

Oracle TDE: How to Use It and Survive

Marcin Przepiorowski | Technical Manager | December, 2019

About me

Oracle consultant/DBA since 2000

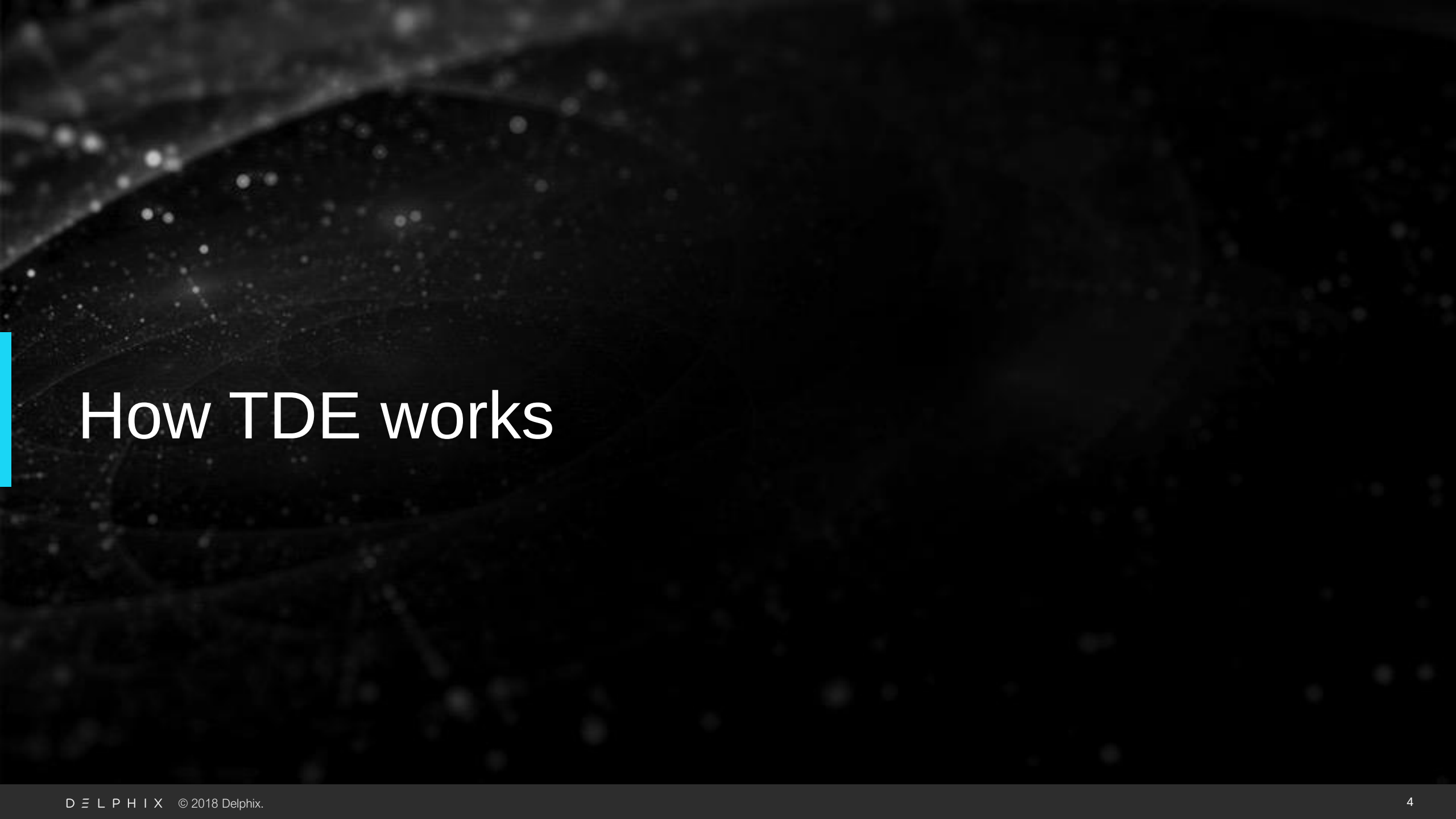
co-developer of OraSASH – free ASH/AWR like repository

Automation / DevOps freak

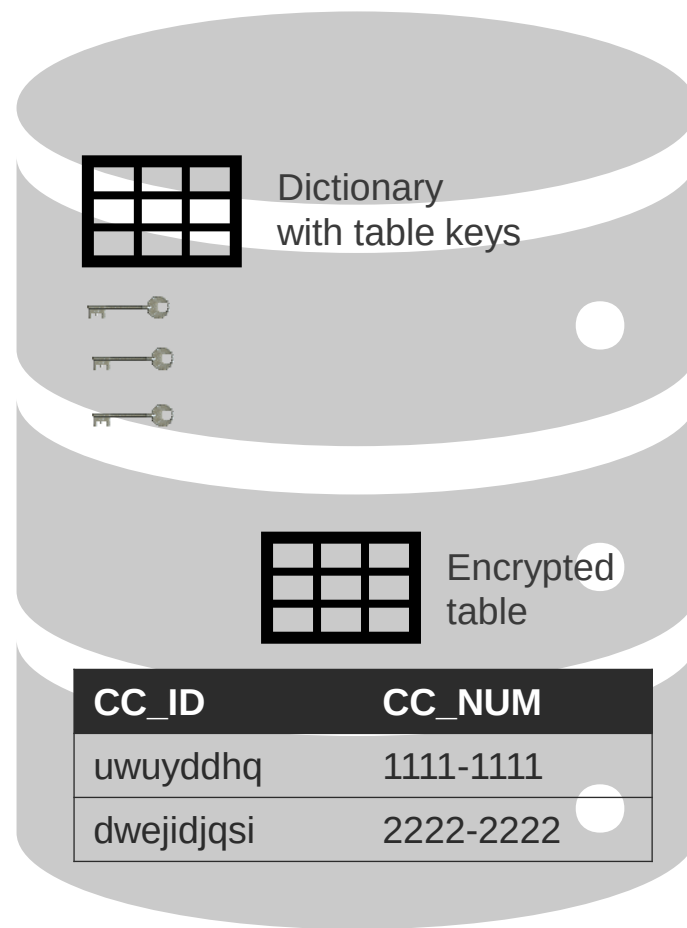
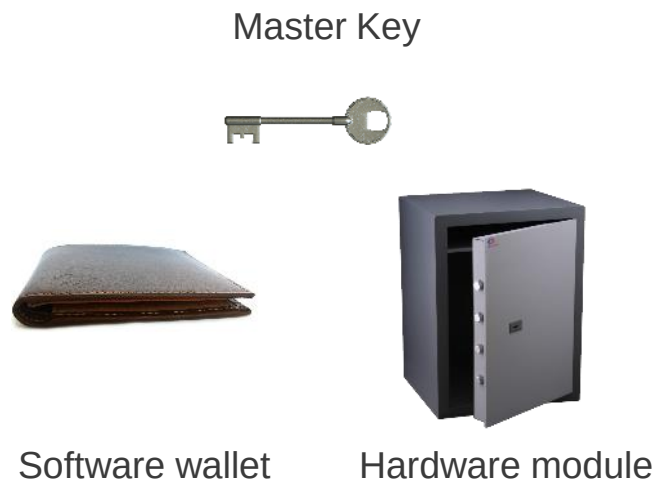


