Db2 Server and Entrust HSMs has been successfully tested in the following configurations:

Product	Version
Linux	Red Hat Enterprise Linux 9
IBM Db2 Server	12.1.4

1.3.1. Supported nShield hardware and software versions

Entrust has successfully tested the following nShield hardware and software versions:

HSM	Security World Software	Firmware	Image	OCS	Softcard	Module	FIPS Level 3
nShield Connect	13.6.15	12.72.4 (FIPS 140-2 certified)	13.6.15	✓	✓	✓	✓
nShield 5c	13.6.15	13.4.5 (FIPS 140-3 certified)	13.6.15	✓	✓	✓	✓

1.4. Hotfix

It is important to note that this integration requires the hotfix "TAC-1325".

After installing the Security World Software, make sure the hotfix is installed. This hotfix contains an updated version of the nShield pkcs11 library for Security World v13.6.15 on the Linux platform. It addresses an issue where the PKCS#11 CMAC output length was set to half the block size (8 bytes instead of 16 bytes).

Chapter 2. Procedures

2.1. Installation overview

To integrate IBM Db2 with the Entrust HSM using a PKCS11 keystore, you must:

1. Install the IBM Db2 server.
2. Install the HSM.
3. Install the Security World software and create a Security World.
4. Install the TAC-1325 hotfix.
5. Set up /opt/nfast/cknfastrc.
6. Run the `db2server` with the nShield Security World in it..
7. `db2inst1` home directory.
8. Set up a keystore.
9. Verify that encryption is working and that IBM Db2 is using the keystore to manage keys.
10. Migrating from a local keystore to a PKCS #11 keystore..
11. Rotate the master encryption key with IBM Db2.

2.2. Install the IBM Db2 server

See the IBM Db2 online documentation for details on how to install IBM Db2 in your environment.

To provide context for the installation process performed for this guide, here is an example of a IBM Db2 installation on a Red Hat Enterprise Linux 9 server.

2.2.1. Install Docker Engine on the IBM Db2 server

See https://docs.docker.com/engine/install/rhel/ for more information.

1. Uninstall old versions:

```
% sudo dnf remove docker \
                  docker-client \
                  docker-client-latest \
                  docker-common \
                  docker-latest \
                  docker-latest-logrotate \
                  docker-logrotate \
                  docker-engine \
```

```
                podman \
                runc
```

2. Set up the repository:

```
% sudo dnf -y install dnf-plugins-core
% sudo dnf config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
```

3. Install Docker packages:

```
% sudo dnf install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

4. Start Docker engine:

```
% sudo systemctl enable --now docker
```

5. Verify that the installation was successful by running the `hello-world` image:

```
% sudo docker run hello-world
```

2.2.2. Install IBM Db2 on the server

See https://www.ibm.com/docs/en/db2/12.1.x for more information.

1. Create and navigate to a new directory for the Docker image:

```
% mkdir -p ~/Docker
% cd ~/Docker
```

2. Pull the Docker image:

```
% sudo docker pull icr.io/db2_community/db2
```

3. Create a `.env_list` file with the following content:

```
LICENSE=accept
DB2INSTANCE=db2inst1
DB2INST1_PASSWORD=password
DBNAME=testdb
BLU=false
ENABLE_ORACLE_COMPATIBILITY=false
UPDATEAVAIL=NO
TO_CREATE_SAMPLEDB=false
REPODB=false
IS_OSXFS=false
PERSISTENT_HOME=true
```

```
HADR_ENABLED=false
ETCD_ENDPOINT=
ETCD_USERNAME=
ETCD_PASSWORD=
```

4. Run `db2server`:

```
% sudo docker run -h db2server --name db2server --restart=always --detach --privileged=true -p 50000:50000
--env-file .env_list -v /Docker:/database icr.io/db2_community/db2

a0157dd6b59127fde9c4a287436934161dd8da6fdb35b6800bfb3aa471d2925f
```

   Allow up to five minutes for the database to set up properly. If you need to troubleshoot any problems, rerun the command without the `--detach` flag so you can track progress when the command executes.

5. Run the following command to access the Db2 instance that is running in your Docker container:

```
% sudo docker exec -ti db2server bash -c "su - db2inst1"

Last login: Mon Mar 23 18:14:37 UTC 2026
[db2inst1@db2server ~]$
```

6. Create a sample database:

```
% db2sampl -force -sql

  Creating database "SAMPLE"...
  Connecting to database "SAMPLE"...
  Creating tables and data in schema "DB2INST1"...

