**Prechecks for PDB Cloning:**

1) Verify if LOCAL UNDO is enabled on source.

=============================================

column property\_name format a30

column property\_value format a30

select property\_name, property\_value

from   database\_properties

where  property\_name = 'LOCAL\_UNDO\_ENABLED';

If local undo is enabled we can do the hot clone else we need to put the pdb in read only mode during cloning.

Steps to enable local undo.

conn / as sysdba

srvctl stop database -d ${ORACLE\_UNQNAME}

startup upgrade;

alter database local undo on;

srvctl stop database -d ${ORACLE\_UNQNAME}

srvctl start database -d ${ORACLE\_UNQNAME}

2) Execute the below commands on source and target CDB's.

========================================================

alter system set global\_names=FALSE;

grant create pluggable database to system container=all;

3) Unlock SYSTEM account at both source and target CDB/PDB.

==========================================================

alter user SYSTEM account unlock;

++++++++++++++++++++++**PDB Cloning Steps**++++++++++++++++++++++++++++++++++++

-- Below Steps at TARGET

0) Drop pluggable database

==============================

ALTER PLUGGABLE DATABASE <PDB\_NAME> close instances=all;

DROP PLUGGABLE DATABASE <PDB\_NAME> INCLUDING DATAFILES;

1) Make a note of the services running at target.

===============================================

srvctl status database -d db\_unique\_name –v

2) Create database link using below command.

================================================

Replace the below values in the below command.

<password>

<hostname>

<port#>

<service\_name>

create database link CLONE\_PDB connect to  system identified by <password> using '(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = <hostname>)(PORT = <port#>)) (CONNECT\_DATA = (SERVER = DEDICATED) (SERVICE\_NAME = <service\_name>)))';

3) Test the database link.

==============================================

select name from v$pdbs@CLONE\_PDB;

4) Start the source PDB in read only mode( only if local undo is not enabled) .

=====================================================================================

\*\*\* Convert source database to read only

alter pluggable database <pdb\_name> open read only;

5) Initiate the pdb clone using below script.

======================================================================================

@ Target

-- Update <ORACLE\_SID> , <ORACLE\_HOME\_PATH>, <path\_to\_store\_logs>, <pdb\_name>, <source\_pdb\_name> and <password>

vi pdb\_clone.sh

export DTIME=`date +"%d%b%y\_%H%M%S"`

export ORACLE\_SID=<ORACLE\_SID>

export ORACLE\_HOME=<ORACLE\_HOME\_PATH>

export PATH=$ORACLE\_HOME/bin:$PATH

export LOGS\_PATH=<path\_to\_store\_logs>

sqlplus / as sysdba << EOF > ${LOGS\_PATH}/${ORACLE\_SID}\_clone\_${DTIME}.out

spool ${LOGS\_PATH}/${ORACLE\_SID}\_clone${DTIME}.log

select name, open\_mode from v\$database;

set time on timing on feedback on verify on head on echo on

select name from v\$pdbs@CLONE\_PDB;

select name from v\$database@CLONE\_PDB;

create pluggable database <pdb\_name> from <source\_pdb\_name>@CLONE\_PDB storage unlimited tempfile reuse keystore identified by "<password>" STANDBYS=NONE FILE\_NAME\_CONVERT=('+DATAC2','+DATAC2');

SELECT name, open\_mode FROM v\$pdbs WHERE name = '<pdb\_name>';

spool off

EOF

-- invoke script

nohup sh pdb\_clone.sh &

-- Target PDB will be in mount state. Open it using below SQL.

alter pluggable database <pdb\_name> open instances=all;

6) Verify plug\_in\_violations

=======================================================================

set pages 999 lines 200

col cause for a10

col name for a10

col message for a35 word\_wrapped

select name,cause,type,message,status from PDB\_PLUG\_IN\_VIOLATIONS where name='<pdb\_name>' and STATUS='PENDING';

Resolve violations if any.

alter pluggable database <pdb\_name> close immediate instances=all;

alter pluggable database <pdb\_name> open read write instances=all; --> make sure pdb is not started in restricted mode.

alter pluggable database <pdb\_name> save state instances=all; --> execute only if pdb is not in restricted mode.