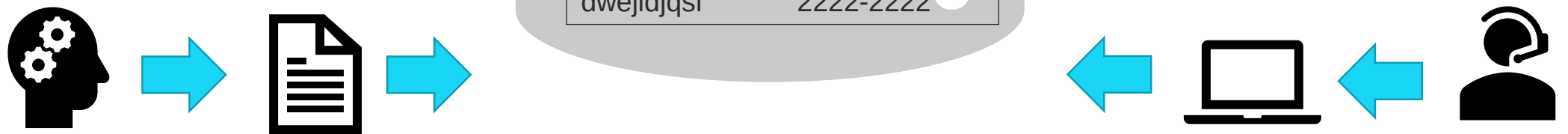
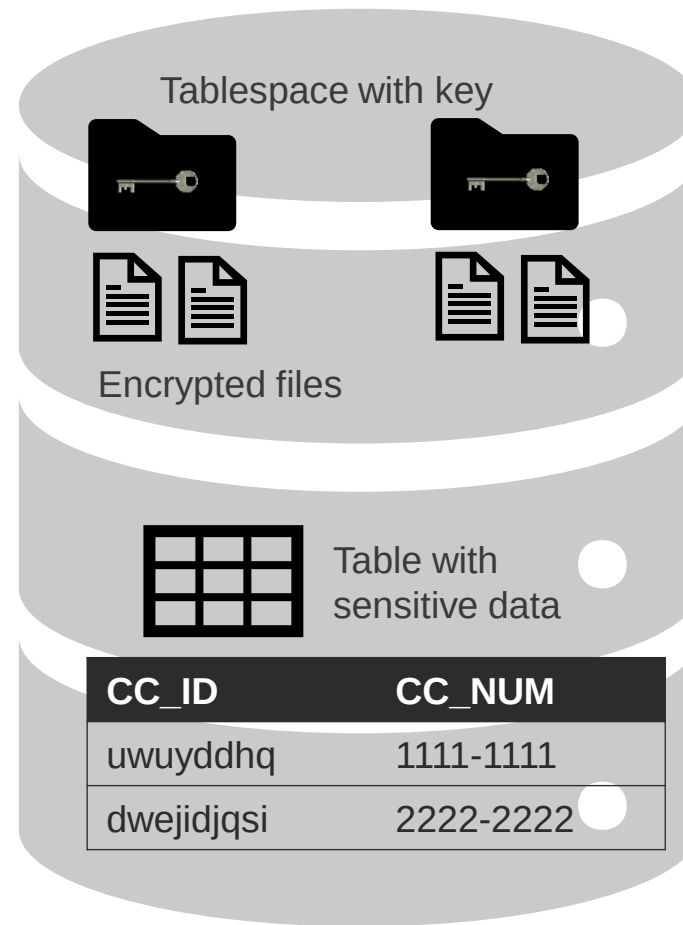
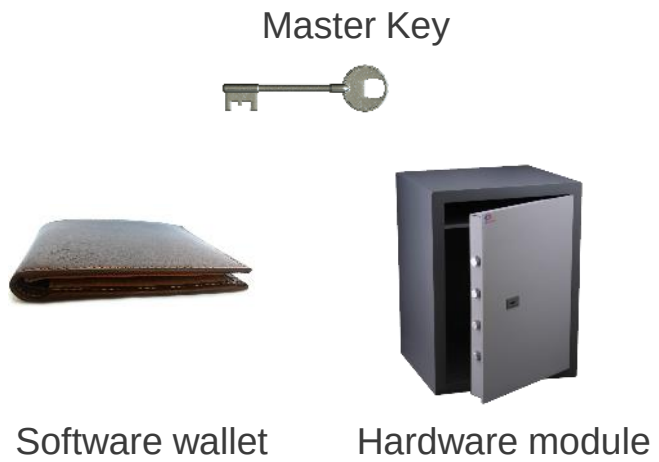
Delphix Dynamic Data Platform





How TDE works





Checking encryption

```
create table piro.test_enc_column (id number, cc varchar2(50) encrypt) tablespace users;
Table created.
insert into piro.test_enc_column values (1, 'marcin');
1 row created.
commit;
Commit complete.
```

```
create table piro.test_noenc (id number, cc varchar2(50) ) tablespace users;
Table created.
insert into piro.test_noenc values (1, 'przepiorowski');
1 row created.
commit;
Commit complete.
shutdown immediate
```

```
[oracle@oracle18 ~]$ grep -i przepiorowski /u01/app/oracle/oradata/TEST18/users01.dbf
Binary file /u01/app/oracle/oradata/TEST18/users01.dbf matches
```

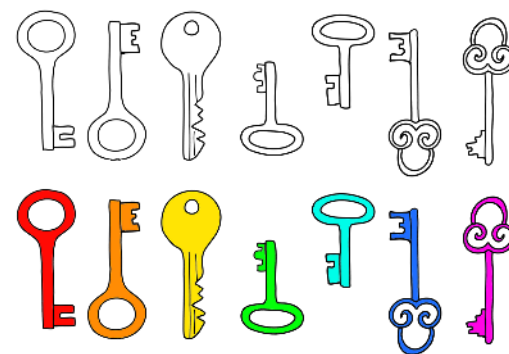
```
[oracle@oracle18 ~]$ grep -i marcin /u01/app/oracle/oradata/TEST18/users01.dbf
[oracle@oracle18 ~]$
```

Table vs Tablespace encryption

Table	Tablespace
No range scan on indexes	Unique and range scans on indexes
Space overhead (1 to 52 bytes per row)	No space overhead
Potential impact on execution plan 728292.1.	No impact on execution plan
Always encrypted	Decrypted in SGA
Key per table	Key per tablespace

TDE Master Note - MOS1228046.1

TDE 12c FAQ - MOS 2253348.1



Wallets

Multitenant keystores (wallet) types:

- United – TDE master encryption key for CDB and all PDBs are in same keystore
- Isolated – TDE master encryption key for CDB and PDB's are in individual PDB's keystores

```
SQL> select con_id, wrl_parameter, status, wallet_type, keystore_mode from v$encryption_wallet;
```

CON_ID	WRL_PARAMETER	STATUS	WALLET_TYPE	KEYSTORE
1	/u01/app/oracle/admin/wallets/test18mt/	OPEN	PASSWORD	NONE
2		CLOSED	UNKNOWN	UNITED
3		OPEN	PASSWORD	UNITED

Wallets

Locations:

- WALLET_ROOT (init.ora but also TDE_CONFIGURATION value)
- WALLET_LOCATION (sqlnet.ora)
- ENCRYPTION_WALLET_LOCATION (sqlnet.ora)
- \$ORACLE_BASE/admin/db_unique_name/wallet
- \$ORACLE_HOME/admin/db_unique_name/wallet

TNS_ADMIN can do a trick as well 😊

```
ls -l /u01/app/oracle/admin/wallets/test18mt/  
-rw-----. 1 oracle oinstall 5819 Oct 30 14:35 ewallet.p12
```

Wallets

The use of PKI (orapki) encryption with Transparent Data Encryption is deprecated.
Use the ADMINISTER KEY MANAGEMENT SQL statement.

```
[oracle@oracle18 test18mt]$ orapki wallet display -wallet ewallet.p12 -summary
Oracle PKI Tool Release 18.0.0.0.0 - Production
Version 18.1.0.0.0
Copyright (c) 2004, 2017, Oracle and/or its affiliates. All rights reserved.
```

- Enter wallet password:
- null

Check if TDE was ever enabled

You can't recreate a wallet if database was encrypted before

Check table `x$kcbdbk`

Clean database (no encryption enabled)

11g

```
BITAND(FLAGS,8) == 0
```

12c+

```
mkloc == 0
```



Non Multitenant setup

Non Multitenant setup

```
mkdir -p /u01/app/oracle/admin/wallets/test18
```

```
sqlnet.ora
```

```
WALLET_LOCATION=
```

```
(SOURCE=
```

```
(METHOD=FILE)
```

```
(METHOD_DATA=
```

```
(DIRECTORY=/u01/app/oracle/admin/wallets/${ORACLE_SID}/)))
```

Non Multitenant setup

```
select wr1_parameter, status, wallet_type from v$encryption_wallet;
```

WRL_PARAMETER	STATUS	WALLET_TYPE
/u01/app/oracle/admin/wallets/test18/	NOT_AVAILABLE	UNKNOWN

```
administer key management create keystore '/u01/app/oracle/admin/wallets/test18/' identified by xxx;  
keystore altered.
```

```
select wr1_parameter, status, wallet_type from v$encryption_wallet;
```

WRL_PARAMETER	STATUS	WALLET_TYPE
/u01/app/oracle/admin/wallets/test18/	CLOSED	UNKNOWN

Non Multitenant setup

```
administer key management set keystore open identified by xxx;  
keystore altered.
```

```
select wrl_parameter, status, wallet_type from v$encryption_wallet;
```

WRL_PARAMETER	STATUS	WALLET_TYPE
/u01/app/oracle/admin/wallets/test18/	OPEN_NO_MASTER_KEY	PASSWORD

Non Multitenant setup

```
select mkloc from x$kcdbbk;
```

```
MKLOC
```

```
-----
```

```
0
```

```
administer key management set key using tag "first master key" identified by xxx with backup using  
'backup before master';
```

```
keystore altered.
```

```
select wrl_parameter, status, wallet_type from v$encryption_wallet;
```

```
WRL_PARAMETER
```

```
STATUS
```

```
WALLET_TYPE
```

```
-----
```

```
/u01/app/oracle/admin/wallets/test18/
```

```
OPEN
```