  'db2sampl' processing complete.
```

7. Connect to the database:

```
% db2 connect to sample


   Database Connection Information

 Database server        = DB2/LINUXX8664 12.1.4.0
 SQL authorization ID   = DB2INST1
 Local database alias   = SAMPLE


% db2 "select * from department"

DEPTNO DEPTNAME                             MGRNO  ADMRDEPT LOCATION
------ ------------------------------------ ------ -------- ----------------
A00    SPIFFY COMPUTER SERVICE DIV.         000010 A00      -
B01    PLANNING                             000020 A00      -
...
I22    BRANCH OFFICE I2                     -      E01      -
```

```
J22    BRANCH OFFICE J2                   -      E01      -

  14 record(s) selected.
```

8. Drop the database that was created:

```
% db2 force applications all
% db2 drop db sample
```

2.3. Install the HSM

Install the HSM by following the instructions in the *Installation Guide* for the HSM.

Entrust recommends that you install the HSM before configuring the Security World software with your IBM Db2 Server.

2.4. Install the Security World software and create a Security World

1. On the computer running the IBM Db2 Server, install the latest version of the Security World software as described in the *Software Installation Guide* for the HSM.

   Entrust recommends that you uninstall any existing nShield software before installing the new nShield software.

2. Create the Security World as described in the *User Guide*, creating the ACS and OCS that you require.

2.5. Install the TAC-1325 hotfix

Get the hotfix from Entrust support and install it as instructed.

2.6. Set up /opt/nfast/cknfastrc

Change the `/opt/nfast/cknfastrc` file according to the protection method being used:

- Module protection

```
CKNFAST_DEBUG=10
CKNFAST_DEBUGFILE=/path/to/debug/file
CKNFAST_OVERRIDE_SECURITY_ASSURANCES=all
CKNFAST_FAKE_ACCELERATOR_LOGIN=1
```

- Softcard/OCS protection

```
CKNFAST_DEBUG=10
CKNFAST_DEBUGFILE=/path/to/debug/file
CKNFAST_OVERRIDE_SECURITY_ASSURANCES=all
CKNFAST_FAKE_ACCELERATOR_LOGIN=1
CKNFAST_LOADSHARING=1
```

Turn debug off in a production environment.

2.7. Run the db2server with the nShield Security World in it.

1. If the db2server is already running, stop it:

```
% sudo docker stop db2server
% sudo docker rm db2server
% sudo rm -rf /Docker
```

2. Run the db2server with the Security World:

```
% cd ~/Docker
% sudo docker run -h db2server --name db2server \
  --restart=always --detach --privileged=true \
  -p 50000:50000 \
  --env-file .env_list \
  -v /Docker:/database \
  -v /opt/nfast:/opt/nfast \
  -v /opt/nfast/kmdata:/opt/nfast/kmdata \
  -v /opt/nfast/toolkits:/opt/nfast/toolkits \
  icr.io/db2_community/db2

4b9b4f9b69f88cfe7cd84fb82ca030c61909d281be23f9f4346647d7f7b9edf5
```

Db2 runs inside the container, but the Security World is on the host. The Db2 PKCS #11 operations must interact directly with: /opt/nfast/toolkits/pkcs11/libcknfast.so.

3. Run the following command to access the Db2 instance that is running in your Docker container:

```
% sudo docker exec -ti db2server bash -c "su - db2inst1"

Last login: Mon Mar 23 18:14:37 UTC 2026
[db2inst1@db2server ~]$
```

2.8. db2inst1 home directory

The "Home" directory for the db2inst1 user depends on the type of IBM Db2 installation

you have. In this case it is set to `/database/config/db2inst1/` so this guide refers to it as `$DB2INST1HOME`.

2.9. Set up a keystore

This section explains how to create the keystore that will be used for encryption by IBM Db2. In our scenario, we have 2 types of keystore:

- Local keystore
- PKCS #11 keystore that uses the HSM.

The use case determines the type of keystore that needs creating. We have the following use cases:

- Generating the master encryption key directly on the HSM.

  This use case only needs the PKCS #11 Keystore.

- Migrating from a local keystore to a PKCS## Keystore.

  This use case needs both types of keystore.

Migration from a local keystore to a PKCS #11 keystore is not supported when using a FIPS-140 Level 3 world.

2.9.1. Local keystore

This type of keystore is native to IBM Db2. In this guide, it is only required when migrating from a local keystore to the the PKCS #11 keystore, which is used by the HSM.

FIPS-140 Level 3 world files can not be used when a local keystore is in place.

If FIPS-140 Level 3 is a requirement, migration is not an option and generation of the master key should be performed directly in a PKCS #11 keystore. Local keystores on IBM Db2 only support FIPS-140 level 2.

Local keystores must be created with the `-fips` option, because Entrust HSMs enforce FIPS-140 Level 2 by default.

Inside the container:

1. Setup the environment for Db2:

```
% . $DB2INST1HOME/sqllib/db2profile
```

2. Create the local keystore by executing the `gsk8capicmd` command.

   To use Db2 native encryption, GSKit must be installed and configured. The Db2 installer installs GSKit locally. For each instance, the GSKit libraries will be located in the `$DB2INST1HOME/sqllib/gskit/bin` directory.

```
% $DB2INST1HOME/sqllib/gskit/bin/gsk8capicmd_64 -keydb -fips -create \
  -db "$DB2INST1HOME/sqllib/security/my-keystore.p12" \
  -pw "ncipher" \
  -type pkcs12 \
  -stash
```

You must use the `-fips` option when creating the local keystore. The keystore password provided in the command is saved in a stash file (`.sth`) with the same base name as the keystore file. Local keystores do not support FIPS-140 Level 3, so should not be used with FIPS-140 Level 3 world files. In this case you should use a PKCS #11 keystore where the master encryption key is generated directtly onto the HSM.

3. Configure the Db2 instance to use the keystore for native encryption by setting the following database manager configuration parameters: `keystore_location` and `keystore_type`.

```
% db2 update dbm cfg using KEYSTORE_LOCATION $DB2INST1HOME/sqllib/security/my-keystore.p12

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.

% db2 update dbm cfg using KEYSTORE_TYPE pkcs12

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.
```

4. Stop and restart Db2:

```
% db2stop
% db2start
```

Occasionally the following error occurs when stopping Db2:

```
04/21/2026 13:02:30     0   0   SQL1025N  The database manager was not stopped
because databases are still active.
SQL1025N  The database manager was not stopped because databases are still active.
```

If you see this error, run the following commands to resolve it:

```
% db2 force applications all
% db2stop
```

2.9.2. PKCS #11 keystore

This type of keystore is used by the HSM.

It is used in the following scenarios:

- Generating the master encryption key straight in the HSM.

  In this scenario you can use a FIPS-140 Level 2 or Level 3 world file.

- Migrating from a local keystore to a PKCS #11 keystore.

  In this scenario you can only use a FIPS-140 Level 2 world file.

To set up the PKCS #11 keystore:

1. Create a PKCS #11 keystore configuration file:

   IBM provides explicit formatting and confirms support for nCipher via `libcknfast.so`: The PKCS #11 LIBRARY path for nCipher is `/opt/nfast/toolkits/pkcs11/libcknfast.so`. PRODUCT_NAME supports “nCipher” for Entrust nShield HSMs.

   a. Inside the container, create and open the configuration file, for example:

   ```
   % vi $DB2INST1HOME/pkcs11.cfg
   ```

   b. Update the file accordingly:

   ```
   VERSION=1
   PRODUCT_NAME=nCipher
   ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP=true
   LIBRARY=/opt/nfast/toolkits/pkcs11/libcknfast.so
   SLOT_LABEL=testSC
   KEYSTORE_STASH=/database/config/db2inst1/sqllib/security/pkcs11_pw.sth
   NEW_OBJECT_TYPE=PRIVATE
   ```

For module protection use **SLOT_LABEL=accelerator**.

2. Create an optional stash file.

   This addresses operational concerns involving access to PKCS #11 credentials. The stash file will store the softcard or OCS password.

```
% db2credman -stash -password ncipher -to $DB2INST1HOME/sqllib/security/pkcs11_pw.sth
% chmod 600 $DB2INST1HOME/sqllib/security/pkcs11_pw.sth
```

3. Configure IBM Db2 to use PKCS #11 keystore:

```
% db2 update dbm cfg using KEYSTORE_LOCATION $DB2INST1HOME/pkcs11.cfg

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.

