

# Enabling TDE encryption in an Oracle 19c DB

## 1. Edit SQLNET.ORA file

```
[oracle@node1 encryption_keystore]$ cat $ORACLE_HOME/network/admin/sqlnet.ora  
ENCRYPTION_WALLET_LOCATION =  
  (SOURCE =(METHOD = FILE) (METHOD_DATA =  
    (DIRECTORY = /oracle/app/oracle/admin/PRODCDB/encryption_keystore/)))  
[oracle@node1 encryption_keystore]$  
[oracle@node1 encryption_keystore]$
```

## 2. Configure wallet parameters

Connect to the root container and create the keystore.

```
SQL> alter system set wallet_root='/oracle/app/oracle/admin/PRODCDB/encryption_keystore'  
scope=spfile;
```

```
SQL> alter system set wallet_root='/oracle/app/oracle/admin/PRODCDB/encryption_keystore' scope=spfile;  
System altered.
```

```
SQL> ALTER SYSTEM SET TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=FILE" SCOPE = BOTH;
```

```
SQL> ALTER SYSTEM SET TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=FILE" SCOPE = BOTH;  
System altered.
```

```
SQL>  
SQL> show parameter wallet_root  


| NAME        | TYPE   | VALUE                                                |
|-------------|--------|------------------------------------------------------|
| wallet_root | string | /oracle/app/oracle/admin/PRODCDB/encryption_keystore |

  
SQL> show parameter TDE_CONFIGURATION  


| NAME              | TYPE   | VALUE                       |
|-------------------|--------|-----------------------------|
| tde_configuration | string | KEYSTORE_CONFIGURATION=FILE |

  
SQL>
```

## 3. Create Key store

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE  
'/oracle/app/oracle/admin/PRODCDB/encryption_keystore' IDENTIFIED BY mypassword;
```

```
SQL>
SQL> ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/oracle/app/oracle/admin/PRODCDB/encryption_keystore' IDENTIFIED BY mypassword;
keystore altered.

SQL>
SQL> !ls -lrth
total 4.0K
-rw----- 1 oracle asmadmin 2.5K Mar  2 12:16 ewallet.p12

SQL>
SQL> !pwd
/oracle/app/oracle/admin/PRODCDB/encryption_keystore
```

#### 4. Convert keystore to AUTOLOGIN

ADMINISTER KEY MANAGEMENT CREATE AUTO\_LOGIN KEYSTORE FROM KEYSTORE  
'/oracle/app/oracle/admin/PRODCDB/encryption\_keystore' IDENTIFIED BY mypassword;

```
SQL> ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE '/oracle/app/oracle/admin/PRODCDB/encryption_keystore' IDENTIFIED BY mypassword;
keystore altered.

SQL>
SQL>
SQL> !ls -lrth
total 8.0K
-rw----- 1 oracle asmadmin 2.5K Mar  2 12:16 ewallet.p12
-rw----- 1 oracle asmadmin 2.6K Mar  2 12:17 cwallet.sso

SQL> !pwd
/oracle/app/oracle/admin/PRODCDB/encryption_keystore
```

#### 5. Open keystore

SQL> administer key management set keystore open identified by mypassword CONTAINER=ALL;

```
SQL>
SQL> administer key management set keystore open identified by mypassword CONTAINER=ALL;
keystore altered.
```

#### 6. Set master encryption key

SQL> ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY mypassword WITH  
BACKUP;

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY mypassword WITH BACKUP;
keystore altered.
```

Run same command by connecting to PDB DB

```
SQL>
SQL> ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY mypassword WITH BACKUP;
keystore altered.

SQL>
```

#### 7. Close Wallet

- administer key management set keystore close identified by mypassword CONTAINER=ALL;

```
SQL> select a.con_id, wallet_type, wrl_parameter, status from v$encryption_wallet a, v$pdbs b where a.con_id=b.con_id(+);
```

CON_ID	WALLET_TYPE	WRL_PARAMETER	STATUS
2	AUTOLOGIN		OPEN
3	AUTOLOGIN		OPEN
1	AUTOLOGIN	/oracle/app/oracle/admin/PRODCDB/encryption_keystore/tde/	OPEN

## 8. Encrypting New Tablespaces

Run the `CREATE TABLESPACE` the statement, using its encryption clauses.

```
SQL>
SQL> CREATE TABLESPACE ENC_TABLESPACE
DATAFILE '+DATA' SIZE 100M
ENCRYPTION USING 'AES256' ENCRYPT; 2 3

Tablespace created.
```

## 9. Encrypting Future Tablespaces

You can set the `ENCRYPT_NEW_TABLESPACES` database initialization parameter to automatically encrypt future tablespaces that you create.

- SQL> alter system set "\_tablespace\_encryption\_default\_algorithm" = 'AES256' scope = both;
- alter system set encrypt\_new\_tablespaces = ALWAYS scope = both;

```
SQL>
SQL> alter system set "_tablespace_encryption_default_algorithm" = 'AES256' scope = both;

System altered.

SQL> alter system set encrypt_new_tablespaces = ALWAYS scope = both;

System altered.
```

## 10. Adding encryption algorithm for the SYSTEM tablespace:

- alter tablespace SYSTEM encryption ONLINE encrypt;

```
SQL> alter tablespace SYSTEM encryption ONLINE encrypt;

Tablespace altered.
```

```
SQL> select TABLESPACE_NAME,STATUS,ENCRYPTED from dba_tablespaces;
```

TABLESPACE_NAME	STATUS	ENC
SYSTEM	ONLINE	YES
SYSAUX	ONLINE	NO
UNDOTBS1	ONLINE	NO
TEMP	ONLINE	NO
USERS	ONLINE	NO
UNICODE_REPO	ONLINE	NO
ENC_TABLESPACE	ONLINE	YES

```
7 rows selected.
```

Notes:

- You must set the compatible, wallet\_root and TDE\_CONFIGURATION initialization parameters on all instances of the database (RAC or standby nodes) before creating an encrypted tablespace.
- Copy the wallet directory to all nodes in case of Oracle RAC if the wallet is not created in an ASM disk group location.
- Copy the wallet to all standby nodes as well as any DR nodes.

-----END-----