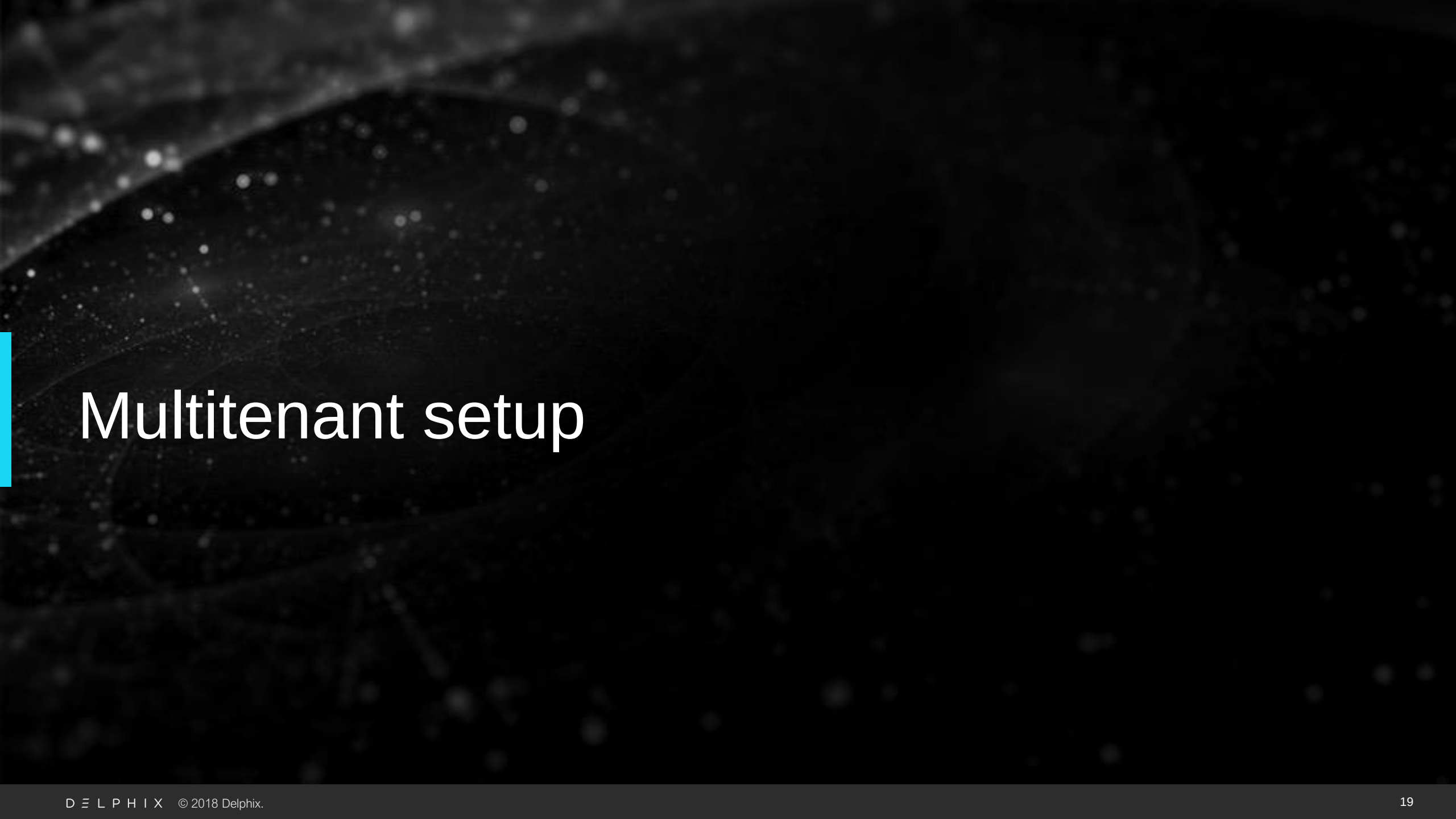
```
PASSWORD
```

```
select mkloc from x$kcdbbk;
```

```
MKLOC
```

```
-----
```

```
1
```



Multitenant setup

Multitenant setup

```
mkdir -p /u01/app/oracle/admin/wallets/test18mt
```

```
sqlnet.ora
```

```
WALLET_LOCATION=
```

```
(SOURCE=
```

```
(METHOD=FILE)
```

```
(METHOD_DATA=
```

```
(DIRECTORY=/u01/app/oracle/admin/wallets/${ORACLE_SID}/)))
```

Multitenant setup

```
SQL> select con_id, wrl_parameter, status, wallet_type from v$encryption_wallet
```

CON_ID	WRL_PARAMETER	STATUS	WALLET_TYPE
1	/u01/app/oracle/admin/wallets/test18mt/	NOT_AVAILABLE	UNKNOWN
2		NOT_AVAILABLE	UNKNOWN
3		NOT_AVAILABLE	UNKNOWN

```
SQL> select con_id, mkloc from x$kcdbdk
```

CON_ID	MKLOC
1	0
2	0
3	0

Multitenant setup

```
administer key management create keystore '/u01/app/oracle/admin/wallets/test18mt/' identified by xxx;
```

```
select con_id, wrl_parameter, status, wallet_type from v$encryption_wallet;
```

CON_ID	WRL_PARAMETER	STATUS	WALLET_TYPE
1	/u01/app/oracle/admin/wallets/test18mt/	CLOSED	UNKNOWN
2		CLOSED	UNKNOWN
3		CLOSED	UNKNOWN

```
administer key management set keystore open identified by xxx;
```

```
select con_id, wrl_parameter, status, wallet_type from v$encryption_wallet;
```

CON_ID	WRL_PARAMETER	STATUS	WALLET_TYPE
1	/u01/app/oracle/admin/wallets/test18mt/	OPEN_NO_MASTER_KEY	PASSWORD
2		CLOSED	UNKNOWN
3		CLOSED	UNKNOWN

Multitenant setup

```
administer key management set key using tag "first master key cdb" identified by delphix with backup
using 'backup before master cdb';
```

```
select con_id, wrl_parameter, status, wallet_type from v$encryption_wallet;
```

CON_ID	WRL_PARAMETER	STATUS	WALLET_TYPE
1	/u01/app/oracle/admin/wallets/test18mt/	OPEN	PASSWORD
2		CLOSED	UNKNOWN
3		CLOSED	UNKNOWN

```
select con_id, mkloc from x$kcdbbk;
```

CON_ID	MKLOC
1	1
2	0
3	0

Multitenant setup

```
administer key management set keystore open identified by delphix container = all;
```

```
select con_id, wrl_parameter, status, wallet_type from v$encryption_wallet;
```

CON_ID	WRL_PARAMETER	STATUS	WALLET_TYPE
1	/u01/app/oracle/admin/wallets/test18mt/	OPEN	PASSWORD
2		CLOSED	UNKNOWN
3		OPEN_NO_MASTER_KEY	PASSWORD

```
administer key management set key using tag "first master key all" identified by delphix with backup  
using 'backup before master all' container=all;
```

```
select con_id, wrl_parameter, status, wallet_type from v$encryption_wallet;
```

CON_ID	WRL_PARAMETER	STATUS	WALLET_TYPE
1	/u01/app/oracle/admin/wallets/test18mt/	OPEN	PASSWORD
2		CLOSED	UNKNOWN
3		OPEN	PASSWORD

