FORTANIX TECHNICAL DOCUMENTATION

Fortanix®

Integration Guide

USING DATA SECURITY MANAGER WITH ORACLE TDE

VERSION 7.1

Fortanix Inc. | 800 West El Camino Real | Suit 180 | Mountain View, CA 94040 | United States of America | +1 (650) 943-2484 | info@fortanix.com | www.fortanix.com

TABLE OF CONTENTS

1.0 INTRODUCTION

This article describes how to integrate **Fortanix Data Security Manager (DSM) with Oracle Transparent Data Encryption (TDE)**. It also contains the information that a user needs to:

- Generate a TDE master encryption key in Fortanix DSM
- Encrypt a tablespace or columns in a table
- Configure auto-login hardware security module (HSM)

2.0 TERMINOLOGY REFERENCES

1. **Fortanix Data Security Manager**

   Fortanix DSM is the cloud solution secured with Intel® SGX. With Fortanix DSM, you can securely generate, store, and use cryptographic keys and certificates, as well as secrets, such as passwords, API keys, tokens, or any blob of data.

2. **TDE – Transparent Data Encryption**

   Transparent Data Encryption (TDE) enables you to encrypt sensitive data that you store in tables and tablespaces. Oracle Database uses authentication, authorization, and auditing mechanisms to secure data in the database, but not in the operating system data files where data is stored. To protect these data files, Oracle Database provides Transparent Data Encryption (TDE). To prevent unauthorized decryption, TDE stores the encryption keys in a security module external to the database, called a keystore. *For more information, see Introduction to Transparent Data Encryption*.

3. **TDE Table Key**

   TDE Table key is an encryption key that is associated with a table whose columns are marked for encryption. The TDE master encryption key encrypts this table encryption key.

4. **Tablespace Encryption Key**

   A Tablespace Encryption key is an encryption key for the encryption of a tablespace. The TDE master encryption key encrypts the tablespace encryption key, which in turn encrypts and decrypts data in the tablespace.

5. **HSM – Hardware Security Module**

   A Hardware Security Module is an external keystore or a separate server or device that provides secure storage for encryption keys. External keystores are external to an Oracle database. Oracle Database can interface with external keystores but cannot manipulate them outside of the Oracle interface. The Oracle database can request the external keystore to create a key, but it cannot define how this key is stored in an external database.

3.0 PREREQUISITES

- Oracle Database must be on Fortanix DSM-supported versions. Currently, the supported database versions are: **11g R2, 12c, 18c, 19c**. For Oracle 11g, make sure Oracle Database patch **18948524** is applied. This patch enables the Auto-login mode of the HSM wallet.
- Download the latest Fortanix PKCS#11 library from here. Copy it to the database server.
- CA certificate for on-premises installation of Fortanix DSM in PEM format.

4.0 CONFIGURING FORTANIX DATA SECURITY MANAGER FOR TDE

4.1 OBTAINING ACCESS IN FORTANIX DATA SECURITY MANAGER

1. Create an Account in Fortanix DSM if you do not have one already. *See Getting Started for more information*.
2. Create a new Group, for example: “ORACLE TDE”, for storing the TDE master keys.
3. Create an App in Fortanix DSM in the group created in *Step 2* and copy the API key.
   a. In your Fortanix DSM account, go to the **Applications** tab, create a new App in the same group as *Step 2*.
   b. After the app is created, click **COPY API KEY** to copy the API key and save it in a notepad.

5.0 CONFIGURING ORACLE DATABASE FOR INTEGRATION

1. Oracle Database requires PKCS#11 protocol implemented library of the HSM. Copy the downloaded Fortanix library file to the following directory structure.
   `/opt/oracle/extapi/{32,64}/hsm/<vendor_name>/<pkcs_lib_version>`

For example, if your lib version is 3.16.1311, then run the following commands to create the directory and copy the downloaded library file.

```
sudo mkdir -p /opt/oracle/extapi/64/hsm/fortanix/3.16.1311
```

Copy the Fortanix library file to the location above, and name it as `libpkcs11.so`.

```
cp fortanix_pkcs11_3.16.1311.so
/opt/oracle/extapi/64/hsm/fortanix/3.16.1311/ libpkcs11.so
```

Set the permission and ownership of the folder.

```
sudo chown -R oracle:oinstall /opt/oracle

sudo chmod -R 775 /opt/oracle
```

