Transparent Data Encryption 19c

Who we are

Experts At Your Service

> Over 70 specialists in IT infrastructure

> Certified, experienced, passionate

Based In Switzerland

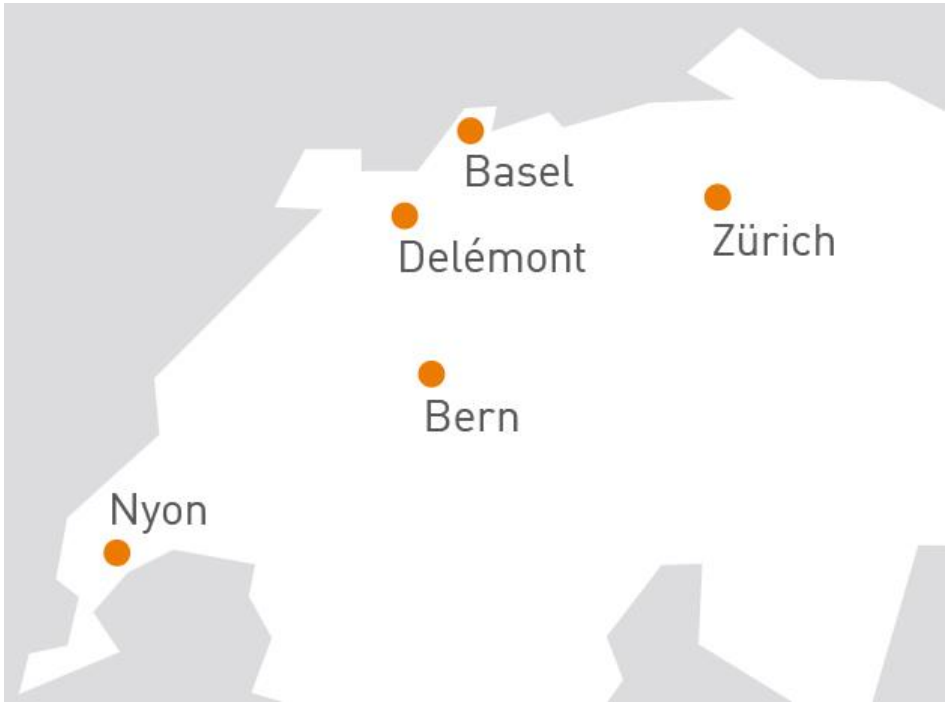
> 100% self-financed Swiss company

> Over CHF 12.3 mio. Turnover

Leading In Infrastructure Services

> More than 200 customers in CH and EU

> Over 60 SLAs dbi FlexService contracted



About me

Thomas Rein

Senior Consultant

+41 78 901 67 05

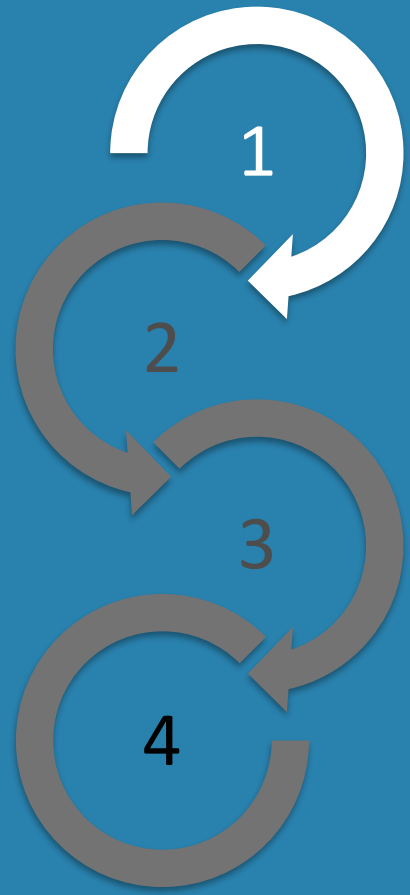
thomas.rein@dbi-services.com

Agenda

1. Introduction into TDE
2. TDE configuration (On Premises)
3. TDE configuration (OCI)
4. Conclusion

Introduction into TDE

- What is TDE
- Licensing
- Evolution of TDE
- What's new in 18c & 19c



Transparent Data Encryption

Transparent encryption, also known as real-time encryption and on-the-fly encryption (OTFE), is a method used by some disk encryption software. "Transparent" refers to the fact that data is automatically encrypted or decrypted as it is loaded or saved.

....

In general, every method in which data is seamlessly encrypted on write and decrypted on read, in such a way that the user and/or application software remains unaware of the process, can be called transparent encryption.