% db2 update dbm cfg using KEYSTORE_TYPE PKCS11

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.
```

4. Stop and restart Db2:

```
% db2stop
% db2start
```

Occasionally, the following error occurs when stopping Db2:

```
04/21/2026 13:02:30     0   0   SQL1025N  The database manager was not stopped
because databases are still active.
SQL1025N  The database manager was not stopped because databases are still active.
```

If you see this error, run the following commands to resolve it:

```
% db2 force applications all
% db2stop
```

2.10. Verify that encryption is working and that IBM Db2 is using the keystore to manage keys

After configuring IBM Db2 to use the keystore, verify that encryption is working.

If you are not already connected to the docker container running the Db2 server, connect to it before starting the verification process:

```
% sudo docker exec -ti db2server bash -c "su - db2inst1"

Last login: Mon Mar 23 18:14:37 UTC 2026
[db2inst1@db2server ~]$
```

2.10.1. Reset connections

Reset all connections in the database:

```
% db2 QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS;

SQL1224N  The database manager is not able to accept new requests, has
terminated all requests in progress, or has terminated the specified request
because of an error or a forced interrupt.  SQLSTATE=00000

% db2 CONNECT RESET

DB20000I  The SQL command completed successfully.
```

2.10.2. Generate the master encryption key in the keystore.

This process creates a master encryption key in the keystore, regardless of whether it is a local keystore or a PKCS #11 keystore. It occurs when we create an encrypted database.

The database name for the purpose of this guide is `mydb`.

```
% db2 create db mydb encrypt

DB20000I  The CREATE DATABASE command completed successfully.
```

Verify that the database has been successfully encrypted by Db2 and ensure that the value of the `Encrypted database` db configuration parameter value is `YES` in the output of the following command:

```
% db2 get db cfg for mydb

       Database Configuration for Database mydb

 Database configuration release level                  = 0x1600
 Database release level                                = 0x1600

 Update to database level pending                      = NO (0x0)
 Database territory                                    = US
 Database code page                                    = 1208
 Database code set                                     = UTF-8
```

```
 Database country/region code                            = 1
 Database collating sequence                             = IDENTITY
 Alternate collating sequence              (ALT_COLLATE) =
 Number compatibility                                    = OFF
 Varchar2 compatibility                                  = OFF
 Date compatibility                                      = OFF
 Database page size                                      = 4096

.
.
.

 Encrypted database                                      = YES
 Procedural language stack trace         (PL_STACK_TRACE) = NONE
 HADR SSL certificate label              (HADR_SSL_LABEL) =
 HADR SSL Hostname Validation        (HADR_SSL_HOST_VAL) = BASIC

 BUFFPAGE size to be used by optimizer    (OPT_BUFFPAGE) = 0
 LOCKLIST size to be used by optimizer    (OPT_LOCKLIST) = 0
 MAXLOCKS size to be used by optimizer    (OPT_MAXLOCKS) = 0
 SORTHEAP size to be used by optimizer    (OPT_SORTHEAP) = 0
```

If you are using a PKCS #11 keystore, you should see the newly created key in the `/opt/nfast/kmdata/local` folder. When using a local keystore, the key is managed by IBM Db2 internally.

```
% ls -alrt /opt/nfast/kmdata/local

-rw-r--r--. 1 db2inst1 975  7452 Apr 17 14:32 key_pkcs11_ua675d0fcd739aefbfc43616eb767cbbe64d6185ed
```

You can also use the `rocs` utility to check where the key was created in the HSM:

```
% /opt/nfast/bin/rocs
`rocs' key recovery tool
Useful commands: `help', `help intro', `quit'.
rocs> list keys
  No. Name                    App        Protected by
    1 DB2_SYSGEN_db2inst1_MYDB pkcs11     testSC
rocs> exit
```

2.10.3. Exercise the encryption

1. Connect to the database to use the encryption:

```
% db2 connect to mydb

   Database Connection Information

 Database server        = DB2/LINUXX8664 12.1.4.0
 SQL authorization ID   = DB2INST1
 Local database alias   = MYDB
```

2. Run `db2` to launch interactive mode.