2. Create the configuration file `pkcs11.conf` for connecting to Fortanix DSM.
   The default location is `/etc/fortanix`.
   If the requirement is to use a custom location, then set the environment variable `FORTANIX_PKCS11_CONFIG_PATH` for the path of the custom location. For example:

```
export FORTANIX_PKCS11_CONFIG_PATH=/u01/app/oracle/fortanix/pkcs11.conf
```

Create a file named `pkcs11.conf` in the desired folder with the following parameters:

```
api_endpoint = "https://sdkms.fortanix.com"

api_key = "MWY5YT...TO5n"
```

```
prevent_duplicate_opaque_objects = true
retry_timeout_millis = 3000
[log]
file = "<log filename>"
```

where,

- `api_endpoint` is URL endpoint of the Fortanix DSM installation.
- `api_key` is the API Key you copied to notepad after creating the app.
- `prevent_duplicate_opaque_objects = true`, to prevent creating duplicate opaque objects.
- `retry_timeout_millis` can be set in milliseconds, which allows for retries in case of failures from the service side. By default, this is set to 3 seconds.
- `file` is an optional log file location that can be set. By default, logging is done in `/var/log/syslog`.

3. If Fortanix DSM is installed in an on-premises environment, the CA certificate used for the HTTPS endpoint needs to be provided in PKCS#11 configuration. If not set, the TLS communication to Fortanix DSM will not be successful.
    a. Download the CA certificate as a PEM file on the Oracle Database machine. The file name can be `sdkms-ca.crt`. Make sure the full certificate chain is part of the file.
    b. Add the location of the CA file in PKCS#11 conf file as follows:

```
api_endpoint = "https://sdkms.fortanix.com"
api_key = "MWY5YT...TO5n"
prevent_duplicate_opaque_objects = true
retry_timeout_millis = 3000
ca_certs_file = "/path/to/sdkms-ca.pem"
[log]
file = "<log filename>"
```

4. Test the URL. If it works, then the TLS connection is successful.

```
curl –v <API ENDPOINT>
```

6.0 INTEGRATION STEPS

6.1 ORACLE DATABASE VERSION 19C

6.1.1 SET HARDWARE KEYSTORE TYPE

1. In Oracle 19c, it is not required to edit `sqlnet.ora` file. The setup of the keystore type can be done using Oracle initialization parameters.
2. Set the Oracle wallet location. Even though HSM Wallet does not reside on a local disk, this step is required to set TDE.

```
ALTER SYSTEM SET WALLET_ROOT="$ORACLE_BASE/admin/$ORACLE_SID/wallet"
 scope=spfile;
```

Where `$ORACLE_BASE/admin/$ORACLE_SID/wallet` is the wallet location must be created before running the command. ORACLE_BASE and ORACLE_SID environment variables will be translated based on the database environment variables.

3. Shutdown the system.

```
SQL> shutdown
```

4. Startup the system.

```
SQL> startup
```

5. Set the wallet type to HSM.

```
SQL> ALTER SYSTEM SET TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=HSM"
 scope=both;
```

6.1.2 SET UP KEYSTORE

1. Log in with User `sys` with role `sysdba`.

```
sqlplus / as sysdba
```

To open the hardware key store, run the following command:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "file://
/etc/fortanix/pkcs11.conf" CONTAINER = ALL;
```

In the specification above,

- `IDENTIFIED BY` points to the location of the PKCS#11 Configuration file prefixed with `file://`.
- `CONTAINER` is for use in a multitenant environment. Enter `ALL` to set the keystore in all the pluggable databases (PDBs) in this container database (CDB), or `CURRENT` for the current PDB.

6.1.3 SET TDE MASTER KEY

Next, you must create a TDE master encryption key that is stored inside the Fortanix hardware keystore. Oracle Database uses the TDE master encryption key to encrypt or decrypt TDE table keys or tablespace encryption keys inside the hardware security module

1. Run the following command:

```
ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY file:///etc/fortanix
/pkcs11.conf;