Source: Wikipedia

Changes in Database Editions / Offerings

- Classical Editions (on premises)
  - Personal Edition (PE)
  - Standard Edition 2 (SE2)
  - Enterprise Edition (EE)
- Oracle Database Cloud Service (OCI)
  - Standard Edition (DBCS SE)
  - Enterprise Edition (DBCS EE)
  - Enterprise Edition High Performance (DBCS EE-HP)
  - Enterprise Edition Extreme Performance (DBCS EE-EP)
- TDE is part of Advanced Security Option
  - Not included in SE
  - Additional cost EE
  - Included in PE, DBCS (SE, EE, EE-HP and EE-EP) without additional costs

Introduction into TDE
Licensing

Advanced Security Option

- Transparent Data Encryption (TDE) for tablespaces and columns (including Oracle SecureFiles)
- DataPump Export File encryption
- RMAN backup encryption to disk
- TDE master key storage in an Oracle Wallet or external Hardware Security Module
- Data Redaction of sensitive data returned to applications (Full, Partial, Regular Expression, and Random techniques)
- Network encryption and authentication services are no longer part of Oracle Advanced Security and are available in all licensed editions of all supported releases of Oracle Database

Introduction into TDE
Evolution of TDE

Encryption SQL*Net
DBMS_OBFUSCATION_TOOLKIT

10.1 DBMS_CRYPTO

10.2 (TDE)
Column Encryption
ALTER SYSTEM SET ENCRYPTION WALLET
ALTER SYSTEM SET KEY

11.1
Tablespace Encryption (New Tablespaces)
Encrypted Dump File Sets
Hardware-Based Master Key Encryption

11.2
Unified Master Keys Column/Tablespace Encryption
Master rekey possible

12.1
TDE CDB & PDB
ADMINISTER KEY MANAGEMENT
Export / import of keys

12.2
Encrypt existing tablespaces
Fully encrypt databases

Keystore for each pluggable database
User-Defined Master Encryption Key
Encrypted passwords for database links
with Oracle Data Pump

Close of keystore even (SYSTEM, SYSAUX, TEMP and UNDO is encrypted
Transparent online conversion support for non OMF datafiles
New offline encryption formats (GOST, ARIA, 3DES)

Introduction into TDE
What's new in 18c / 19c

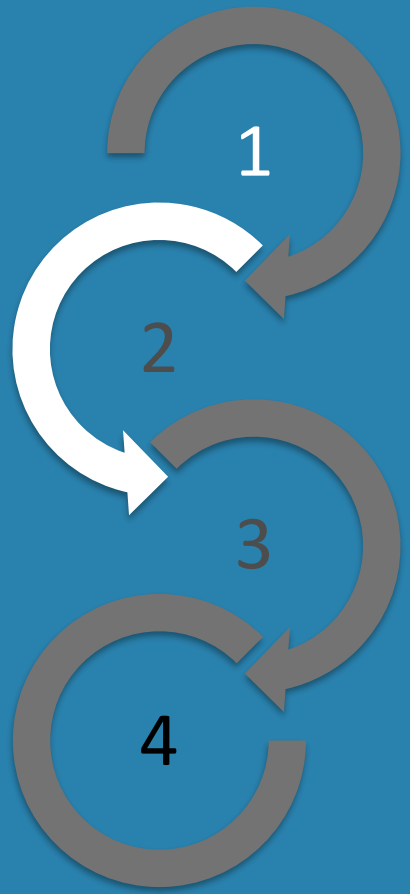
General
> Static parameter WALLET_ROOT to specify the keystore path
> SQLNET.ORA no longer needed
> Dynamic parameter TDE_CONFIGURATION to specify the type of keystore (Software/Hardware)
> User defined master keys (bring your own key)

Multitenant environments
> Keystore in united mode (one Keystore for CDB & PDBs)
> Keystore in isolated mode (each PDB has its own keystore)

New with 19c
> Close of keystore even if SYSTEM, SYSAUX, TEMP and UNDO is encrypted
> Easier online conversion for non OMF datafiles

TDE configuration (On Premises)