```
% db2

(c) Copyright IBM Corporation 1993,2007
Command Line Processor for DB2 Client 12.1.4.0

...

db2 =>
```

3. Create the EMPLOYEE_SALARY table in the database:

```
db2 => CREATE TABLE EMPLOYEE_SALARY (DEPTNO CHAR(3) NOT NULL,DEPTNAME VARCHAR(36) NOT NULL,EMPNO CHAR(6)
NOT NULL,SALARY DECIMAL(9,2) NOT NULL WITH DEFAULT)

DB20000I  The SQL command completed successfully.
```

4. Load the following data into the EMPLOYEE_SALARY table:

```
db2 => INSERT INTO EMPLOYEE_SALARY VALUES (001,'IT',001,10000)
db2 => INSERT INTO EMPLOYEE_SALARY VALUES (001,'IT',002,15000)
db2 => INSERT INTO EMPLOYEE_SALARY VALUES (001,'IT',003,20000)
```

5. Display the contents of the EMPLOYEE_SALARY table:

```
db2 => SELECT * FROM EMPLOYEE_SALARY

DEPTNO DEPTNAME                             EMPNO  SALARY
------ ------------------------------------ ------ -----------
1      IT                                   1         10000.00
1      IT                                   2         15000.00
1      IT                                   3         20000.00

  3 record(s) selected.
```

6. Verify the access to the keystore by moving or renaming the keystore configuration file to break the connection to the database and ensure that it is not available.

   Local Keystore

```
% mv $DB2INST1HOME/sqllib/security/my-keystore.p12 $DB2INST1HOME/sqllib/security/my-keystore.p12.backup
```

   PKCS #11 Keystore

```
% mv $DB2INST1HOME/pkcs11.cfg $DB2INST1HOME/pkcs11.cfg.backup
```

7. Attempt to connect to the database.

   An error message appears because the keystore is not available.

```
% db2 connect to mydb

SQL1781N  An error occurred while parsing the configuration file.
Configuration type: "PKCS11". Reason code: "4".
```

8. Move the keystore configuration file back to the keystore location to make it accessible again.
   This shows that the encryption is working and the master encryption key is secured.

   Local Keystore

   ```
   % mv $DB2INST1HOME/sqllib/security/my-keystore.p12.backup $DB2INST1HOME/sqllib/security/my-keystore.p12
   ```

   PKCS #11 Keystore

   ```
   % mv $DB2INST1HOME/pkcs11.cfg.backup $DB2INST1HOME/pkcs11.cfg
   ```

9. Attempt to connect to the database:

   ```
   % db2 connect to mydb

      Database Connection Information

    Database server        = DB2/LINUXX8664 12.1.4.0
    SQL authorization ID   = DB2INST1
    Local database alias   = MYDB
   ```

10. Read the data in the table:

    ```
    % db2

    db2 => SELECT * FROM EMPLOYEE_SALARY

    DEPTNO DEPTNAME                            EMPNO  SALARY
    ------ ----------------------------------- ------ -----------
    1      IT                                  1         10000.00
    1      IT                                  2         15000.00
    1      IT                                  3         20000.00

      3 record(s) selected.
    ```

    This confirms that encyption is working and that data is only accessible when the keystore configuration is available.

2.10.4. Connection to the HSM (PKCS #11 keystore)

The master encryption key is protected by the HSM when we are using a PKCS #11 keystore. In this section, we are going to break the network connection to the HSM and

attempt to connect to the database and read the encrypted data. This should only work when there is a network connection to the HSM. When the connection is broken, it should fail.

1. Stop the nShield services to intentionally lose the connection to the HSM.

   Do this on the IBM DB2 server that is hosting the IBM DB2 docker container.

   ```
   % sudo /opt/nfast/sbin/init.d-ncipher stop

    -- Running shutdown script 90ncsnmpd

    -- Running shutdown script 75nshieldauditd

    -- Running shutdown script 60raserv

    -- Running shutdown script 50hardserver

    -- Running shutdown script 45nshield5drivers

    -- Running shutdown script 45drivers
   ```

2. Connect to the db2server container:

   ```
   % sudo docker exec -ti db2server bash -c "su - db2inst1"
   ```

3. Attempt to connect to the database.

   It should fail because the HSM is not available.

   ```
   % db2 connect to mydb

   SQL1783N  The command or operation failed because an error was encountered
   accessing the PKCS #11 key manager. Reason code "7:6".
   ```

4. In the IBM Db2 server, restart the nshield services:

   ```
   % sudo /opt/nfast/sbin/init.d-ncipher start

    -- Running startup script 45drivers

    -- Running startup script 45nshield5drivers

    -- Running startup script 50hardserver

    -- Running startup script 60raserv

    -- Running startup script 75nshieldauditd

    -- Running startup script 90ncsnmpd
   ```

5. Connect to the db2server:

```
% sudo docker exec -ti db2server bash -c "su - db2inst1"
```

6. Access the encrypted database.

   You should be able to connect to the database and read the encrypted data.