Note: For each pluggable database, master key must be created separa
tely.
```

The command above sets the Oracle database for TDE. Open the Fortanix DSM Web UI and go to the audit log to confirm if the master key was generated and used.

2. Click the **Settings** tab  , and then in the left panel click the **LOG MANAGEMENT** tab to see the audit logs.

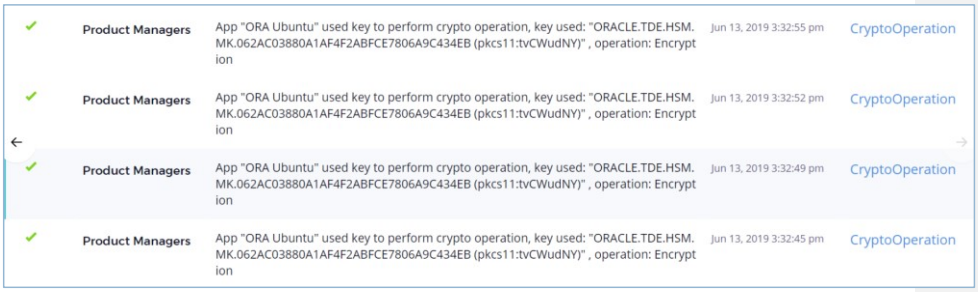

FIGURE 1: AUDIT LOGS

The Fortanix DSM is now successfully integrated with the Oracle TDE.

From here you can proceed to:

1. Start encrypting data. Follow *Section 7.0*.

2. Improve the setup with Auto login. This allows to **auto** open the keystore on database restarts. *See Section 8.1 Auto Login for 19c.*

6.2 FOR ORACLE DATABASE VERSION 12C AND 18C

6.2.1 SET HARDWARE KEYSTORE TYPE IN THE SQLNET.ORA FILE

To configure a keystore for a hardware security module (hardware keystore), you must first define the keystore type in the `sqlnet.ora` file. The Oracle Database checks the `sqlnet.ora` file for the type of keystore and the directory location of the keystore. If not set, it assumes a software keystore by default.

The `sqlnet.ora` file is in the `$ORACLE_HOME/network/admin` directory or in the location set by the `TNS_ADMIN` environment variable. If the file is not present in the directory, create a new one.

Add the following line to the `sqlnet.ora` file to define the hardware keystore type.

```
ENCRYPTION_WALLET_LOCATION = (SOURCE = (METHOD=HSM))
```

6.2.2 SET UP KEYSTORE

1. Log in with User `sys` with role `sysdba`.

```
sqlplus / as sysdba
```

NOTE: If SQL*Plus is already open and you had modified the `sqlnet.ora` file during this time, then reconnect to SQL*Plus. The database session must be changed before the `sqlnet.ora` file changes can take effect.

2. Switch to root container (if it is a container database). The keystore needs to be opened in the root container first. Run the following command in SQL*Plus:

```
ALTER SESSION SET CONTAINER = CDB$ROOT;
```

3. To open the hardware key store, run the following command:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "file:///etc
/fortanix/pkcs11.conf" CONTAINER = ALL;
```

In the specification above,

- `IDENTIFIED BY` points to the location of the PKCS#11 Configuration file prefixed with `file://`.
- `CONTAINER` is for use in a multitenant environment. Enter `ALL` to set the keystore in all the pluggable databases (PDBs) in this container database (CDB), or `CURRENT` for the current PDB.

NOTE: The command above is required to open the keystore for use by the database, at every start of the database.

6.2.3 SET TDE MASTER ENCRYPTION KEY

Next, you must create a TDE master encryption key that is stored inside the Fortanix hardware keystore. Oracle Database uses the TDE master encryption key to encrypt or decrypt TDE table keys or tablespace encryption keys inside the hardware security module

1. Run the following command:

```
ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY file:///etc/fortanix
/pkcs11.conf;

Note: For each pluggable databases, master key needs to be created s
eparately.
```

The command above sets the Oracle database for TDE. Open the Fortanix DSM Web UI and go to the audit log to confirm if the master key was generated and used.

2. Click the **Settings** tab  , and then in the left panel click the **LOG MANAGEMENT** tab to see the audit logs.

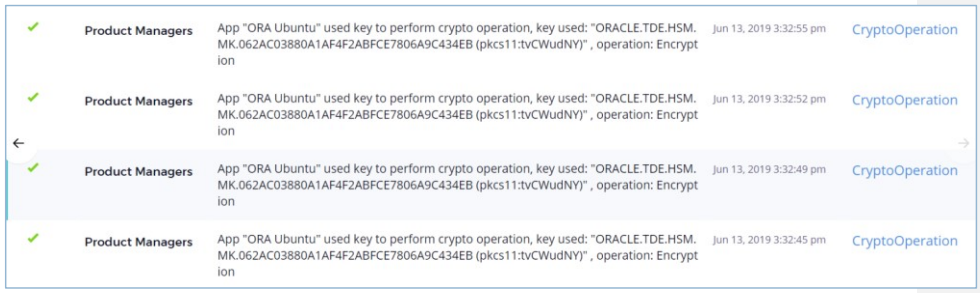

FIGURE 2: AUDIT LOGS

The Fortanix DSM is now successfully integrated with the Oracle TDE.