- Parameters
- Configuration workflow
- Examples

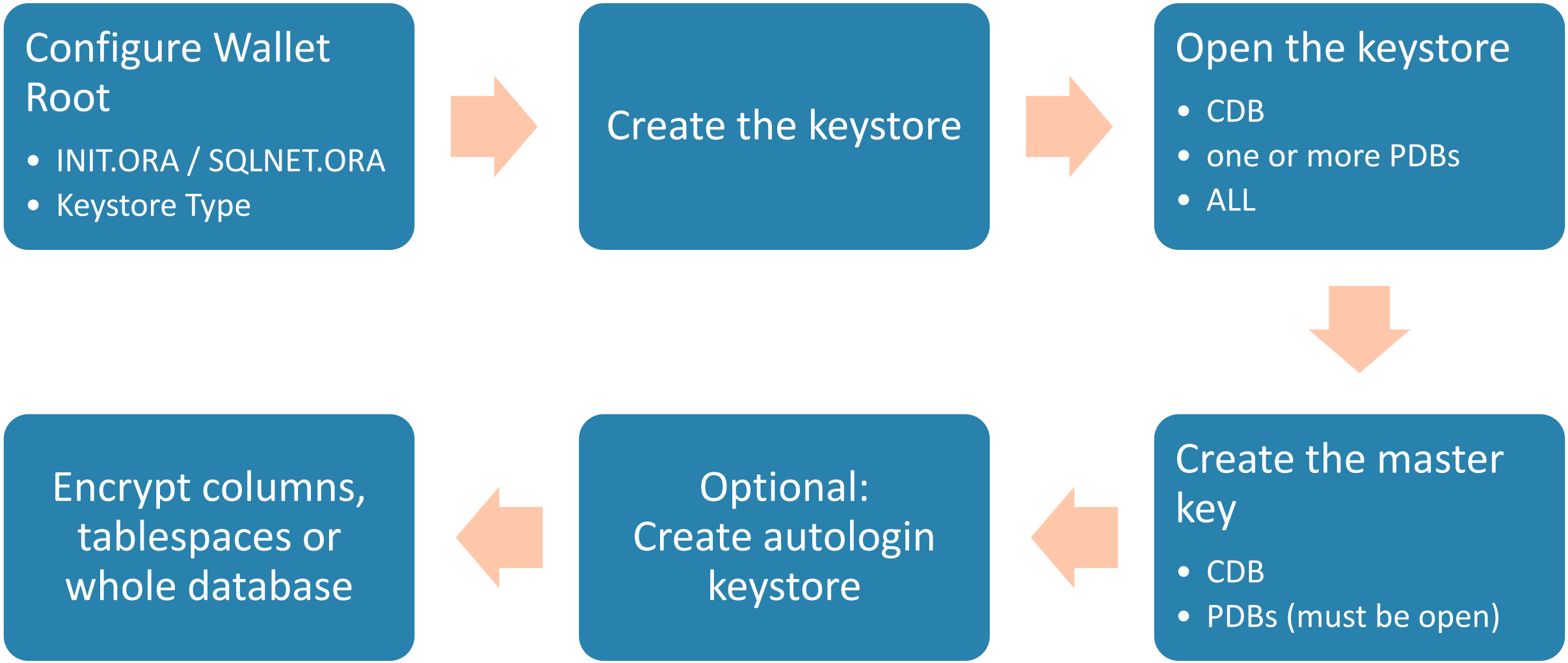


TDE configuration (On Premises)
Parameters

Parameter which affect TDE configuration

- WALLET_ROOT sets the root directory for Keystore files
  - Is a root directory, actual files are stored in <WALLET_ROOT>/tde
  - Isolated keystores are stored in subdirectory <WALLET_ROOT>/<PDB GUID>/tde
  - Wallet configuration in SQLNET.ORA therefore no longer needed
- TDE_CONFIGURATION set the keystore type (Hardware / Software)
  - Syntax TDE_CONFIGURATION = "{ KEYSTORE_CONFIGURATION = value [; CONTAINER = pdb-name] }"
  - Can be set only when WALLET_ROOT is defined
  - If you set the parameter per PDB, CDB$ROOT must be in mount mode
- ENCRYPT_NEW_TABLESPACES dynamic parameter
  - CLOUD_ONLY (default) Tablespaces are encrypted in Oracle Cloud if encryption clause is omitted
  - ALWAYS Tablespaces are always encrypted if encryption clause is omitted
  - DDL Tablespaces are only encrypted if the CREATE TABLESPACE command uses the encryption clause
- ONE_STEP_PLUGIN_FOR_PDB_WITH_TDE
  - Password handling of TDE keys in PDB after it has moved to the target CDB

TDE configuration (On Premises)
Configuration workflow



TDE configuration (On Premises)
Examples

Step 1: Configure the Wallet Root

```
{
  mkdir -p ${ORACLE_BASE}/admin/${ORACLE_SID}/wallet/tde

  sqlplus / as sysdba << EOF

  create pfile='${ORACLE_BASE}/admin/${ORACLE_SID}/pfile/${ORACLE_SID}-`date +%F`.ora' from spfile;

  alter system set WALLET_ROOT="${ORACLE_BASE}/admin/${ORACLE_SID}/wallet" scope=spfile;

  shutdown immediate

  startup

  alter system set TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=FILE";

EOF
}
```