```
%  db2 connect to mydb

   Database Connection Information

 Database server        = DB2/LINUXX8664 12.1.4.0
 SQL authorization ID   = DB2INST1
 Local database alias   = MYDB

% db2

db2 => select * from employee_salary

DEPTNO DEPTNAME                            EMPNO  SALARY
------ ----------------------------------- ------ -----------
1      IT                                  1         10000.00
1      IT                                  2         15000.00
1      IT                                  3         20000.00

  3 record(s) selected.
```

2.11. Migrating from a local keystore to a PKCS #11 keystore.

This section describes the procedure to migrate the master encryption key from the local keystore to the HSM based PKCS #11 keystore.

Migration from a local keystore to a PKCS #11 keystore is not supported when using a FIPS-140 Level 3 world.

It is assumed that the master key is generated in the local keystore and that the database is already encrypted by the master encryption key in the local keystore. It is also assumed that you are already using the native encryption and have generated the master key in local keystore, and now you want to move to a more secure HSM keystore for securing the master encryption key.

2.11.1. Create the PKCS #11 Keystore

1. Connect to `db2server`:

```
% sudo docker exec -ti db2server bash -c "su - db2inst1"
```

2. Create the IBM Db2 PKCS #11 keystore configuration file:
    a. Inside the container, create and open the configuration file, for example:

```
vi $DB2INST1HOME/pkcs11.cfg
```

    b. Update the file accordingly:

```
VERSION=1
PRODUCT_NAME=nCipher
ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP=true
LIBRARY=/opt/nfast/toolkits/pkcs11/libcknfast.so
SLOT_LABEL=testSC
KEYSTORE_STASH=/database/config/db2inst1/sqllib/security/pkcs11_pw.sth
NEW_OBJECT_TYPE=PRIVATE
```

For module protection use **SLOT_LABEL=accelerator**.

3. Create an optional stash file.

    This addresses operational concerns involving access to PKCS #11 credentials. The stash file will store the softcard or OCS password.

```
% db2credman -stash -password ncipher -to $DB2INST1HOME/sqllib/security/pkcs11_pw.sth
% chmod 600 $DB2INST1HOME/sqllib/security/pkcs11_pw.sth
```

2.11.2. Migrate the master encryption key from the local keystore to the PKCS #11 keystore

1. Backup the local keystore, in case the migration fails:

```
cp $DB2INST1HOME/sqllib/security/my-keystore.p12 $DB2INST1HOME/sqllib/security/my-keystore.p12.backup
```

2. Migrate the key using the `db2p12top11` command:

```
% db2p12top11 -from $DB2INST1HOME/sqllib/security/my-keystore.p12 -to $DB2INST1HOME/pkcs11.cfg -pin ncipher

Migrating keys from </database/config/db2inst1/sqllib/security/my-keystore.p12> local keystore
to PKCS#11 HSM using vendor library </opt/nfast/toolkits/pkcs11/libcknfast.so>
defined in configuration file </database/config/db2inst1/pkcs11.cfg>.

Migrating key: <DB2_SYSGEN_db2inst1_MYDB_2026-04-21-13.04.59_33E7E454> ... Successful.

Out of 1 key(s): 1 key(s) inserted successfully, 0 failed.
```

`ncipher` is the token_password we used. In this case the password is for the softcard or ocs.

3. Verify that the key was created on the HSM:

```
% /opt/nfast/bin/rocs

'rocs' key recovery tool
Useful commands: `help', `help intro', `quit'.
rocs> list keys
  No. Name                    App        Protected by
    1 DB2_SYSGEN_db2inst1_MYDB pkcs11     testSC
rocs> quit
```

If you try to do the migration with a FIPS-140 Level 3 world file, the following error will occur when you use the `db2p12top11` utility:

```
% db2p12top11 -from $DB2INST1HOME/sqllib/security/my-keystore.p12 -to
$DB2INST1HOME/pkcs11.cfg -pin ncipher

Migrating keys from </database/config/db2inst1/sqllib/security/my-keystore.p12>
local keystore
to PKCS#11 HSM using vendor library </opt/nfast/toolkits/pkcs11/libcknfast.so>
defined in configuration file </database/config/db2inst1/pkcs11.cfg>.

Migrating key: <DB2_SYSGEN_db2inst1_MYDB_2026-04-16-18.03.25_61ED442F> ... Failed!