From here you can proceed to:

1. Start encrypting data. Follow *Section 7.0*.

2. Improve the setup with Auto login. This allows to **auto** open the keystore on database restarts. *See Section 8.2 Auto Login for 12c and 18c.*

6.3 FOR ORACLE DATABASE VERSION 11G

6.3.1 SET HARDWARE KEYSTORE TYPE IN THE SQLNET.ORA FILE

To configure a keystore for a hardware security module (hardware keystore), you must first define the keystore type in the `sqlnet.ora` file. The Oracle Database checks the `sqlnet.ora` file for the type of keystore and the directory location of the keystore. If not set, it assumes a software keystore by default.

The `sqlnet.ora` file is in the `$ORACLE_HOME/network/admin` directory or in the location set by the `TNS_ADMIN` environment variable. If the file is not present in the directory, create a new one.

Add the following line to the `sqlnet.ora` file to define the hardware keystore type.

```
ENCRYPTION_WALLET_LOCATION = (SOURCE = (METHOD=HSM))
```

1. Log in with User `sys` with role `sysdba`.

```
sqlplus / as sysdba
```

NOTE: If SQL*Plus is already open and you had modified the `sqlnet.ora` file during this time, then reconnect to SQL*Plus. The database session must be changed before the `sqlnet.ora` file changes can take effect.

6.3.2 OPEN KEYSTORE

1. To open the hardware key store, run the following command:

```
ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY "file:///etc/fortanix/pkcs11.conf";
```

In the specification above,

- `IDENTIFIED BY` points to the location of the PKCS#11 Configuration file prefixed with `file://`.

NOTE: The command above is required to open the keystore for use by the database, at every start of the database.

6.3.3 SET TDE MASTER ENCRYPTION KEY

Next, you must create a TDE master encryption key that is stored inside the Fortanix hardware keystore. Oracle Database uses the TDE master encryption key to encrypt or decrypt TDE table keys or tablespace encryption keys inside the hardware security module

1. Run the following command:

```
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY "file:///etc/fortanix/
pkcs11.conf";
```

The command above sets the Oracle database for TDE. Open the Fortanix DSM Web UI and go to the audit log to confirm if the master key was generated and used.

2. Click the **Settings** tab  , and then in the left panel click the **LOG MANAGEMENT** tab to see the audit logs.

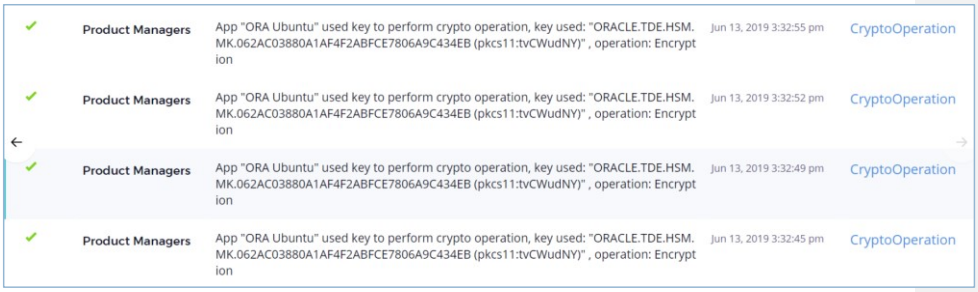

FIGURE 3: AUDIT LOGS

The Fortanix DSM is now successfully integrated with the Oracle TDE.

From here you can proceed to:

1. Start encrypting data. Follow *Section 7.0*.

2. Improve the setup with Auto login. This allows to **auto** open the keystore on database restarts. *See Section 8.3 Auto Login for 11g.*

7.0 ENCRYPT YOUR DATA

Once the TDE is set up on the Oracle Database, we can proceed with data encryption. There are 2 ways of encrypting data:

1. **Column Encryption**: Selected columns of a table to be encrypted.
2. **Tablespace Encryption**: Encrypt the entire tablespace using TDE. This is recommended by Oracle due to its better performance.