TDE configuration (On Premises)
Examples

Wallet location

> The location set by the WALLET_ROOT location in SPFILE
> The location specified in the ENCRYPTION_WALLET_LOCATION setting in the sqlnet.ora file (deprecated since 18c)
> The location specified by the WALLET_LOCATION setting in the sqlnet.ora file
> The $ORACLE_BASE/admin/<db_unique_name>/wallet directory

TDE configuration (On Premises)
Examples

Step 2: Create the password protected key store

```
{
  sqlplus / as sysdba << EOF

  administer key management
  create keystore '${ORACLE_BASE}/admin/${ORACLE_SID}/wallet/tde' identified by welcome1;

  set lines 300
  column WRL_PARAMETER format a40

  select WRL_TYPE, WRL_PARAMETER, STATUS, CON_ID from v\$encryption_wallet;

EOF

 ll ${ORACLE_BASE}/admin/${ORACLE_SID}/wallet/tde
}

WRL_TYPE             WRL_PARAMETER                                STATUS                         CON_ID
-------------------- -------------------------------------------- ------------------------------ ----------
FILE                 /u01/app/oracle/admin/CN01/wallet/tde/       CLOSED                         1
FILE                                                              CLOSED                         2
FILE                                                              CLOSED                         3

-rw-------. 1 oracle oinstall 2555 May 17 13:59 ewallet.p12
```

TDE configuration (On Premises)
Examples

Step 3: Open the key store CDB & PDB

```
{
  sqlplus / as sysdba << EOF

  alter pluggable database all open;

  administer key management set keystore open force keystore identified by welcome1 container = all;

  set lines 300
  column WRL_PARAMETER format a40

  select WRL_TYPE, WRL_PARAMETER, STATUS, CON_ID from v\$encryption_wallet;
EOF

}

WRL_TYPE             WRL_PARAMETER                                STATUS                         CON_ID
-------------------- -------------------------------------------- ------------------------------ ----------
FILE                 /u01/app/oracle/admin/CN01/wallet/tde/   OPEN_NO_MASTER_KEY               1
FILE                                                          OPEN_NO_MASTER_KEY               2
FILE                                                          OPEN_NO_MASTER_KEY               3
```

TDE configuration (On Premises)
Examples

Step 4.1: Create the master key for the container database

```
{
  sqlplus / as sysdba << EOF

  administer key management set key identified by welcome1 with backup;

  set lines 300
  column WRL_PARAMETER format a40
  column NAME format a10

  select WRL_TYPE, WRL_PARAMETER, STATUS, NAME
  from v\$encryption_wallet a, v\$pdbs b
  where a.con_id = b.con_id (+);

EOF
}

WRL_TYPE             WRL_PARAMETER                            STATUS                         NAME
-------------------- ---------------------------------------- ------------------------------ ----------
FILE                                                          OPEN                           PDB$SEED
FILE                                                          OPEN_NO_MASTER_KEY             PDB01
FILE                 /u01/app/oracle/admin/CN01/wallet/tde/   OPEN
```

TDE configuration (On Premises)
Examples

Step 4.2: Create the master key for PDB (unified mode)

```
{
  sqlplus / as sysdba << EOF

  alter session set container = PDB01;
  administer key management set key identified by welcome1 with backup;

  set lines 300
  column WRL_PARAMETER format a40
  column NAME format a10

  select WRL_TYPE, WRL_PARAMETER, STATUS, NAME
  from v\$encryption_wallet a, v\$pdbs b
  where a.con_id = b.con_id (+);

EOF
}

WRL_TYPE             WRL_PARAMETER                            STATUS                         NAME
-------------------- ---------------------------------------- ------------------------------ ----------
FILE                                                          OPEN                           PDB01
```

TDE configuration (On Premises)
How to configure

Step 5: Create an autologin keystore for the CDB

```
{
  sqlplus / as sysdba << EOF

  administer key management create auto_login keystore
  from keystore '${ORACLE_BASE}/admin/${ORACLE_SID}/wallet/tde'
  identified by welcome1;
EOF

 ll ${ORACLE_BASE}/admin/${ORACLE_SID}/wallet/tde
}

-rw-------. 1 oracle oinstall 4040 May 19 16:24 cwallet.sso
-rw-------. 1 oracle oinstall 2555 May 19 16:21 ewallet_2019051914213006.p12
-rw-------. 1 oracle oinstall 3995 May 19 16:21 ewallet.p12
-rw-------. 1 oracle oinstall    0 May 19 14:32 ewallet.p12.lck
```