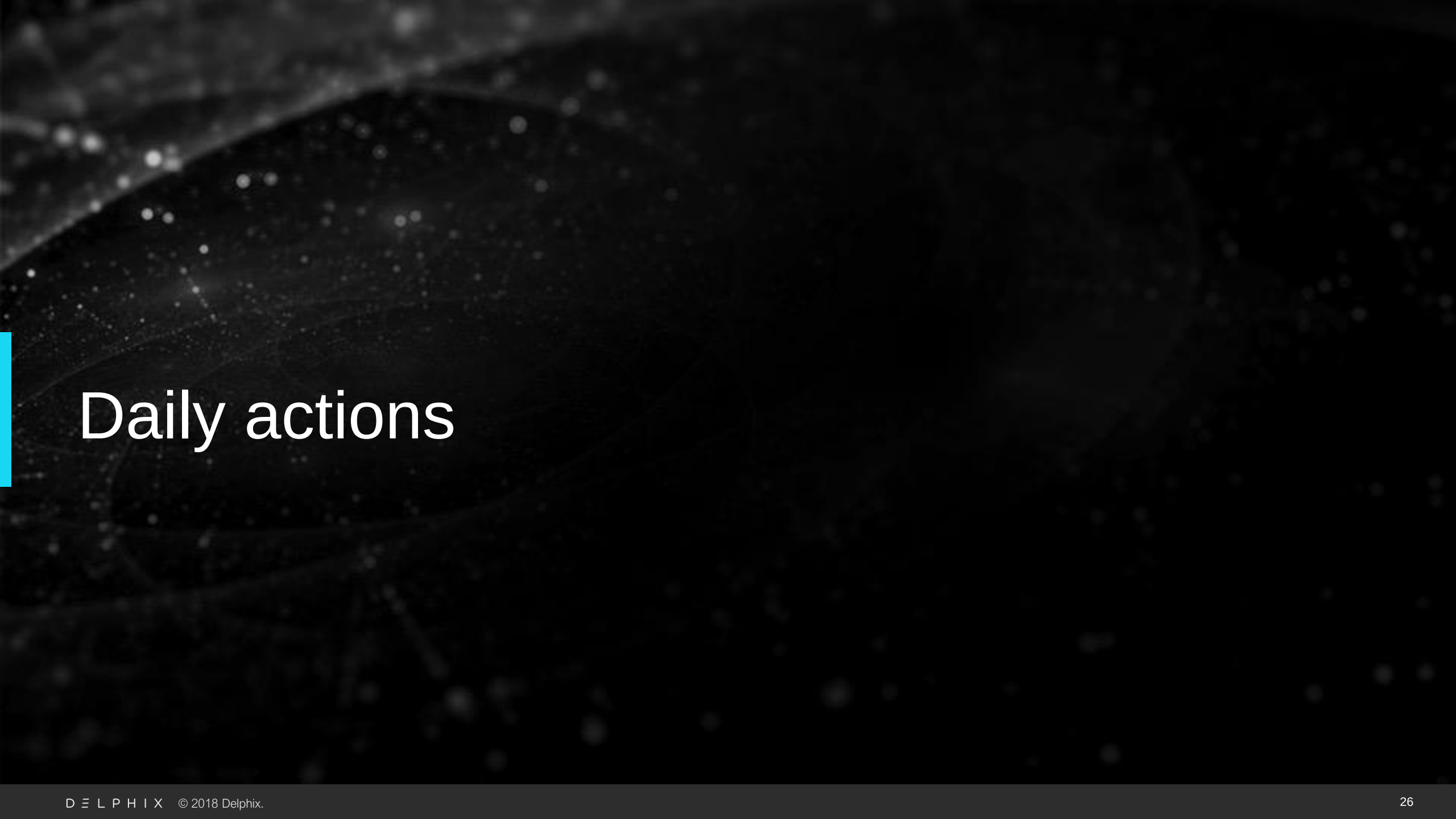
Multitenant setup

```
alter session set container = pdb1;
```

```
select con_id, wrl_parameter, status, wallet_type from v$encryption_wallet;
```

CON_ID	WRL_PARAMETER	STATUS	WALLET_TYPE
3		OPEN	PASSWORD

```
create tablespace ts_pii datafile '/u01/app/oracle/oradata/TEST18MT/PDB1/ts_pii01.dbf' size 10M
encryption using 'AES256' default storage(encrypt);
```



Daily actions

Encrypting system objects

```
SQL> create table test_enc_column (id number, cc varchar2(50) encrypt) tablespace users;  
create table test_enc_column (id number, cc varchar2(50) encrypt) tablespace users  
                                *  
ERROR at line 1:  
ORA-28336: cannot encrypt SYS owned objects
```

You can encrypt SYSTEM, SYSAUX, TEMP and UNDO tablespace but you can't close wallet manually anymore.
MOS 2393734.1

Tablespace level encryption

```
create tablespace ts_enc datafile '/u01/app/oracle/oradata/TEST18/ts_enc01.dbf' size 100M encryption  
using 'AES256' encrypt;
```

ALTER SYSTEM SET ENCRYPT_NEW_TABLESPACES = value;

- CLOUD_ONLY transparently encrypts the tablespace in the Cloud using the AES128 algorithm CLOUD_ONLY is the default.
- ALWAYS automatically encrypts the tablespace using the AES128 algorithm if you omit the ENCRYPTION clause of CREATE TABLESPACE
- DDL encrypts the tablespace using the specified setting of the ENCRYPTION

Tablespace level encryption

```
alter system set encrypt_new_tablespaces = always;  
System altered.
```

```
create tablespace ts_enc2 datafile '/u01/app/oracle/oradata/TEST18/ts_enc2_01.dbf' size 10M;  
Tablespace created.
```

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

TABLESPACE_NAME	ENC
SYSTEM	NO
SYSAUX	NO
UNDOTBS1	NO
TEMP	NO
USERS	NO
TS_ENC	YES
TS_ENC2	YES

Wallet backup

- Backup your wallet
- Backup your wallet always and probably include with database backup
- Backing up using administer command is not enough – you need to backup a physical file to other location
- Did I say to backup your wallet ??????

REKEY master key

```
alter system flush buffer_cache; alter system checkpoint;
```

```
[oracle@oracle18 ~]$ dd if=/u01/app/oracle/oradata/TEST18MT/PDB1/ts_pii01.dbf of=block.dmp_enc bs=8k  
count=1 skip=131
```

```
administer key management set key using tag "rekey pdb" identified by delphix with backup using  
"backup";
```

```
keystore altered.
```

```
alter system flush buffer_cache; alter system checkpoint;
```

```
[oracle@oracle18 ~]$ dd if=/u01/app/oracle/oradata/TEST18MT/PDB1/ts_pii01.dbf of=block.dmp_enc_rekey  
bs=8k count=1 skip=131
```

```
[oracle@oracle18 ~]$ md5sum block.dmp_enc
```

```
ab35da0dbe93a006424d94566c906db2 block.dmp_enc
```

```
[oracle@oracle18 ~]$ md5sum block.dmp_enc_rekey
```

```
ab35da0dbe93a006424d94566c906db2 block.dmp_enc_rekey
```

REKEY tablespace online

```
alter session set container = pdb1;  
Session altered.
```

```
select key_version, status from V$ENCRYPTED_TABLESPACES;  
KEY_VERSION STATUS  
-----  
0 NORMAL
```

```
alter tablespace ts_pii encryption using 'AES192' rekey file_name_convert = ('ts_pii01.dbf',  
'ts_pii01_new.dbf');
```

Tablespace altered.

```
select key_version, status from V$ENCRYPTED_TABLESPACES;  
KEY_VERSION STATUS  
-----  
1 NORMAL
```

REKEY tablespace online

```
select name from v$datafile;
```

NAME

```
-----  
/u01/app/oracle/oradata/TEST18MT/PDB1/system01.dbf  
/u01/app/oracle/oradata/TEST18MT/PDB1/sysaux01.dbf  
/u01/app/oracle/oradata/TEST18MT/PDB1/undotbs01.dbf  
/u01/app/oracle/oradata/TEST18MT/PDB1/users01.dbf  
/u01/app/oracle/oradata/TEST18MT/PDB1/ts_pii01_new.dbf
```

```
[oracle@oracle18 ~]$ dd if=/u01/app/oracle/oradata/TEST18MT/PDB1/ts_pii01_new.dbf  
of=block.dmp_enc_rekey_new bs=8k count=1 skip=131
```

```
[oracle@oracle18 ~]$ md5sum block.dmp_enc  
ab35da0dbe93a006424d94566c906db2  block.dmp_enc
```

```
[oracle@oracle18 ~]$ md5sum block.dmp_enc_rekey_new  
c4f4537701ea28a7f48fc3fab8950c86  block.dmp_enc_rekey_new
```