7.1 COLUMN ENCRYPTION

1. Connect to SQL*plus as a non-sysadmin user to enable encryption on a table.
2. Next, create a table with an encrypted column. Use the `CREATE TABLE` SQL statement with the `ENCRYPT` clause.

```
CREATE TABLE employee (first_name VARCHAR2(128), last_name VARCHAR2(
128), empID NUMBER, salary NUMBER(6) ENCRYPT);
```

3. Now insert some data into the table.

```
INSERT INTO employee VALUES ('JOHN', 'SMITH',001, 10000);
```

4. To list the encrypted columns in your database, run the following command:

```
select * from dba_encrypted_columns;

OWNER    TABLE_NAME  COLUMN_NAME  ENCRYPTION_ALG    SALT   INTEGRITY_ALG
-------  ----------  -----------  --------------    ----   -------------
OE       EMPLOYEE     SALARY      AES 192 bits key   YES    SHA-1
```

More details can be found here https://docs.oracle.com/database/121/TDPSG/GUID-61259237-5514-4531-AFB4-CF716F93F1E5.htm#TDPSG44324.

7.2 TABLESPACE ENCRYPTION

1. Connect to SQL*plus as a non-sysadmin user to create a new encrypted tablespace.
2. Make sure that `COMPATIBLE` Initialization parameter setting is `11.2.0.0` or higher. This is typically an issue with older 11g databases.

```
SHOW PARAMETER COMPATIBLE
NAME                                 TYPE        VALUE
compatible                           string      11.2.0.0
noncdbcompatible                     BOOLEAN     FALSE

// If above is less than 11.2.0.0, then run following
alter system set COMPATIBLE='11.2.0.0' scope=spfile;
shutdown
startup
```

3. Create a new encrypted tablespace using `CREATE TABLESPACE` SQL statement with the `ENCRYPT` clause.

```
CREATE TABLESPACE encrypted_ts
DATAFILE '/u01/app/oracle/oradata/DB11G/encrypted_ts01.dbf' SIZE 128K
AUTOEXTEND ON NEXT 64K
ENCRYPTION USING 'AES256'
DEFAULT STORAGE(ENCRYPT);

ALTER USER test QUOTA UNLIMITED ON encrypted_ts;
```

4. Now insert some data into the table or tablespace.

```
CREATE TABLE ets_test (
  id    NUMBER(10),
  data  VARCHAR2(50)
) TABLESPACE encrypted_ts;
```

```
INSERT INTO ets_test (id, data) VALUES (1, 'This is a secret!');
COMMIT;
```

5. The `ENCRYPTED` column of the `DBA_TABLESPACES` and `USER_TABLESPACES` views indicate if the tablespace is encrypted or not.

```
SELECT tablespace_name, encrypted FROM dba_tablespaces;

TABLESPACE NAME                ENC
------------------------------ ---
SYSTEM                         NO
SYSAUX                         NO
ENCRYPTED_TS                   YES

3 rows selected.
```

8.0 CONFIGURING AUTO-LOGIN

The method described in ***Section 6.0*** requires saving the Fortanix DSM API key in a config file. If you do not want to expose the API key in a file, you can configure the HSM key store to use auto-login using App ID and Password. The advantages would be:

1. The API Key is not exposed to the database machine.

2. In the case of database restart, the wallet is opened automatically.

In this method we will store the Fortanix DSM App's secret/password in an auto-login keystore using a software keystore and the App Id (App UUID) will be stored in a config file. The Oracle TDE will pass the app secret stored in the auto-login keystore as PIN to the Fortanix DSM PKCS11 library. Fortanix DSM KMS PKCS11 library will use this PIN and combine it with App Id in the config file to create the basic authentication token for authenticating to Fortanix DSM.

For additional details on Oracle TDE and auto-login keystore, please see:
https://docs.oracle.com/database/121/ASOAG/managing-keystore-and-tde-master-encryption-key.htm#ASOAG10434

8.1 AUTO-LOGIN IN ORACLE 19C

NOTE: Complete the Steps for HSM Wallet as per ***Section 6.1***, before proceeding with auto-login steps. The TDE master key needs to be created before auto-login can be set.

1. Check if the HSM key store is open. You can check the status of whether a keystore is open or closed by querying the `STATUS` column of the `V$ENCRYPTION_WALLET` view.

```
Select * from V$ENCRYPTION_WALLET
```

2. Close the HSM keystore if it is open.

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE IDENTIFIED BY "file:///
etc/fortanix/pkcs11.conf" CONTAINER = ALL;
```

3. Create a directory for the software wallet.

