Hot cloning of pdb with TDE

SENARIO:

We have Two containers **DB19C** & **CONTAINER2.** Both of them have Transparent Data Encryption **(TDE).**
Our task is to clone PDB from **DB19C** TO **CONTAINER2** while the **PDB** is up and running in the source container.

PROCESS OVERVIEW

- ➢ **First we will check character set.**   **##destination char >= source**
- ➢ **Verifying local undo mode.**   **##source**
- ➢ **Creating public database link.**   **##destination**
- ➢ **Executing CREATE PLUGGABLE command**   **##destination**

PRE-REQUISITES FOR HOT CLONING:

- ➢ Container should be Oracle12.2 onwards.
- ➢ Same database version and patch level
- ➢ Destination Container must have Same or higher character set
- ➢ Source PDB must be in LOCAL UNDO mode
- ➢ Container of PDB must be in Archive log mode, else PDB should be open in read only mode.
- ➢ The Source DB/container must be in archivelog mode.
- ➢ Both DBs must have the same endianness.
- ➢ Both DBs must either have the same options installed, or the source database (container/non-container) must have a subset of those present on the destination database (container).
- ➢ 12.1.0.2 => Hot clone is possible when no DML are in progress (**Doc ID 2538667.1)**
- ➢ 12.2.0.1 => Onward it is possible to perform hot clone when DML are in progress when local undo used. (**Doc ID 2538667.1)**

As far as the concern of database version and patch level so it's not a big deal we can upgrade the database after cloning.

As far as the concern is about character set then if the destination container which will maintain the cloned PDB is using **AL32UTF8** as Database character set and **AL16UTF16** as national character set then pdbs with any character set can be cloned in it else we will changes the character set of the pdb.

Source:

```
set lines 200;
column PROPERTY_NAME format a30;
column PROPERTY_VALUE format a20;
column DESCRIPTION format a50;
SELECT property_name, property_value, description
FROM database_properties
WHERE property_name IN ('NLS_CHARACTERSET', 'NLS_NCHAR_CHARACTERSET')
ORDER BY 1;
```

```
SQL> set lines 200;
SQL> column PROPERTY_NAME format a30;
SQL> column PROPERTY_VALUE format a20;
SQL> column DESCRIPTION format a50;
SQL> SELECT property_name, property_value, description
  2  FROM database_properties
  3  WHERE property_name IN ('NLS_CHARACTERSET', 'NLS_NCHAR_CHARACTERSET')
  4  ORDER BY 1;

PROPERTY_NAME                  PROPERTY_VALUE       DESCRIPTION
------------------------------ -------------------- --------------------------------------------------
NLS_CHARACTERSET               AL32UTF8             Character set
NLS_NCHAR_CHARACTERSET         AL16UTF16            NCHAR Character set
```

SELECT name, local_undo **FROM** v$pdbs;

```
SQL> SELECT name, local_undo FROM v$pdbs;

NAME                                                                             LOCAL_UNDO
-------------------------------------------------------------------------------- ----------
PDB$SEED                                                                                  1
PDB1                                                                                      1
     PDB                                                                                  1
```

NOTE: As can be seen Local Undo Mode is enabled already. If in your case it is not enabled then the value for the LOCAL_UNDO column will be 0 so in such case you will have to enable the local undo management by using the following procedure.

```
SQL>> SHUTDOWN IMMEDIATE;

SQL>> STARTUP UPGRADE;

SQL>> ALTER DATABASE SET LOCAL UNDO ON;

SQL>> SHUTDOWN IMMEDIATE;

SQL>> STARTUP
```

CREATE USER c##cloneusr IDENTIFIED BY 123 DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp CONTAINER=ALL;

GRANT CREATE SESSION, CREATE PLUGGABLE DATABASE, SYSOPER TO c##cloneusr CONTAINER = ALL;

SHOW CON_NAME;

```
SQL> CREATE USER c##cloneusr IDENTIFIED BY 123 DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp CONTAINER=ALL;

User created.

SQL> GRANT CREATE SESSION, CREATE PLUGGABLE DATABASE, SYSOPER TO c##cloneusr CONTAINER = ALL;

Grant succeeded.

SQL> SHOW CON_NAME;

CON_NAME
------------------------------
CDB$ROOT
```

Dost Muhammad Khalil (OCP)

<u>DESTINATION:</u>

CREATE PUBLIC DATABASE LINK clonemypdb CONNECT TO c##cloneusr IDENTIFIED BY "123" USING 'CLONE_TNS';

NOTE: Add TNS entry in tnsnames.ora file for the source and check using tnsping wether the tns entry is correct.
When DB_Link is created and you issue the CREATE PLUGGABLE DATABASE command you may encounter invalid username/password in such case copy your command to notepad double check the password and replace the double quotes with new one in notepad. Most of the time double quotes of pdf or word are not understandable by sqlplus.

```
SQL> !tnsping clone_tns

TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 29-AUG-2024 12:13:23

Copyright (c) 1997, 2023, Oracle.  All rights reserved.

Used parameter files:
/u01/app/oracle/product/19.0.0/dbhome_1/network/admin/sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = WSISBDBUAT.mobilinkbank.com)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_
NAME = DB19C)))
OK (10 msec)
```

CREATE PLUGGABLE DATABASE � FROM MONETAPDB@clonemypdb keystore identified by "YOUR_PASSWORD";

Clone is in progress and this block is executing in source.

```
SQL>  begin
  2   for i in 1..1000000000 loop
  3   insert into dost.test values(i,'Dost');
  4   commit;
  5   end loop;
  6   end;
  7   /
```

As can be seen the cloning in progress.

```
SQL>
SQL>
SQL> CREATE PLUGGABLE DATABASE █████████ FROM ████████@clonemypdb keystore identified by "██████";
```

Bonus Point : Transactions which are done on pdb in source before the recovery started on pdb in destination will be reflect in the pdb on destination and the transactions done after the recovery started will not be present on destination pdb.