7)Execute the utlrp.sql script couple of times and validate the registry.

==============================================================================

@?/rdbms/admin/utlrp.sql

set lines 300 pages 300

col COMP\_NAME for a40

col STATUS for a30

col VERSION for a30

select COMP\_NAME,VERSION,status from dba\_registry;

**Post Cloning steps:**

1)Lock SYSTEM account at both source and target CDB/PDB.

==========================================================

alter user SYSTEM account lock;

2) Ensure all database services are up at target

===================================================

During step 1 of the previous section, a list of services was collected.

3) Drop db link at target

======================

drop database link CLONE\_PDB;

**--------------------------**

**+++ ERRORS and Fixes**

**--------------------------**

**[1]**

**--- ERROR ---**

**Connected to: Oracle Database 19c EE Extreme Perf Release 19.0.0.0.0 - Production**

**ORA-39002: invalid operation**

**ORA-39405: Oracle Data Pump does not support importing from a source database with TSTZ version 38 into a target database with TSTZ version 32.**

--- Cause ---

It is due to timezone version mismatch between source and target

--- Fix ---

You to upgrade timezone to match the source version

pre-upgrade steps

alter pluggable database UNTRAUG close immediate instances = all;

-- PDB open mode will be - mounted

alter pluggable database UNTRAUG open upgrade;

-- PDB open mode will be - migrate

Pluggable database altered.

alter session set container=UNTRAUG;

EXEC DBMS\_DST.CREATE\_AFFECTED\_TABLE('AFFECTED\_TABLES');

EXEC DBMS\_DST.CREATE\_ERROR\_TABLE('ERR\_TABLE');

exec DBMS\_DST.BEGIN\_PREPARE(38);

exec DBMS\_DST.FIND\_AFFECTED\_TABLES(affected\_tables => 'AFFECTED\_TABLES', log\_errors\_table => 'ERR\_TABLE', log\_errors => TRUE);

exec DBMS\_DST.END\_PREPARE();

select \* from ERR\_TABLE;

-- there should be "no rows selected"

exec DBMS\_DST.BEGIN\_UPGRADE(38);

alter pluggable database UNTRAUG close immediate instances = all;

alter pluggable database UNTRAUG open read write;

SQL> sho pdbs

CON\_ID CON\_NAME OPEN MODE RESTRICTED

---------- ------------------------------ ---------- ----------

4 UNTRAUG READ WRITE NO

set lines 200

column name format a30

select INST\_ID, CON\_ID, NAME, OPEN\_MODE, RESTRICTED

from gv$pdbs

where CON\_ID=4;

INST\_ID CON\_ID NAME OPEN\_MODE RES

---------- ---------- ------------------------------ ---------- ---

1 4 UNTRAUG READ WRITE NO

3 4 UNTRAUG MOUNTED

4 4 UNTRAUG MOUNTED

2 4 UNTRAUG MOUNTED

SQL> alter pluggable database UNTRAUG close immediate instances = all;

Pluggable database altered.

SQL> sho pdbs

CON\_ID CON\_NAME OPEN MODE RESTRICTED

---------- ------------------------------ ---------- ----------

4 UNTRAUG MOUNTED

SQL> exit

Disconnected from Oracle Database 19c EE Extreme Perf Release 19.0.0.0.0 - Production

Version 19.16.0.0.0

[oracle@ausyd25db03vcn1 /var/oracle/export/sde>] s

SQL\*Plus: Release 19.0.0.0.0 - Production on Tue Nov 29 22:01:56 2022

Version 19.16.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:

Oracle Database 19c EE Extreme Perf Release 19.0.0.0.0 - Production

Version 19.16.0.0.0

SQL> alter pluggable database UNTRAUG open read write;

Pluggable database altered.