```
mkdir $ORACLE_BASE/admin/$ORACLE_SID/wallet/tde
```

   Where the wallet location should resemble **`$WALLET_ROOT/tde`**.

4. Change the keystore type to a software wallet.

```
ALTER SYSTEM SET TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=FILE"
```

5. Create a software keystore. This keystore will store the password to the HSM wallet for the purpose of auto-login.

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '$ORACLE_BASE/admin/$ORACL
E_SID/wallet/tde' IDENTIFIED BY
```

```
"<SDKMS_APP_PASSWORD>";
```

NOTE:

- Provide a password of at least 8 characters above. This will be your software wallet password. Remember it for future use.
- From Oracle version 12.2 onwards you can use the `FORCE` option for key rotation with the auto-login enabled wallet. For the 12.2 database and above, it is also required to keep the wallet password same as "`<SDKMS_APP_PASSWORD>`".

6. Open the software keystore.

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "<SDKMS_AP
P_PASSWORD>";
```

7. Add Fortanix DSM App password as a secret in the software wallet using the client as `HSM_PASSWORD`. This is an Oracle-defined client name that is used to represent the HSM password as a secret in the software keystore. The app password can be fetched as per *Section 4.1*.

```
ADMINISTER KEY MANAGEMENT ADD SECRET '<SDKMS_APP_PASSWORD>' FOR CLIE
NT 'HSM_PASSWORD' IDENTIFIED BY "<SDKMS_APP_PASSWORD>" WITH BACKUP;
```

8. Edit the existing PKCS#11 config file:
   - Remove API Key if exists in the file.
   - Add App Id. This value can be fetched from *Section 4.1*.

   The updated contents of `pkcs11.conf` is:

```
api_endpoint = "https://sdkms.fortanix.com"
app_id = "<App UUID>"
prevent_duplicate_opaque_objects = true
```

```
retry_timeout_millis = 3000
[log]
file = "<log filename>"
```

NOTE: The rest of the contents in `pkcs11.conf` file can be kept same. For example, `ca_certs`, `cert`, `key`, etc. This means the configuration is applicable for certificate based authenticate mode too.

9. Close the software keystore.

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE IDENTIFIED BY "<SDKMS_A
PP_PASSWORD>";
```

10. Now create the auto-login keystore.

```
ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE '
$ORACLE_BASE/admin/$ORACLE_SID/wallet/tde IDENTIFIED BY "<SDKMS_APP_
PASSWORD>";
```

11. Set the wallet location to HSM backed by auto-login.

```
ALTER SYSTEM SET TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=HSM|FILE"
```

12. Shut down and restart the database for changes to pick up correctly.

13. Run the following command to verify that the wallet is auto-open.

```
SELECT * FROM V$ENCRYPTION_WALLET;
```

This completes the auto-login steps for Oracle 19c.

8.2 AUTO-LOGIN IN ORACLE 12C AND 18C

NOTE: Complete the Steps for HSM Wallet as per *Section 6.2*, before proceeding with auto-login steps. The TDE master key needs to be created before auto-login can be set.

1. Check if the HSM key store is open. You can check the status of whether a keystore is open or closed by querying the `STATUS` column of the `V$ENCRYPTION_WALLET` view.

```
Select * from V$ENCRYPTION_WALLET
```

2. Close the HSM keystore if it is open.

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE IDENTIFIED BY "file:///
etc/fortanix/pkcs11.conf" CONTAINER = ALL;
```

3. Create a directory for the software wallet.

```
mkdir /opt/oracle/admin/ORCLCDB/wallet/Fortanix
```

Where **`$ORACLE_BASE/admin/$ORACLE_SID/wallet/Fortanix`** is the wallet location.

4. Now create a software keystore. This keystore will store the password to the HSM wallet for the purpose of auto-login. Make sure you use "`<SDKMS_APP_PASSWORD>`" as the wallet password in this step.

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE "/opt/oracle/admin/ORCLCDB
/wallet/Fortanix" IDENTIFIED BY "<SDKMS_APP_PASSWORD>";
```

NOTE:

- Provide a password of at least 8 characters above. This will be your software wallet password. Remember it for future use.

- From Oracle version 12.2 onwards you can use the `FORCE` option for key rotation with the auto-login enabled wallet. For the 12.2 database and above, it is also required to keep the wallet password the same as "`<SDKMS_APP_PASSWORD>`".