Changing the Wallet password

Changing the Password-Protected wallet Password

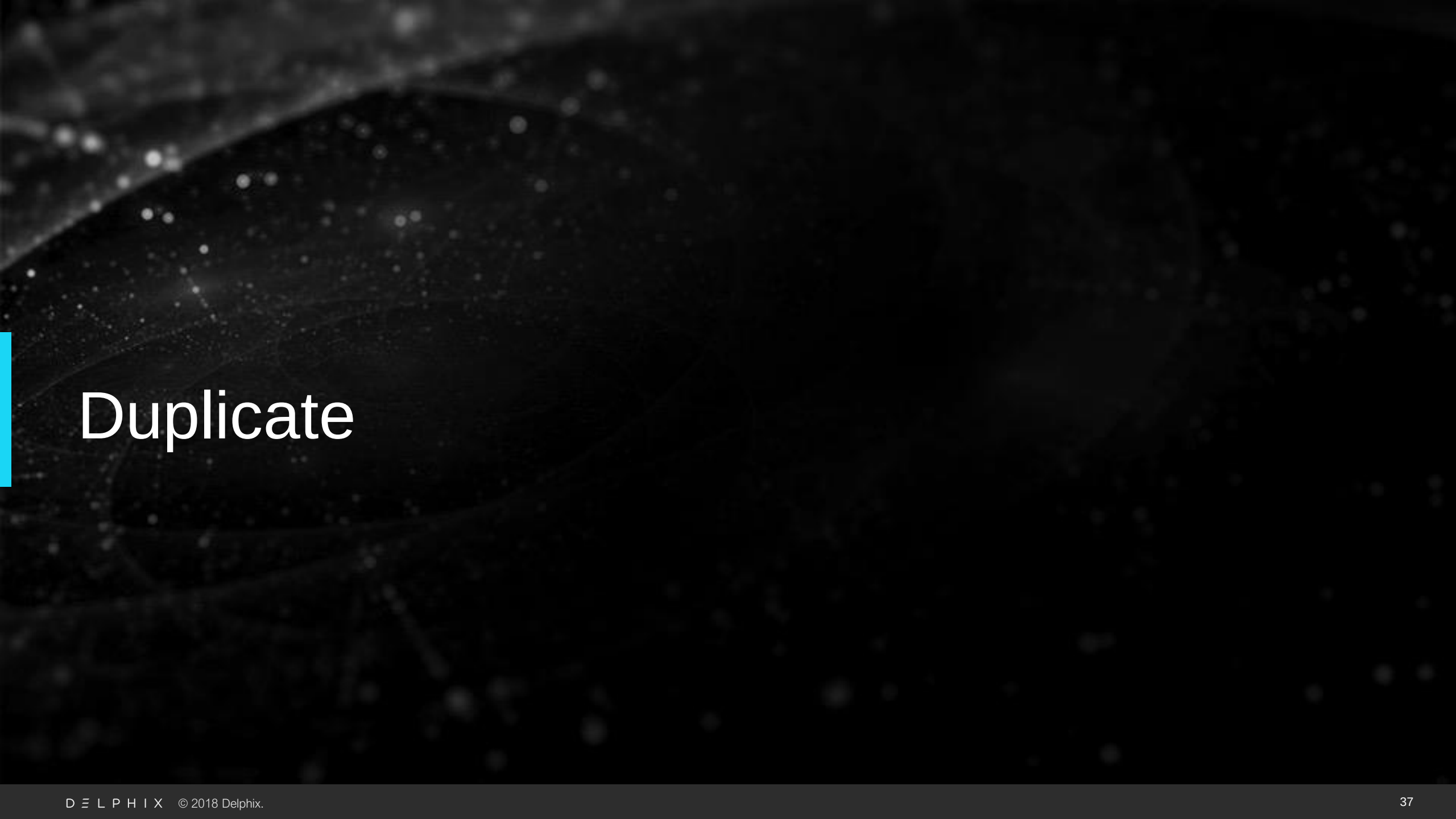
```
administer key management alter keystore password [force keystore]  
identified by old_password set new_password  
with backup;
```

Autologin wallet

Creating a Auto login for wallet

```
administer key management create auto_login keystore from keystore '/path' identified by delphix;
```

```
administer key management create local auto_login keystore from keystore '/path' identified by delphix;
```



Duplicate

Duplicate database with encrypted tables

```
select wrl_type, status from v$encryption_wallet;
```

WRL_TYPE	STATUS
FILE	NOT_AVAILABLE

Duplicate database with encrypted tables

contents of Memory Script:

```
{
  Alter clone database open resetlogs;
}
executing Memory Script
database opened
Finished Duplicate Db at 2019-10-28 11:52:27
Recovery Manager complete.
```

```
select count(*) from pioro.TEST_ENC_COLUMN;
```

```
      COUNT(*)
-----
      266013
```

```
SQL> select * from pioro.TEST_ENC_COLUMN;
select * from pioro.TEST_ENC_COLUMN
      *
```

```
ERROR at line 1:
ORA-28365: wallet is not open
```

Duplicate database with encrypted tablespaces

Oracle instance shut down

RMAN-00571: =====

RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====

RMAN-00571: =====

RMAN-03002: failure of Duplicate Db command at 10/28/2019 11:57:19

RMAN-05501: aborting duplication of target database

RMAN-03015: error occurred in stored script Memory Script

ORA-00283: recovery session canceled due to errors

RMAN-11003: failure during parse/execution of SQL statement: alter database recover logfile
'/u01/app/oracle/oradata/clonedb/FRA/CLONEDB/archivelog/2019_10_28/o1_mf_1_49_gvfovxxl_.arc'

ORA-00283: recovery session canceled due to errors

ORA-28365: **wallet is not open**

Duplicate database with encrypted tablespaces

```
scp /path/ewallet.p12 oracle@172.16.180.12:/path/ewallet.p12
ewallet.p12
```

```
startup nomount pfile=/u01/app/oracle/rman/initclonedb.ora force
ORACLE instance started.
```

```
administer key management set keystore open identified by delphix;
keystore altered.
```

```
Oracle instance shut down
```

```
RMAN-00571: =====
```

```
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
```

```
RMAN-00571: =====
```

```
RMAN-03002: failure of Duplicate Db command at 10/28/2019 12:03:52
```

```
RMAN-05501: aborting duplication of target database
```

```
RMAN-03015: error occurred in stored script Memory Script
```

```
ORA-00283: recovery session canceled due to errors
```

```
RMAN-11003: failure during parse/execution of SQL statement: alter database recover logfile
'/u01/app/oracle/oradata/clonedb/FRA/CLONEDB/archivelog/2019_10_28/o1_mf_1_60_gvfp85bn_.arc'
```

```
ORA-00283: recovery session canceled due to errors
```

```
ORA-28365: wallet is not open
```

Duplicate database with encrypted tablespaces

```
startup nomount pfile=/u01/app/oracle/rman/initclonedb.ora force
```

```
administer key management set keystore open identified by delphix;  
keystore altered.
```

```
administer key management create local auto_login keystore from keystore  
'/u01/app/oracle/admin/wallets/clonedb/' identified by delphix;  
keystore altered.
```

```
ls -l /u01/app/oracle/admin/wallets/clonedb/  
total 16  
-rw-----. 1 oracle oinstall 4232 Oct 28 12:07 cwallet.sso  
-rw-----. 1 oracle oinstall 4171 Oct 28 12:00 ewallet.p12
```

contents of Memory Script:

```
{  
    Alter clone database open resetlogs;  
}
```

executing Memory Script

database opened

Finished Duplicate Db at 2019-10-28 12:10:24



Pluggable databases

Unplug/plug PDB

```
alter pluggable database pdb1 close;  
Pluggable database altered.
```

```
alter pluggable database  pdb1 UNPLUG INTO '/tmp/pdb1.pdb';  
alter pluggable database  pdb1 UNPLUG INTO '/tmp/pdb1.pdb'  
*
```

ERROR at line 1:

ORA-46680: master keys of the container database must be exported

Unplug/plug PDB

```
administer key management export encryption keys with secret "delphix" to '/tmp/pdb1.p12' identified by  
xxx;  
administer key management export encryption keys with secret "delphix" to '/tmp/pdb1.p12' identified by  
xxx  
*
```

ERROR at line 1:

ORA-46658: keystore not open in the container

```
alter pluggable database pdb1 open read write;  
Pluggable database altered.
```

```
alter session set container = pdb1;  
Session altered.
```

```
administer key management export encryption keys with secret "password" to '/tmp/pdb1.key' force  
keystore identified by xxx;  
keystore altered.
```