Out of 1 key(s): 0 key(s) inserted successfully, 1 failed.
```

The `pkcs11.log` file shows that it uses a wrapping mechanism that is not supported on a FIPS-140 Level 3 world file: `CKM_AES_ECB`.

```
2026-04-16 18:30:17 [36240]: pkcs11: 000008CB >    hSession 0x000008CB
2026-04-16 18:30:17 [36240]: pkcs11: 000008CB >    pMechanism->mechanism
0x00001081 (CKM_AES_ECB)
2026-04-16 18:30:17 [36240]: pkcs11: 000008CB >    hKey 0x0000045E
2026-04-16 18:30:17 [36240]: pkcs11: 00000000 D    NFC__hash_session 0x000008CB
.
.
.
2026-04-16 18:30:17 [36240]: pkcs11: 00000000 D    NFC__GetMechanismInfoEx type
0x00001081
2026-04-16 18:30:17 [36240]: pkcs11: 000008CB Application error: Not an unwrapping
mechanism
2026-04-16 18:30:17 [36240]: pkcs11: 000008CB <    rv 0x00000070
(CKR_MECHANISM_INVALID)
2026-04-16 18:30:17 [36240]: pkcs11: 000008CB >>   C_GetSessionInfo
```

2.11.3. Set the **ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP** parameter to **FALSE**

In the `$DB2INST1HOME/pkcs11.cfg` file, change the `ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP` parameter value to `false`.

2.11.4. Configure the IBM Db2 instance to use the PKCS #11 keystore

Change the keystore type and location:

```
% db2 update dbm cfg using KEYSTORE_LOCATION $DB2INST1HOME/pkcs11.cfg

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.

% db2 update dbm cfg using KEYSTORE_TYPE PKCS11

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.
```

2.11.5. Restart the database to apply the changes

Run the following commands to stop and restart the database:

```
% db2stop
% db2start
```

Occasionally, the following error occurs when stopping Db2:

```
% db2stop
04/21/2026 13:52:51     0   0   SQL1025N  The database manager was not stopped because
databases are still active.
SQL1025N  The database manager was not stopped because databases are still active.
```

If you see this error, run the following commands to resolve it:

```
% db2 force applications all
% db2stop
```

2.11.6. Verify encryption using the HSM keystore

1. Rename or move the local keystore to ensure that it is not available and the database is only using the master key which is now protected by the HSM:

   ```
   % mv $DB2INST1HOME/sqllib/security/my-keystore.p12 $DB2INST1HOME/sqllib/security/my-keystore.p12.old
   ```

2. Attempt to connect to the database:

```
% db2 connect to mydb
```

3. Run `db2` to launch interactive mode.
4. Display the contents of the **EMPLOYEE_SALARY** table with the following command.

   The command should display the encrypted table in clear text.

```
db2 => SELECT * FROM EMPLOYEE_SALARY

DEPTNO DEPTNAME                            EMPNO  SALARY
------ ----------------------------------- ------ -----------
1      IT                                  1         10000.00
1      IT                                  2         15000.00
1      IT                                  3         20000.00

  3 record(s) selected.
```

5. Verify that the database is accessing the new master key from HSM by running the following command:

```
% db2 get dbm cfg
```

   Ensure that the **KEYSTORE_TYPE** and **KEYSTORE_LOCATION** have the following values:

```
Keystore type (KEYSTORE_TYPE) = PKCS11
Keystore location (KEYSTORE_LOCATION) = /database/config/db2inst1/pkcs11.cfg
```

6. Stop the nShield services to intentionally lose the connection to the HSM.

   Do this on the IBM DB2 server that is hosting the IBM DB2 docker container.

```
% sudo /opt/nfast/sbin/init.d-ncipher stop

 -- Running shutdown script 90ncsnmpd

 -- Running shutdown script 75nshieldauditd

 -- Running shutdown script 60raserv

 -- Running shutdown script 50hardserver

 -- Running shutdown script 45nshield5drivers

 -- Running shutdown script 45drivers
```

7. Connect to the db2server container:

```
% sudo docker exec -ti db2server bash -c "su - db2inst1"
```

8. Attempt to connect to the database.

   It should fail because the HSM is not available.