5. Reconfigure the `sqlnet.ora` file and add the keystore location of the software keystore created in *Step 3* to the `DIRECTORY` setting of the `ENCRYPTION_WALLET_LOCATION` setting. For example:

```
ENCRYPTION_WALLET_LOCATION=
 (SOURCE=(METHOD=FILE)(METHOD_DATA=
   (DIRECTORY=/opt/oracle/admin/ORCLCDB/wallet/Fortanix)))
```

6. Reconnect to the database, or log out and then log back in again, so that the change that you made in the previous step takes effect.
7. Open the software keystore.

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY password;
```

8. Add Fortanix DSM App password as a secret in the software wallet using the client as `HSM_PASSWORD`. This is an Oracle-defined client name that is used to represent the HSM password as a secret in the software keystore. The app password can be fetched as per *Section 4.1*.

```
ADMINISTER KEY MANAGEMENT ADD SECRET '<SDKMS_APP_PASSWORD>' FOR CLIE
NT 'HSM_PASSWORD' IDENTIFIED BY "<SDKMS_APP_PASSWORD>" WITH BACKUP;
```

9. Edit the existing PKCS#11 config file:
   - Remove API Key if it exists in the file.
   - Add App Id. This value can be fetched from Section 4.1.

   The updated contents of `pkcs11.conf` is:

```
api_endpoint = "https://sdkms.fortanix.com"
app_id = "<App UUID>"
prevent_duplicate_opaque_objects = true
retry_timeout_millis = 3000
[log]
file = "<log filename>"
```

NOTE: The rest of the contents in `pkcs11.conf` file can be kept the same. For example, `ca_certs`, `cert`, `key`, etc. This means the configuration is applicable for certificate based authenticate mode too.

10. Close the software keystore.

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE IDENTIFIED BY "<SDKMS_APP_PASSWORD>";
```

11. Now create the auto-login keystore.

```
ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE '/opt/oracle/admin/ORCLCDB/wallet/Fortanix' IDENTIFIED BY "<SDKMS_APP_PASSWORD>";
```

12. Update the `sqlnet.ora` file to use the hardware security module location. Comment the previously configured wallet location (adjust `DIRECTORY` to the path where your wallet is stored).

```
ENCRYPTION_WALLET_LOCATION = (SOURCE = (METHOD=HSM)(METHOD_DATA=(DIRECTORY=/opt/oracle/admin/ORCLCDB/wallet/Fortanix)))
```

13. Shutdown and restart the database for changes to pick up.

14. Run the following command to verify that the wallet is auto-open.

```
SELECT * FROM V$ENCRYPTION_WALLET;
```

This completes the auto-login steps for Oracle 12c and 18c.

8.3 AUTO LOGIN IN ORACLE 11G

NOTE: Complete the Steps for HSM Wallet as per *Section 6.3*, before proceeding with auto-login steps. The TDE master key needs to be created before auto-login can be set.

1. Close the HSM keystore if it is open.

```
ALTER SYSTEM SET KEYSTORE CLOSE IDENTIFIED BY "file:///etc/fortanix/
pkcs11.conf" CONTAINER = ALL;
```

2. Create a new entry in `sqlnet.ora` file (Replace the existing WALLET setting).

```
ENCRYPTION_WALLET_LOCATION = (SOURCE = (METHOD = HSM) (METHOD_DATA =
 (DIRECTORY = "$ORACLE_BASE/admin/$ORACLE_SID/wallet/tde")))
```

   Where **`$ORACLE_BASE/admin/$ORACLE_SID/wallet/tde`** is the wallet location.

3. Create an auto-open and encryption wallet in the directory mentioned above:

```
orapki wallet create -wallet . -auto_login -pwd <password>
```

   Where "." signifies the current directory.

   For example,

```
orapki wallet create -wallet . -auto_login -pwd <SDKMS_APP_PASSWORD>
```

NOTE: When prompted for a password, provide a password of at least 8 characters. This will be your software wallet password. Remember it for future use.

4. Add the following entry to the empty wallets to enable an 'auto-open HSM':

```
mkstore -wrl . -createEntry ORACLE.TDE.HSM.AUTOLOGIN <any-non-empty-string>
```

Where,

- `<any-non-empty-string>` could be any string and it is used only one time.

For example:

```
mkstore -wrl . -createEntry ORACLE.TDE.HSM.AUTOLOGIN TESTDE
```

This command will prompt for a password, so enter the software wallet password set earlier.

a. Oracle opens the encryption wallet first and if not present then it will open the auto wallet. Rename the encryption wallet (`ewallet.p12`) or move it out of the '`ENCRYPTION_WALLET_LOCATION`' defined in the '`sqlnet.ora`' file to a secure location;

IMPORTANT: Do not delete the encryption wallet and do not forget the wallet password.