Unplug/plug PDB

```
alter session set container = cdb$root;  
Session altered.
```

```
alter pluggable database pdb1 UNPLUG INTO '/tmp/pdb1.pdb';  
Pluggable database altered.
```

Unplug/plug PDB

```
create pluggable database pdb1 using '/tmp/pdb1_new.pdb'  
FILE_NAME_CONVERT=('/tmp/', '/u01/app/oracle/oradata/CLT18MT/PDB1/');  
Pluggable database created.
```

```
alter pluggable database pdb1 open read write;  
Warning: PDB altered with errors.
```

```
select message, status from PDB_PLUG_IN_VIOLATIONS where name = 'PDB1';
```

MESSAGE	STATUS
-----	-----
PDB needs to import keys from source.	PENDING

```
alter session set container=pdb1;  
Session altered.
```

```
administer key management import encryption keys with secret "password" from '/tmp/pdb1.key' force  
keystore identified by delphixclone with backup;  
keystore altered.
```

Unplug/plug PDB

```
alter session set container=cdb$root;  
Session altered.
```

```
alter pluggable database pdb1 close;  
Pluggable database altered.
```

```
alter pluggable database pdb1 open read write;  
Pluggable database altered.
```

Unplug/plug PDB

```
alter session set container=pdb1;  
Session altered.
```

```
select count(*) from pioro.test_ts_enc;  
select count(*) from pioro.test_ts_enc  
                        *
```

```
ERROR at line 1:  
ORA-28365: wallet is not open
```

```
administer key management set keystore open identified by xxxclone;  
keystore altered.
```

```
select count(*) from pioro.test_ts_enc;
```

```
      COUNT (*)  
-----  
          9999
```

Unplug/plug PDB

```
alter pluggable database pdb1 unplug into '/tmp/pdb1.xml' encrypt using myhiddenpassword;  
Pluggable database altered.
```

```
create pluggable database pdb1 using '/tmp/pdb1.xml' nocopy tempfile reuse decrypt using  
myhiddenpassword keystore identified by xxxclone;  
Pluggable database created.
```

```
alter pluggable database pdb1 unplug into '/tmp/pdb1.pdb' encrypt using myhiddenpassword;  
Pluggable database altered.
```

```
create pluggable database pdb1 using '/tmp/pdb1.pdb'  
file_name_convert=('/tmp/', '/u01/app/oracle/oradata/CLT18MT/PDB1/') decrypt using myhiddenpassword  
keystore identified by xxxclone;  
Pluggable database created.
```

Export single key – useful for PDB to lower environment

```
select key_id, tag, ACTIVATION_TIME from v$encryption_keys;
```

KEY_ID	TAG	ACTIVATION_TIME
AZROuKlafk96v73nWf2gGtoAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb	05-NOV-19 03.28.07.172800 PM +00:00
AdUsu9sNsU8wv/vq4J0OHXYAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey 1535	05-NOV-19 03.35.50.950753 PM +00:00
AWU4CsWVu0+HvwA3rKskvZAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey 1535	05-NOV-19 03.35.49.212075 PM +00:00
AZZzCkrae08Zv1Th20jqQ3MAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey 1212	05-NOV-19 12.15.57.553021 PM +00:00
AcZm34lI509ov5gqgkASU3EAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey 1314	04-NOV-19 01.14.24.843571 PM +00:00

```
alter session set container = cdb$root;
Session altered.

administer key management export encryption keys with secret "exportpassword" to
'/tmp/pdb_keys_only_current.p12' identified by delphix with IDENTIFIER IN '
AdUsu9sNsU8wv/vq4J0OHXYAAAAAAAAAAAAAAAAAAAAAAAAAAAA';
keystore altered.
```

Clone PDB

```
create pluggable database pdb2 from pdb1 file_name_convert=('PDB1','PDB2');  
create pluggable database pdb2 from pdb1 file_name_convert=('PDB1','PDB2')  
*
```

ERROR at line 1:

ORA-46697: Keystore password required.

```
create pluggable database pdb2 from pdb1 file_name_convert=('PDB1','PDB2') keystore identified by xxx;
```

Pluggable database created.

Clone PDB

```
@show_keys
```

	CON_ID	KEY_ID	TAG
3	AZAsw6mosk/pv7DQnWwc2XMAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey	1212
1	AZZzCkrae08Zv1Th20jqQ3MAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey	1212
3	AWYbVyB7b08Iv2cLfpFuetcAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey	1554

```
alter session set container = pdb2;  
Session altered.
```

```
select * from pioro.test_ts_enc where rownum < 10;  
select * from pioro.test_ts_enc where rownum < 10  
      *
```

```
ERROR at line 1:  
ORA-28365: wallet is not open
```

Clone PDB

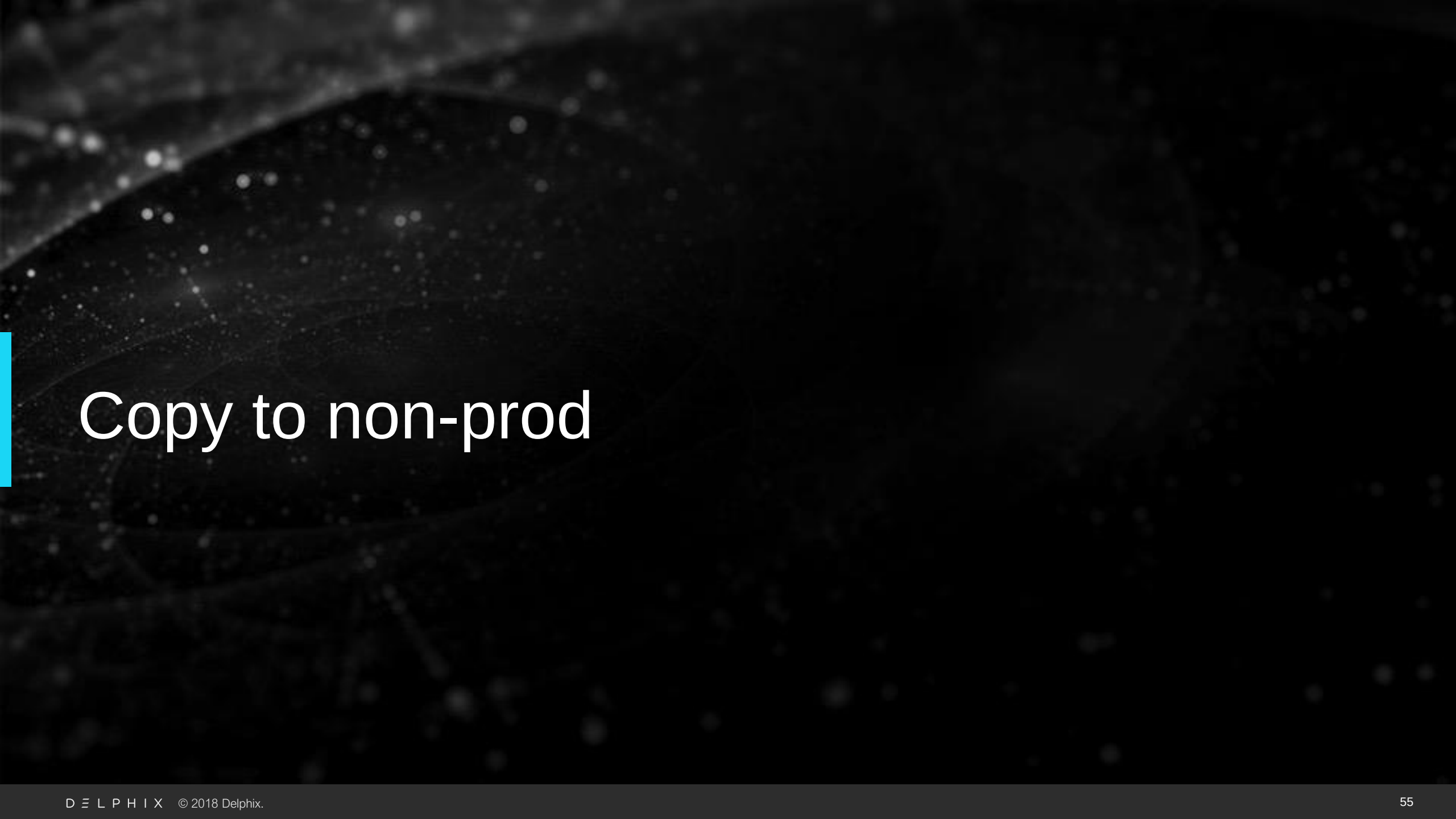
```
administer key management set keystore open force keystore identified by delphix;  
keystore altered.
```

```
administer key management set key using tag "rekey new pdb" identified by delphix with backup using  
"backup_0511_before_newpdb" container = all;
```

```
keystore altered.
```

```
@show_keys
```

CON_ID	KEY_ID	TAG
6	AZROuKlafk96v73nWf2gGtoAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb
3	Ad9MKwByoU9yv1bOAnw2NhkAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb
1	AeVcqaBYnU9+v+z9zwgarIEAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb



Copy to non-prod

Copy to non-prod

Is it Possible to Remove/Disable TDE?