SQL> set lines 200

column name format a30

select INST\_ID, CON\_ID, NAME, OPEN\_MODE, RESTRICTED

from gv$pdbs

where CON\_ID=4;SQL> SQL> 2 3

INST\_ID CON\_ID NAME OPEN\_MODE RES

---------- ---------- ------------------------------ ---------- ---

1 4 UNTRAUG READ WRITE NO

2 4 UNTRAUG MOUNTED

3 4 UNTRAUG MOUNTED

4 4 UNTRAUG MOUNTED

SQL> alter session set container=UNTRAUG;

Session altered.

VAR numfail number

BEGIN

DBMS\_DST.UPGRADE\_DATABASE(:numfail,

parallel => TRUE,

log\_errors => TRUE,

log\_errors\_table => 'SYS.DST$ERROR\_TABLE',

log\_triggers\_table => 'SYS.DST$TRIGGER\_TABLE',

error\_on\_overlap\_time => TRUE,

error\_on\_nonexisting\_time => TRUE);

DBMS\_OUTPUT.PUT\_LINE('Failures:'|| :numfail);

END;

/

PL/SQL procedure successfully completed.

VAR numfail number

BEGIN

DBMS\_DST.END\_UPGRADE(:numfail);

DBMS\_OUTPUT.PUT\_LINE('Failures:'|| :numfail);

END;

/

PL/SQL procedure successfully completed.

SQL>

SQL>

SQL>

SQL>

SQL> select \* from V$TIMEZONE\_FILE;

FILENAME VERSION CON\_ID

-------------------- ---------- ----------

timezlrg\_38.dat 38 0

SQL> alter pluggable database UNTRAUG close immediate instances = all;

Pluggable database altered.

SQL> alter pluggable database UNTRAUG open read write instances = all;

Pluggable database altered.

**[2]**

**--- ERROR ---**

**ORA-39083: Object type USER:"DS\_USER" failed to create with error:**

**ORA-12911: permanent tablespace cannot be temporary tablespace**

--- FIX ---

Manually create user with like so

create user DS\_USER identified by values 'ISweedrs\_\_2020222' default tablespace USERS temporary tablespace TEMP1 profile MERCER\_APP\_PROFILE;

and rerun import

**[3]**

**--- ERROR ---**

**If there are any missing role, please create it.**

--- FIX ---

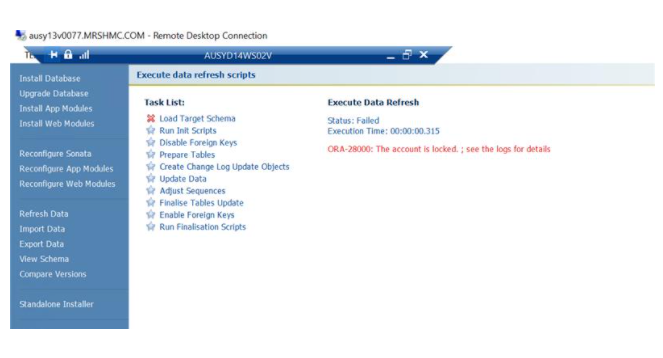
CREATE ROLE "ARCHIVE\_READ\_ONLY\_AES";

--- Rerun import

impdp directory=CO2214937 dumpfile=Three\_19dec\_%U.dmp logfile=db\_19dec\_Imp\_sde01.log schemas=DMU\_ADMIN,DS\_USER,UREGISTRY\_READ

**[4]**

**--- ERROR ---**



Issue:

When an environment is cloned [PDB] sometimes the dba\_servcies lists the service from the source environment at target

and that causes issues [wrong application connectivity]

--- Fix ---

To solve this stop and delete the undesired service from target.

eg: If Q was cloned from M then we got to make sure that there are no M entries in Q.

-- To list services

col name for a30

col NETWORK\_NAME for a30

set lines 50

set lines 200

SELECT name,network\_name FROM dba\_services;

-- Stop service

exec dbms\_service.stop\_service('UNTRAUQ\_SRVC', DBMS\_SERVICE.ALL\_INSTANCES);

-- Then Delete servcie

BEGIN

DBMS\_SERVICE.DELETE\_SERVICE(

service\_name => 'UNTRAUQ\_SRVC'

);

END;

/