For example:

```
mv ewallet.p12 ewallet.p12.orig
```

5. Edit the existing PKCS#11 config file:
   - Remove API Key if exists in the file.
   - Add App Id. This value can be fetched from *Section 4.1*.

The updated contents of `pkcs11.conf` is:

```
api_endpoint = "https://sdkms.fortanix.com"
```

```
app_id = "<App UUID>"
prevent_duplicate_opaque_objects = true
retry_timeout_millis = 3000
[log]
file = "<log filename>"
```

NOTE: The rest of the contents in the `pkcs11.conf` file can be kept the same. For example, `ca_certs`, `cert_file`, `key_file`, and so on. This means the configuration is applicable for certificate-based authenticate mode too.

6. Create a TDE master encryption key inside the HSM using App password. The password can be fetched as per *Section 4.1*.

```
ALTER system set encryption key identified by "<SDKMS_APP_PASSWORD>";
```

This will insert "`SDKMS_APP_PASSWORD`" into the auto-login wallet. From now on, no password is required to access encrypted data with the TDE master encryption key stored in an HSM.

7. Run the following command to check the wallet open status.

```
SELECT * FROM V$ENCRYPTION_WALLET;
```

8. Shut down the database immediately.

9. Startup the database.

10. Run the following command to verify that the wallet is auto-open.

```
SELECT * FROM V$ENCRYPTION_WALLET;
```

This completes the auto-login steps for Oracle 11g.

8.4 ROTATE MASTER KEY

Use the following commands to rotate a master key with/without auto-login wallet using a container and non-container database.

8.4.1 WITHOUT USING AUTO-LOGIN WALLET

Using container database:

```
ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY "file:///etc/fortan
ix/pkcs11.conf" CONTAINER = ALL;
```

Using non-container database:

```
ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY "file:///etc/fortan
ix/pkcs11.conf";
```

8.4.2 USING AUTO LOGIN WALLET

Using container database:

```
ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY "DSM
_APP_PASSWORD" CONTAINER = ALL;
```

 **NOTE**: In few database versions, you may need to run this command within each pluggable database to rotate the individual master key.

Using non-container database:

```
ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY "DSM
_APP_PASSWORD";
```

NOTE: After the master key is rotated, the older master key may be accessed by the database Gen0 process to perform heartbeat operations. This key will continue to be used until the next database restart where the Gen0 process will pick up the most recent master key and perform the heartbeat operation.

9.0 ENABLE PREFETCHING RESPONSES OF UPCOMING HEARTBEAT REQUESTS LOCALLY (OPTIONAL)

Commented [NR1]: Enabling where ?

To ensure connection to the Fortanix DSM, the Oracle Database makes repeated requests for encrypting dummy data. These requests are called heartbeats. By default, the heartbeat is created every 3 seconds, and if the database server does not receive a response within the time limit, the wallet closes. But a delay of 3 seconds can happen even due to factors like minor outages or network latency.

Commented [NR2]: What is secs?

Commented [NR3]: We don't use short forms, refer to our style guide

So, in cases of connectivity loss, the results for the upcoming heartbeats are prefetched and used. Note that this is just used to delay the wallet closure. If the connectivity is not restored before the prefetched values are used up, the wallet will still close but only after a delay.

If you are expecting a network latency in the communication channel between Fortanix DSM and Oracle Database Server, you can (optionally) enable the prefetching of heartbeats functionality by specifying the following in the PKCS#11 configuration file:

```
api_endpoint = "https://sdkms.fortanix.com"
api_key = "MWY5YT...TO5n"
prevent_duplicate_opaque_objects = true
retry_timeout_millis = 3000
oracle_tde_cache_config = {} #enabling prefetching of heartbeats
[log]
file = "<log filename>"
```

For more information on enabling prefetching of heartbeats, refer to Clients:PKCS#11 Library.

10.0 DOCUMENT INFORMATION

10.1 DOCUMENT LOCATION

The latest published version of this document is located at the URL:

https://support.fortanix.com/hc/en-us/articles/360016188991-Using-Fortanix-Data-Security-Manager-with-Oracle-TDE

10.2 DOCUMENT UPDATES

This document will typically be updated on a periodic review and update cycle.

For any urgent document updates, please send an email to: support@fortanix.com