NO

Doc ID 2488898.1 - Last update: 21-Jan-2019 !!!

Deleting the TDE wallet will not disable TDE. Once TDE wallet is configured, the wallet should never be deleted or recreated.

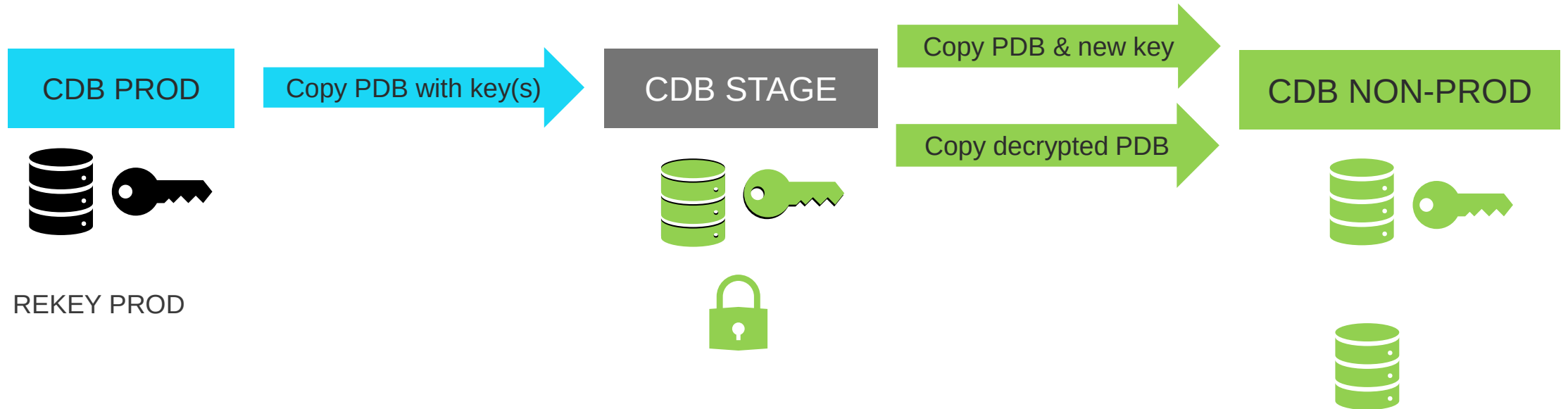
...

Even if there are no encrypted objects in the database, the TDE wallet has to be present in the wallet location. It does not cause any harm.

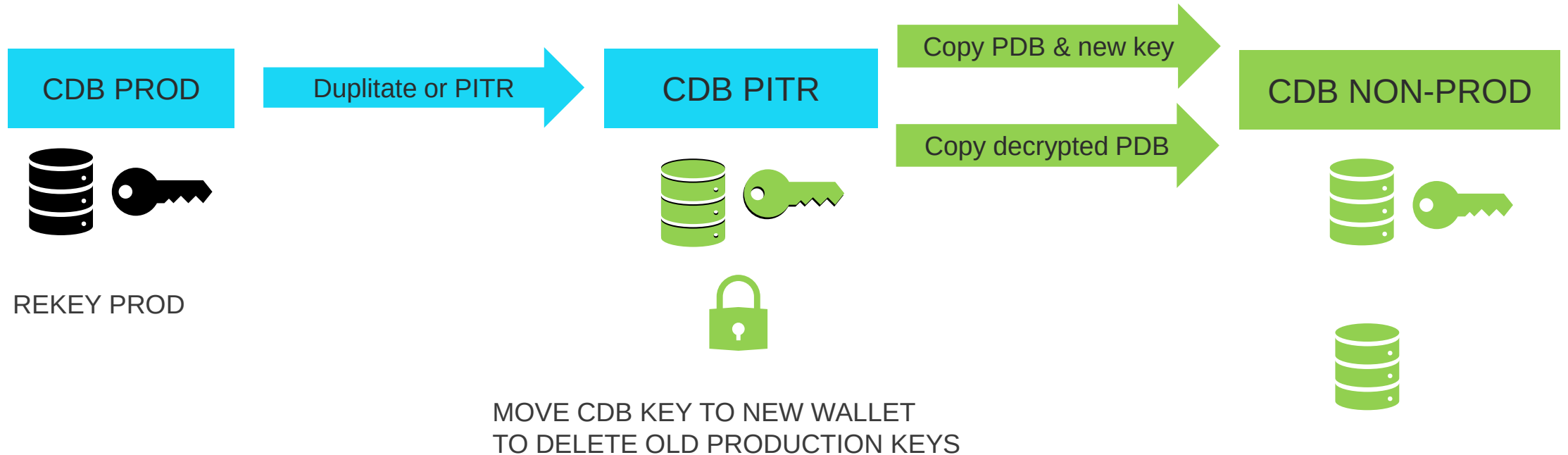
...

Recreating the wallet using any parameters is not supported. Oracle Support/Development team will not help in resolving any issues arising due to such operations.

How to move data to non-prod



How to move data to non-prod



Decrypt data

```
alter tablespace TS_PII offline normal;  
Tablespace altered.
```

```
alter tablespace TS_PII encryption offline decrypt  
*
```

ERROR at line 1:

```
ORA-28435: cannot decrypt data file  
/u01/app/oracle/oradata/TEST18MT/PDB1/ts_pii01.dbf which is not encrypted with  
the database key
```

ORA-28435: Cannot Decrypt Data File %s Which Is Not Encrypted With The Database Key (From ALTER DATABASE DATAFILE ... DECRYPT) (Doc ID 2406173.1)

This is expected behavior ...

Use:

```
alter table ... move tablespace ts_non_encrypted;
```

Decrypt data

...

LOGFILE

```
GROUP 1 '/prodb/delphix/VTEST/datafile/PROD/onlinelog/o1_mf_1_gc1cbf1b_.log' SIZE 52428800,  
GROUP 2 '/prodb/delphix/VTEST/datafile/PROD/onlinelog/o1_mf_2_gc1cbf4q_.log' SIZE 52428800,  
GROUP 3 '/prodb/delphix/VTEST/datafile/PROD/onlinelog/o1_mf_3_gc1cbf57_.log' SIZE 52428800
```

2019-04-30T12:14:54.876814+01:00

Clearing online redo logfile 1 complete

Clearing online redo logfile 2 complete

Clearing online redo logfile 3 complete

Online log /prodb/delphix/VTEST/datafile/PROD/onlinelog/o1_mf_1_gc1cbf1b_.log: Thread 1 Group 1 was previously cleared

Online log /prodb/delphix/VTEST/datafile/PROD/onlinelog/o1_mf_2_gc1cbf4q_.log: Thread 1 Group 2 was previously cleared

Online log /prodb/delphix/VTEST/datafile/PROD/onlinelog/o1_mf_3_gc1cbf57_.log: Thread 1 Group 3 was previously cleared

Online log /prodb/delphix/VTEST/datafile/VTEST/onlinelog/o1_mf_4_gdjcnr1_.log: Thread 1 Group 4 was previously cleared

Online log /prodb/delphix/VTEST/datafile/VTEST/onlinelog/o1_mf_5_gdjcnso_.log: Thread 1 Group 5 was previously cleared

Completed: alter database open resetlogs

2019-04-30T12:14:59.915339+01:00

Decrypt data

2019-04-30T12:15:01.117125+01:00

Shutting down instance (abort) (OS id: 2460)

...

