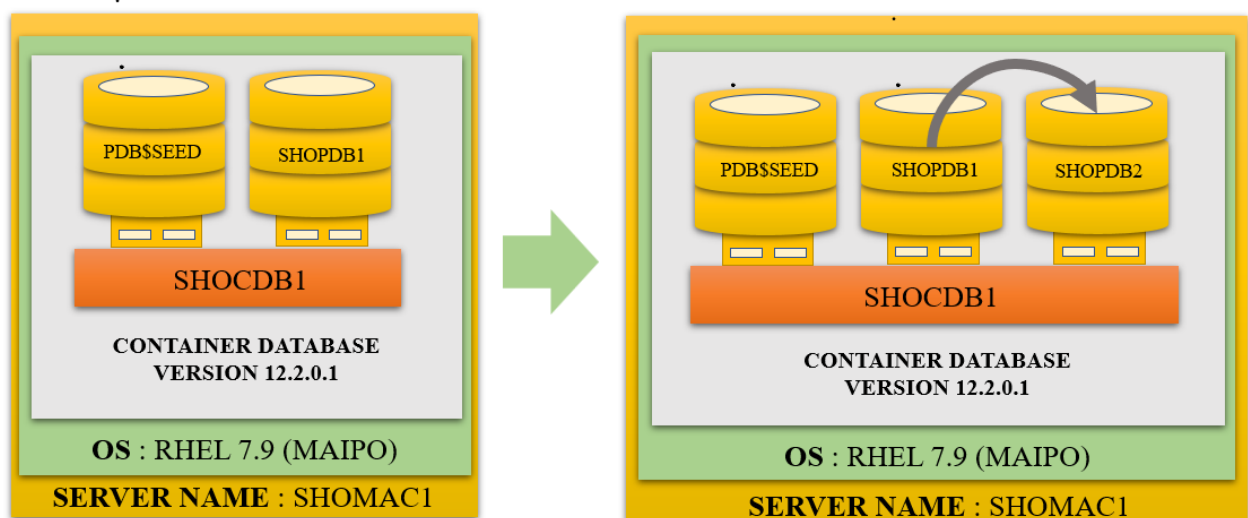


CLONE ORACLE 12c PLUGGABLE DATABASE IN SAME CDB

Clone pluggable database PDB (SHOPDB2) from another pluggable database PDB (SHOPDB1) in the same container database (SHOCDB1)



Let's understand the current environment

We can see that we have 1 container database i.e., SHOCDB1 running over the server i.e., SHOMAC1

```
[oracle@SHOMAC1 ~]$
[oracle@SHOMAC1 ~]$ ps -ef|grep -i pmon |grep -v grep
oracle 26166 1 0 Aug31 ? 00:00:11 ora_pmon_shocdb1
[oracle@SHOMAC1 ~]$
[oracle@SHOMAC1 ~]$
```

When we login into the container database i.e., SHOCDB1

We can see that it has below PDBS

The pluggable database: PDB\$SEED, SHOPDB1

```

[oracle@SHOMAC1 ~]$ . oraenv
ORACLE_SID = [shoaibncdb] ? shocdb1
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@SHOMAC1 ~]$ !sq
sqlplus / as sysdba

SQL*Plus: Release 12.2.0.1.0 Production on Wed Aug 31 20:20:29 2022

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SYS@shocdb1 31-AUG-22>show pdbs

      CON_ID CON_NAME                                OPEN MODE  RESTRICTED
-----
      2 PDB$SEED                                READ ONLY  NO
      3 SHOPDB1                                READ WRITE NO
SYS@shocdb1 31-AUG-22>
SYS@shocdb1 31-AUG-22>

```

The location of the datafiles

When we login into the container database i.e., SHOCDB1. The location of the datafiles for the PDBs are shown as below:

The container database: SHOCDB1 → /u01/app/oracle/oradata/SHOCDB1/datafile/

The Pluggable database: SHOPDB1 →
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/

Note: here we can see that the datafile's naming convention is using OMF (oracle managed format) and SHOPDB1 PDB has a GUID as E79387E71AAD6720E0536538A8C001D5

```

SYS@shocdb1 31-AUG-22>col name for a120
SYS@shocdb1 31-AUG-22>select name from v$datafile;

NAME
-----
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_system_kjzyld81_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_sysaux_kjzy2j5z_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_undotbs1_kjzy3999_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_system_kjzy5sh2_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_sysaux_kjzy5scj_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_users_kjzy3bdx_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_undotbs1_kjzy5sh8_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_system_kjzyhq30_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_sysaux_kjzyhq38_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_undotbs1_kjzyhq39_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_users_kjzyjlcw_.dbf

11 rows selected.

SYS@shocdb1 31-AUG-22>

```

Let's check the status of CDB undo mode?