```
[db2inst1@db2server ~]$ db2 connect to mydb
SQL1783N  The command or operation failed because an error was encountered
accessing the PKCS #11 key manager. Reason code "7:6".
```

9. In the IBM Db2 server, restart the nshield services:

```
% sudo /opt/nfast/sbin/init.d-ncipher start

 -- Running startup script 45drivers

 -- Running startup script 45nshield5drivers

 -- Running startup script 50hardserver

 -- Running startup script 60raserv

 -- Running startup script 75nshieldauditd

 -- Running startup script 90ncsnmpd
```

10. Connect to the db2server:

```
% sudo docker exec -ti db2server bash -c "su - db2inst1"
```

11. Access the encrypted database.

    You should be able to connect to the database and read the encrypted data.

```
% db2 connect to mydb

   Database Connection Information

 Database server        = DB2/LINUXX8664 12.1.4.0
 SQL authorization ID   = DB2INST1
 Local database alias   = MYDB

% db2

db2 => select * from employee_salary

DEPTNO DEPTNAME                             EMPNO  SALARY
------ ------------------------------------ ------ -----------
1      IT                                   1         10000.00
1      IT                                   2         15000.00
1      IT                                   3         20000.00

  3 record(s) selected.
```

2.12. Rotate the master encryption key with IBM Db2

1. List the contents of the database directory:

```
% db2 list db directory

 System Database Directory

 Number of entries in the directory = 2

Database 1 entry:

 Database alias                       = TESTDB
 Database name                        = TESTDB
 Local database directory             = /database/data
 Database release level               = 16.00
 Comment                              =
 Directory entry type                 = Indirect
 Catalog database partition number    = 0
 Alternate server hostname            =
 Alternate server port number         =

Database 2 entry:

 Database alias                       = MYDB
 Database name                        = MYDB
 Local database directory             = /database/data
 Database release level               = 16.00
 Comment                              =
 Directory entry type                 = Indirect
 Catalog database partition number    = 0
 Alternate server hostname            =
 Alternate server port number         =
```

2. Connect to the database:

```
% db2 connect to mydb

   Database Connection Information

 Database server        = DB2/LINUXX8664 12.1.4.0
 SQL authorization ID   = DB2INST1
 Local database alias   = MYDB
```

3. 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
------------------------------------------------------------------------------------------------------------
-------------------- -------------------- ---------------- ---------------- ----------
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
----------------------------------------
------------------------------------------------------------------------------------------------------------
```

```
---------------------------------------------------------------------------------------------------------------
---------------------------------------- -------------------
---------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------
---------------------------------------- ------------ -------------------------------------
---------------
---------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------
-----------------------------------------
---------------------------------------------------------------------------------------------------------------
--------------------
---------------------------------------------------------------------------------------------------------------
-------------------- -------------------------
MYDB
DATABASE             AES              CBC                     256 DB2_SYSGEN_db2inst1_MYDB_2026-04-23-
17.23.07_25EEED6F
/database/config/db2inst1/pkcs11.cfg
PKCS11               testSC                          ;nCipher Corp. Ltd               ;
;644529aad18eeed9;
0                                        Unknown IP Type  DB2_SYSGEN_db2inst1_MYDB_2026-04-23-
17.23.07_25EEED6F
DB2INST1
*LOCAL.db2inst1.260423172307
2026-04-23-17.23.07.000000
  1 record(s) selected.
```

4. 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_MYDB_2026-04-23-17.30.26_4709DC3A

  Return Status = 0
```

5. When the key has rotated, check that the master key was rotated.

   A new key should have been created in the HSM.

```
% ls -alrt /opt/nfast/kmdata/local
...
-rw-r--r--. 1 db2inst1 975  7980 Apr 23 17:23 key_pkcs11_uc644529aad18eeed9de372779e829f48757e617cd-
81cbea9a27138bc4a29f9cec09dbb9c1e60481e9
-rw-r--r--. 1 db2inst1 975  7980 Apr 23 17:30 key_pkcs11_uc644529aad18eeed9de372779e829f48757e617cd-
cc87db324b3c389ff34d2c59d0299956d94effd6

% /opt/nfast/bin/rocs
`rocs' key recovery tool
Useful commands: `help', `help intro', `quit'.
rocs> list keys
  No. Name                    App        Protected by
    1 DB2_SYSGEN_db2inst1_MYDB pkcs11     testSC (testSC)
    2 DB2_SYSGEN_db2inst1_MYDB pkcs11     testSC (testSC)
rocs> q
```

6. Stop and restart db2 to begin using the new key.

```
% db2stop
% db2start
```

Chapter 3. Additional resources and related products

3.1. nShield HSMs

3.2. nShield as a Service

3.3. Entrust products

3.4. nShield product documentation