2019-04-30T12:15:23.271945+01:00

ALTER DATABASE OPEN

2019-04-30T12:15:23.756549+01:00

Beginning crash recovery of 1 threads

parallel recovery started with 7 processes

2019-04-30T12:15:23.938460+01:00

Recovery of Online Redo Log: Thread 1 Group 2 Seq 2 Reading mem 0

Mem# 0: /proddb/delphix/VTEST/datafile/PROD/onlinelog/o1_mf_2_gc1cbf4q_.log

2019-04-30T12:15:23.938755+01:00

Completed redo application of 0.05MB

2019-04-30T12:15:23.979012+01:00

Slave encountered ORA-28365 exception during crash recovery

Slave exiting with ORA-28365 exception

2019-04-30T12:15:23.979455+01:00

Errors in file /product/oracle/12c/ora/diag/rdbms/VTEST/VTEST/trace/VTEST_p003_2601.trc:

ORA-28365: wallet is not open

Decrypt data

- Make sure there is no sign of encryption in:
 - redo logs (switch or recreate them)
 - undo tablespace (recreate ?)
 - archive log ?
- If you are cloning using snapshot / crash consistency tools – make sure you checkpoint your database

Move key between production / non-production

- No easy method so far:
 - Doc ID 2216279.1 - How to delete old master keys from 12c TDE keystore (wallet).
 - Requires a open database but also requires to restart a database during a process
- Move key command [18c]
 - `ADMINISTER KEY MANAGEMENT MOVE [ENCRYPTION] KEYS`
 - Doesn't work as expected – needs more investigation

On production

```
SQL> @show_keys
```

CON_ID	KEY_ID	TAG
6	AdUsu9sNsU8wv/vq4J0OHXYAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey 1535
3	AS+sROMqI08Xv18N62I2Uf0AAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey 1535
1	AWU4CsWVu0+HvwA3rKskvZAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey 1535
6	AZROuKlafk96v73nWf2gGtoAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb
3	Ad9MKwByoU9yv1bOAnw2NhkAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb
1	AeVcqabYnU9+v+z9zwgarIEAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb

```
administer key management export encryption keys with secret "exportpassword" to '/tmp/clone.p12'
identified by delphix with IDENTIFIER IN
'AZROuKlafk96v73nWf2gGtoAAAAAAAAAAAAAAAAAAAAAAAAAAAA', 'Ad9MKwByoU9yv1bOAnw2NhkAAAAAAAAAAAAAAAAAAAAAAAAAAAA', 'AeVcqabYnU9+v+z9zwgarIEAAAAAAAAAAAAAAAAAAAAAAAAAAAA';
keystore altered.
```

On any open database

```
[oracle@oracle18tgt oracle]$ echo $TNS_ADMIN  
/u01/app/oracle/tns
```

```
select con_id, wrl_parameter, status, wallet_type, wallet_type from v$encryption_wallet;
```

```
CON_ID WRL_PARAMETER  
-----  
1 /u01/app/oracle/admin/wallets/test18mt/  
2  
3
```

```
administer key management set keystore open force keystore identified by clone;  
keystore altered.
```

On any open database

```
@show_keys
```

```
no rows selected
```

```
administer key management import encryption keys with secret "exportpassword" from '/tmp/clone.p12'  
force keystore identified by clone with backup;  
keystore altered.
```

```
@show_keys
```

CON_ID	KEY_ID	TAG
6	AZROuKlafk96v73nWf2gGtoAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb
3	Ad9MKwByoU9yv1bOAnw2NhkAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb
1	AeVcqqaBYnU9+v+z9zwgarIEAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb

On target database – before duplication

```
[oracle@oracle18tgt oracle]$ . oraenv
ORACLE_SID = [clt18mt] ? clt18mt
```

```
administer key management set keystore open force keystore identified by clone;
keystore altered.
```

```
@show_keys
```

	CON_ID	KEY_ID	TAG
6	AZROuKlafk96v73nWf2gGto	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb
3	Ad9MKwByoU9yv1bOAnw2Nhk	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb
1	AeVcqqaBYnU9+v+z9zwgarIE	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	rekey new pdb

On target database – before duplication

Run duplicate of PDB1

executing Memory Script

sql statement: alter pluggable database all open

Finished Duplicate Db at 06-NOV-19

sqlplus / as sysdba

select * from pioro.TEST_TS_ENC where rownum < 10;

ID CC

```
-----  
1 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
2 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
3 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

TDE backdoor ?

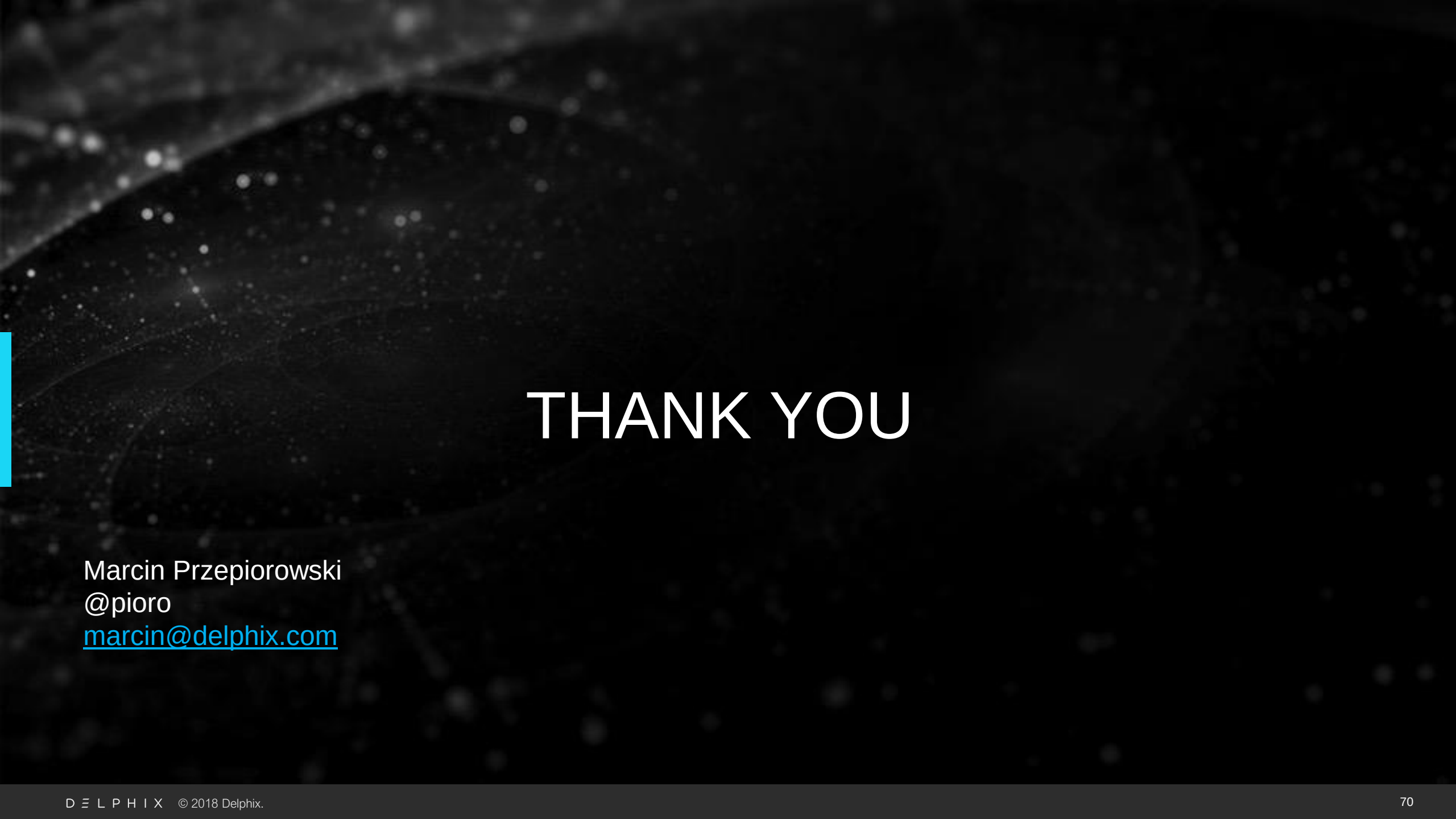
```
select KSPPINM, KSPDESC from x$ksppi where ksppinm like '_db_discard_lost_masterkey';
```

KSPPINM

KSPDESC

_db_discard_lost_masterkey

discard lost masterkey handles



THANK YOU

Marcin Przepiorowski
@pioro
marcin@delphix.com