TDE configuration (On Premises)
How to configure

Step 6.1: Encrypt tablespaces online (CDB)

```
{
  sqlplus / as sysdba << EOF

  alter tablespace users encryption online encrypt;

  select TABLESPACE_NAME, STATUS, ENCRYPTED from DBA_TABLESPACES;
EOF

}

TABLESPACE_NAME                STATUS    ENC
------------------------------ --------- ---
SYSTEM                         ONLINE    NO
SYSAUX                         ONLINE    NO
UNDOTBS1                       ONLINE    NO
TEMP                           ONLINE    NO
USERS                          ONLINE    YES
```

TDE configuration (On Premises)
How to configure

Step 6.2: Encrypt tablespaces online (PDB)

```
{
  sqlplus / as sysdba << EOF
  alter session set container = PDB01;
  select TABLESPACE_NAME, STATUS, ENCRYPTED from DBA_TABLESPACES;
  alter tablespace users encryption online encrypt;
  select TABLESPACE_NAME, STATUS, ENCRYPTED from DBA_TABLESPACES;
EOF
}

TABLESPACE_NAME                STATUS      ENC
------------------------------ --------- ---
SYSTEM                         ONLINE      NO
SYSAUX                         ONLINE      NO
UNDOTBS1                       ONLINE      NO
TEMP                           ONLINE      NO
USERS                          ONLINE      NO

TABLESPACE_NAME                STATUS      ENC
------------------------------ --------- ---
SYSTEM                         ONLINE      NO
SYSAUX                         ONLINE      NO
UNDOTBS1                       ONLINE      NO
TEMP                           ONLINE      NO
USERS                          ONLINE      YES
```

TDE configuration (On Premises)
How to configure

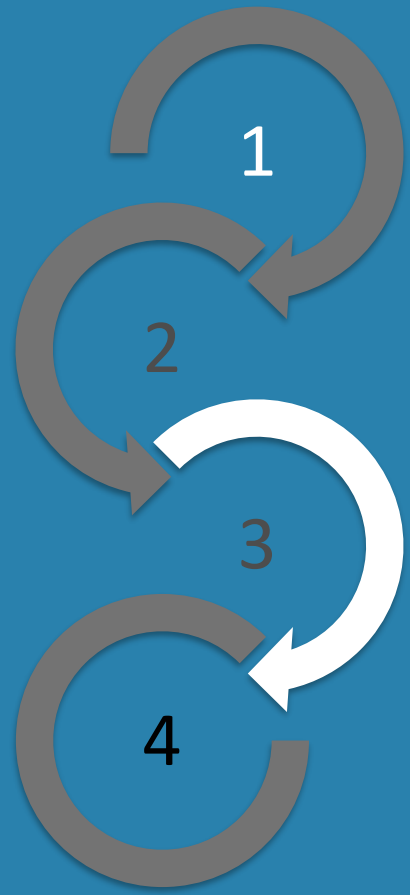
Let’s try to create an isolated keystore

```
{
  mkdir -p ${ORACLE_BASE}/admin/${ORACLE_SID}/wallet/PDB01/tde

  sqlplus / as sysdba << EOF

  alter session set container = PDB01;

  alter system set TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=FILE";

EOF
}

ERROR at line 1:
ORA-32017: failure in updating SPFILE
ORA-12754: Feature 'Per-PDB TDE keystore' is disabled due to missing capability 'Runtime Environment'.
```

MOS Note ID 2489529.1:
This feature is only available to Cloud environment. This is not available for On-Premises Database.

TDE configuration (OCI)

- Differences between On-Premises and OCI
- Encryption in OCI
- More on backup and restore of wallets
- PDB and TDE
- Isolated Keystores



TDE configuration (OCI)
Differences between On-Premises and OCI

Wallet root configuration

> Is done through $ORACLE_HOME/network/admin/sqlnet.ora (classical way)

> Directory is =/opt/oracle/dcs/commonstore/wallets/tde/$ORACLE_UNQNAME

> Keystores are in united mode by default

```
oracle@tre4doag-190604:/home/oracle/ [CLD01] cat $ORACLE_HOME/network/admin/sqlnet.ora

ENCRYPTION_WALLET_LOCATION=
 (SOURCE=
   (METHOD=FILE)
   (METHOD_DATA=
     (DIRECTORY=/opt/oracle/dcs/commonstore/wallets/tde/$ORACLE_UNQNAME))
 )

SQLNET.ENCRYPTION_SERVER=REQUIRED
SQLNET.CRYPTO_CHECKSUM_SERVER=REQUIRED
SQLNET.ENCRYPTION_TYPES_SERVER=(AES256,AES192,AES128)
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER=(SHA1)
SQLNET.ENCRYPTION_CLIENT=REQUIRED
SQLNET.CRYPTO_CHECKSUM_CLIENT=REQUIRED
SQLNET.ENCRYPTION_TYPES_CLIENT=(AES256,AES192,AES128)
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT=(SHA1)
```