You can configure a CDB to use local undo in every container or to use shared undo (default) for the entire CDB.

A CDB runs either in local or shared undo mode. The undo mode applies to the entire CDB. Therefore, every container either uses shared undo or local undo.

To determine the current CDB undo mode, run the following query in the CDB root:

Col PROPERTY_NAME for a25

Col PROPERTY_VALUE for a19

```
SELECT PROPERTY_NAME, PROPERTY_VALUE
FROM DATABASE_PROPERTIES
WHERE PROPERTY_NAME = 'LOCAL_UNDO_ENABLED';
```

If the output of the query returns TRUE for the PROPERTY_VALUE, then the CDB is in local undo mode. Otherwise, the CDB is configured in shared undo mode.

```
SYS@shocdb1 31-AUG-22>
SYS@shocdb1 31-AUG-22>Col PROPERTY_NAME for a25
SYS@shocdb1 31-AUG-22>Col PROPERTY_VALUE for a19
SYS@shocdb1 31-AUG-22>SELECT PROPERTY_NAME, PROPERTY_VALUE FROM DATABASE_PROPERTIES
 2 WHERE PROPERTY_NAME = 'LOCAL_UNDO_ENABLED';

PROPERTY_NAME          PROPERTY_VALUE
-----
LOCAL_UNDO_ENABLED     TRUE

SYS@shocdb1 31-AUG-22>
```

Note: if the CDB is configured in shared undo mode, then the pluggable database PDB must be in open read-only before cloning, follow the below 2 steps before cloning.

STEP 1 - Close the pluggable database PDB.

alter pluggable database SHOPDB1 close immediate;

```
SYS@shocdb1 31-AUG-22>
SYS@shocdb1 31-AUG-22>alter pluggable database SHOPDB1 close immediate;

Pluggable database altered.

SYS@shocdb1 31-AUG-22>
```

STEP 2 - Open the pluggable database PDB to READ ONLY.

alter pluggable database SHOPDB1 open read only;

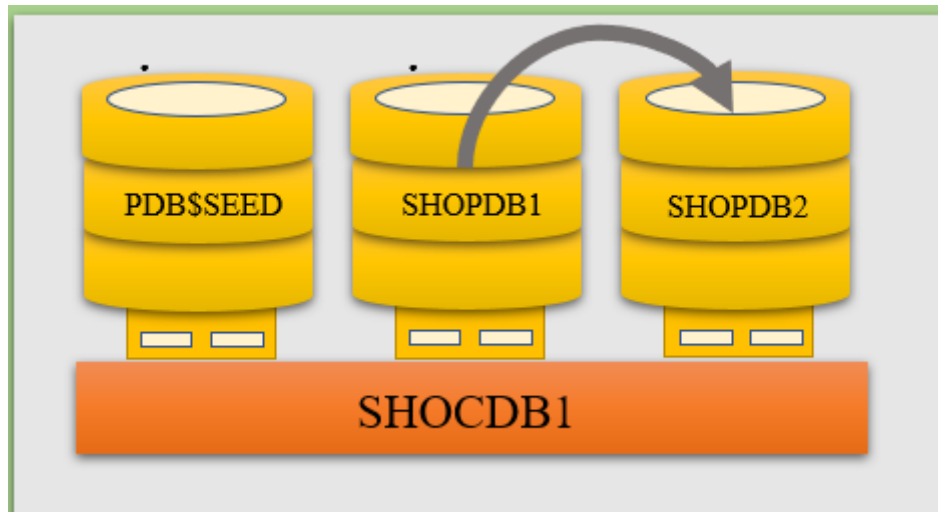
```
SYS@shocdb1 31-AUG-22>alter pluggable database SHOPDB1 open read only;

Pluggable database altered.

SYS@shocdb1 31-AUG-22>show pdbs

  CON_ID  CON_NAME          OPEN MODE  RESTRICTED
-----  -
         2  PDB$SEED         READ ONLY  NO
         3  SHOPDB1          READ ONLY  NO
SYS@shocdb1 31-AUG-22>
SYS@shocdb1 31-AUG-22>
```

Clone pluggable database PDB (SHOPDB2) from another pluggable database PDB (SHOPDB1) using parameter FILE_NAME_CONVERT.



create pluggable database SHOPDB2 from SHOPDB1 storage unlimited tempfile reuse file_name_convert=('E79387E71AAD6720E0536538A8C001D5/datafile/o1_mf_', 'SHOPDB2/') parallel 8;

```
SYS@shocdb1 31-AUG-22>create pluggable database SHOPDB2 from SHOPDB1 storage unlimited
  2 tempfile reuse file_name_convert=('E79387E71AAD6720E0536538A8C001D5/datafile/o1_mf_', 'SHOPDB2/') parallel 8;
Pluggable database created.
SYS@shocdb1 31-AUG-22>
```