TDE configuration (OCI)
Differences between On-Premises and OCI

Wallet Backup

> Is done through DCS based backups (dbcli)
> Backup goes to the object store
> System assumes wallets in /opt/oracle/dcs/commonstore/wallets/tde/$ORACLE_UNQNAME
> Changes in configuration not allowed
> Isolated Keystores not possible (sqlnet.ora vs. WALLET_ROOT parameter)

```
[root@tre4doag-190604 CLD01_fra1wm]# dbcli list-tdebackupreports

DbResID                                  OraDbId              recoveryTag          BackupLocation
---------------------------------------- -------------------- -------------------- ----------------------------------
--
c7d5ad8e-1085-4511-af6f-7fdd335ac03c     3759185031                                https://swiftobjectstorage.eu-
frankfurt-
1.oraclecloud.com/v1/dbbackupfra/bdZ6freKonYbLnnqQZEm/dbSys453niyma/database/3759185031/CLD01_fra1wm/tdewallet/TDEWallet.
tar.gz
```

TDE configuration (OCI)
Encryption in OCI

What is encrypted by default

> All new tablespaces
> SYSTEM, SYSAUX, UNDO and TEMP are not encrypted by default
> You can encrypt the whole database, if needed
> RMAN Backups are encrypted

```
[RMAN configuration parameters for database with db_unique_name CLD01_FRA1WM are:
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 30 DAYS;
CONFIGURE BACKUP OPTIMIZATION OFF;
…
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE 2 G;
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' MAXPIECESIZE 2 G FORMAT   '%d_%I_%U_%T_%t' PARMS
'SBT_LIBRARY=/opt/oracle/dcs/commonstore/pkgrepos/oss/odbcs/libopc.so
ENV=(OPC_PFILE=/opt/oracle/dcs/commonstore/objectstore/opc_pfile/3759185031/opc_CLD01_fra1wm.ora)';
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE ON;
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
```

TDE configuration (OCI)
Encryption in OCI

What is encrypted by default

- DATAPUMP doesn't encrypt by default
- Several parameters for encryption (TDE must be licensed, compatible at least 11.0.0, EE Option)
  - ENCRYPTION = [ALL | DATA_ONLY | ENCRYPTED_COLUMNS_ONLY | METADATA_ONLY | NONE]
  - ENCRYPTION_ALGORITHM = [AES128 | AES192 | AES256]
  - ENCRYPTION_MODE = [DUAL | PASSWORD | TRANSPARENT]
  - ENCRYPTION_PASSWORD = *password*

```
expdp userid=system/***** directory=FOO_DIR tables=system.foo dumpfile=foo.dmp
...
Processing object type TABLE_EXPORT/TABLE/TABLE
. . exported "SYSTEM"."FOO"                              9.692 MB   73207 rows
ORA-39173: Encrypted data has been stored unencrypted in dump file set.

expdp userid=system/SagCloud01_# directory=FOO_DIR tables=system.foo dumpfile=foo.dmp encryption=all
...
Processing object type TABLE_EXPORT/TABLE/TABLE
. . exported "SYSTEM"."FOO"                              9.692 MB   73207 rows
Master table "SYSTEM"."SYS_EXPORT_TABLE_01" successfully loaded/unloaded
```

TDE configuration (OCI)
Encryption in OCI

And Standard Edition in OCI?

> SYSTEM, SYSAUX, TEMP and UNDO unencrypted
> USERS and additional tablespaces are encrypted
> RMAN backups are **not** encrypted
> DATAPUMP exports are not encrypted by default
> Usage of ENCRYPTION parameter not possible due to SE restrictions

```
[oracle@tre4doag190604-3 ~]$ expdp userid=system/SagCloud01_# directory=FOO_DIR tables=system.foo
dumpfile=foo.dmp encryption=all
...
Connected to: Oracle Database 18c Standard Edition 2 Release 18.0.0.0.0 - Production
ORA-39002: invalid operation
ORA-00439: feature not enabled: Dump File Encryption
```

```
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' MAXPIECESIZE 2 G FORMAT   '%d_%I_%U_%T_%t' PARMS
'SBT_LIBRARY=/opt/oracle/dcs/commonstore/pkgrepos/oss/odbcs/libopc.so
ENV=(OPC_PFILE=/opt/oracle/dcs/commonstore/objectstore/opc_pfile/4191515462/opc_CLD03_fra15v.ora)';
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF;
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
```

TDE configuration (OCI)
More on backup and restore of wallets

Backup of wallet

> Is done automatically if OCI backups are used
> Backup created with `dbcli create-backup` or via GUI (automated or manual)
> If own backup solution or different locations is used, it must be done manual
> Backups go to Object Storage
> Logs can be found in /opt/oracle/dcs/log/

```
[root@tre4doag-190604 dcs]# dbcli list-tdebackupreports

DbResID                                  OraDbId               recoveryTag          BackupLocation
---------------------------------------- --------------------- -------------------- ----------------------
------------------
c7d5ad8e-1085-4511-af6f-7fdd335ac03c     3759185031
https://swiftobjectstorage.eu-frankfurt-
1.oraclecloud.com/v1/dbbackupfra/bdZ6freKonYbLnnqQZEm/dbSys453niyma/database/3759185031/CLD01_fra1wm/tdew
allet/TDEWallet.tar.gz
```

TDE configuration (OCI)
More on backup and restore of wallets

Restore of wallet

> Use `dbcli recover-tdewallet` to recover a lost wallet

```
[oracle@tre4doag-190604 ~]$ srvctl stop database -db CLD01_fra1wm
[oracle@tre4doag-190604 ~]$ cd /opt/oracle/dcs/commonstore/wallets/tde/CLD01_fra1w
[oracle@tre4doag-190604 CLD01_fra1wm]$ mv cwallet.sso cwallet.sso.foo
[oracle@tre4doag-190604 CLD01_fra1wm]$ mv ewallet.p12 ewallet.p12.foo
[oracle@tre4doag-190604 CLD01_fra1wm]$ srvctl start database -db CLD01_fra1wm

SQL> select count(*) from system.foo;
select count(*) from system.foo
                       *
ERROR at line 1:
ORA-28365: wallet is not open

SQL> administer key management set keystore open force keystore identified by SagCloud01_#;
administer key management set keystore open force keystore identified by SagCloud01_#
*
ERROR at line 1:
ORA-28367: wallet does not exist

[root@tre4doag-190604 dcs]# dbcli recover-tdewallet -in CLD01 -tp

SQL> select count(*) from system.foo;

  COUNT(*)
----------
     73207
```

Unplug/plug an encrypted PDB in united mode

> Need to export the wallet keys

```
SQL> alter pluggable database pdb01 unplug into '/home/oracle/pdb01.xml';
alter pluggable database pdb01 unplug into '/home/oracle/pdb01.xml'
*
ERROR at line 1:
ORA-46680: master keys of the container database must be exported

SQL> alter pluggable database pdb01 unplug into '/home/oracle/pdb01.xml' encrypt using welcome1;

Pluggable database altered.

SQL> drop pluggable database pdb01;

Pluggable database dropped.
SQL> create pluggable database pdb01 using '/home/oracle/pdb01.xml' nocopy keystore identified by
"welcome1" decrypt using "welcome1";

Pluggable database created.

SQL> alter pluggable database pdb01 open;

Pluggable database altered.
```

TDE configuration (OCI)
Isolated keystore

Prepare your OCI CDB

> Create a wallet directory (/u01/app/oracle/admin/<DB_UNQ_NAME>/wallet/tde)
> Copy/move your keystore to wallet directory
> Set WALLET_ROOT and TDE_CONFIGURATION
> Restart the container database

```
show parameter wallet

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
wallet_root                          string      /u01/app/oracle/admin/CLD01/wa
                                                 llet
SQL> show parameter tde

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
one_step_plugin_for_pdb_with_tde     boolean     FALSE
tde_configuration                    string      KEYSTORE_CONFIGURATION=FILE
```

TDE configuration (OCI)
Isolated keystore

Prepare your PDB

> Create/Open your PDB and set TDE_CONFIGURATION parameter
> Create a keystore
> Open the keystore
> Set a master key for the PDB
> Optional: create autologin keystore

```
SQL> create pluggable database pdb01 admin user pdbadmin identified by welcome1;

SQL> alter system set tde_configuration = "KEYSTORE_CONFIGURATION=FILE" scope=spfile;

SQL> alter database open;

SQL> administer key management create keystore identified by welcome2;

SQL> administer key management set keystore open identified by welcome2;

SQL> administer key management set key identified by welcome2 with backup;

SQL> administer key management create auto_login keystore from keystore identified by welcome2;
```

TDE configuration (OCI)
Isolated keystore

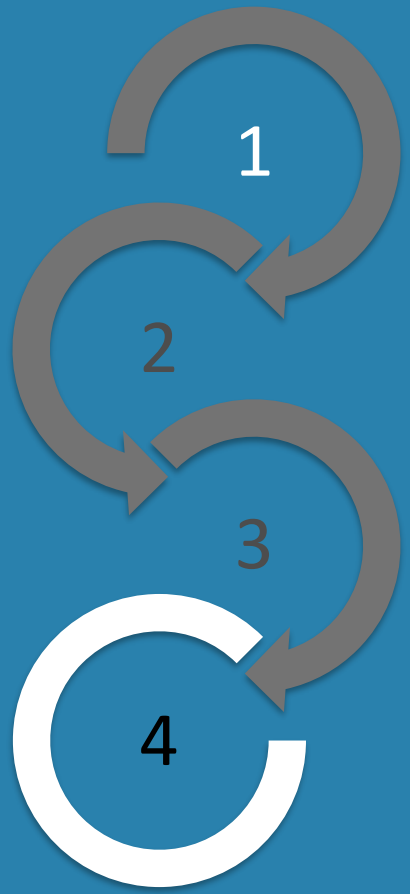
Unplug your PDB

> Easy unplug of you PDB
> No ORA-46680: master keys of the container database must be exported
> PDB is now “self contained”
> Mix of isolated / united keystore possible

```
SQL> alter pluggable database pdb01 close immediate;

Pluggable database altered.

SQL> alter pluggable database pdb01 unplug into '/home/oracle/pdb01.xml';

Pluggable database altered.
```

Conclusion

- Benefits and advantages of TDE
- Problems & Drawbacks



Conclusion
Benefits and advantages of TDE

Security
> Data is safe (some tools don’t encrypt by default)
> Whole database encryption hides also SYSTEM, SYSAUX, TEMP and UNDO data
> You don’t need OMF anymore if you use tablespace online encryption
> Keystore can be closed even SYSTEM, SYAUX and UNDO is encrypted

Handling
> Isolated keystores makes unplug/plug much easier
> New parameters TDE_CONFIGURATION and WALLET_ROOT

OCI
> Without any license costs included
> Automated backup of wallets

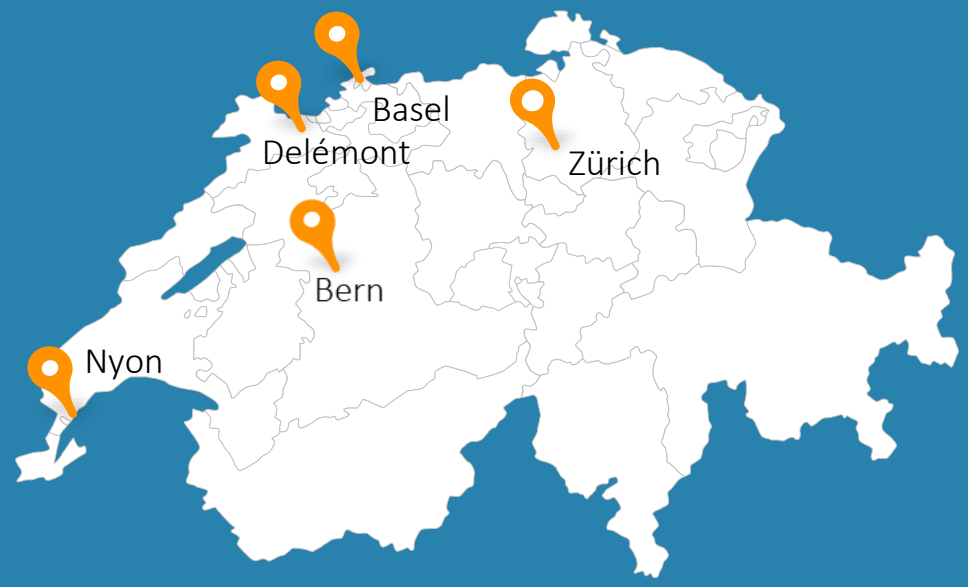
Conclusion
Problems & Drawbacks

OCI
> Not all components available for SE (RMAN backups, data pump)
> Isolated keystores not compatible to standard configuration (SQLNET.ORA)

On-Premises
> Not available for SE (SQL*Net encryption is available, that's good)
> Additional costs through EE Advanced Security Option
> Isolated keystores only for Cloud, ODA and Exadata Versions

Any questions?

Please do ask!



We would love to boost your IT-Infrastructure

How about you?