New pluggable database (SHOPDB2) is created and currently in mount mode

```
SYS@shocdb1 31-AUG-22>show pdbs

  CON_ID  CON_NAME          OPEN MODE  RESTRICTED
-----  -
      2  PDB$SEED         READ ONLY  NO
      3  SHOPDB1          READ WRITE NO
      4  SHOPDB2          MOUNTED
```

Open the pluggable database (SHOPDB2) to READ WRITE.

alter pluggable database SHOPDB1 open;

```

SYS@shocdb1 31-AUG-22>
SYS@shocdb1 31-AUG-22>alter pluggable database SHOPDB2 open;

Pluggable database altered.

SYS@shocdb1 31-AUG-22>show pdbs

  CON_ID CON_NAME                                OPEN MODE  RESTRICTED
-----
      2  PDB$SEED                                READ ONLY  NO
      3  SHOPDB1                                READ WRITE NO
      4  SHOPDB2                                READ WRITE NO
SYS@shocdb1 31-AUG-22>

```

The location of the Datafiles

The container database: SHOCDB1 → /u01/app/oracle/oradata/SHOCDB1/datafile/

The Pluggable database: SHOPDB1 →
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/

The Pluggable database: SHOPDB2 → /u01/app/oracle/oradata/SHOCDB1/SHOPDB2/

Note: here we can see that the SHOPDB2 datafile's naming convention is not using OMF (oracle managed files) and even the GUIDs are not there)

```

SYS@shocdb1 31-AUG-22>col name for a120
SYS@shocdb1 31-AUG-22>select name from v$datafile;

NAME
-----
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_system_kjzyld81_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_sysaux_kjzy2j5z_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_undotbs1_kjzy3999_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_system_kjzy5sh2_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_sysaux_kjzy5scj_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_users_kjzy3bdx_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_undotbs1_kjzy5sh8_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_system_kjzyhq30_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_sysaux_kjzyhq38_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_undotbs1_kjzyhq39_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_users_kjzyjlcw_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB2/system_kjzyhq30_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB2/sysaux_kjzyhq38_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB2/undotbs1_kjzyhq39_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB2/users_kjzyjlcw_.dbf

15 rows selected.

```

The location of the Tempfiles

```
SYS@shocdb1 31-AUG-22>select name from v$tempfile;

NAME
-----
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_temp_kjzy5koj_.tmp
/u01/app/oracle/oradata/SHOCDB1/datafile/temp012022-08-31_20-12-20-238-PM.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_temp_kjzyhq3b_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB2/temp_kjzyhq3b_.dbf

SYS@shocdb1 31-AUG-22>
```

During pluggable database cloning we are using the FILE_NAME_CONVERT parameter. The output may fail with ORA-65005: missing or invalid file name pattern for file.

```
SYS@shocdb1 31-AUG-22>
SYS@shocdb1 31-AUG-22>create pluggable database SHOPDB2 from SHOPDB1 storage unlimited
  2 tempfile reuse file_name_convert=('SHOPDB1','SHOPDB2') parallel 8;
create pluggable database SHOPDB2 from SHOPDB1 storage unlimited
*
ERROR at line 1:
ORA-65005: missing or invalid file name pattern for file -
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_
system_kjzyhq30_.dbf

SYS@shocdb1 31-AUG-22>
```

Solution1:

We can change the directory in DB CREATE_FILE_DEST parameter.

```
SYS@shocdb1 31-AUG-22>show parameter db_create_file

NAME                                 TYPE          VALUE
-----
db_create_file_dest                  string        /u01/app/oracle/oradata
SYS@shocdb1 31-AUG-22>
```

Solution2:

The datafiles using the ORACLE MANAGED FILES (OMF) naming convention start with o1_mf %.

So, we manually type in the path to get rid of OMF prefix in the filenames.

```
SYS@shocdb1 31-AUG-22>create pluggable database SHOPDB2 from SHOPDB1 storage unlimited
  2 tempfile reuse file_name_convert=('E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_', 'SHOPDB2/') parallel 8;

Pluggable database created.

SYS@shocdb1 31-AUG-22>
```

In below example we clone pluggable database PDB (SHOPDB3) from another pluggable database PDB (SHOPDB2) with FILE_NAME_CONVERT. Where the datafiles are not using OMF

create pluggable database SHOPDB3 from SHOPDB2 storage unlimited tempfile reuse file_name_convert=('SHOPDB2','SHOPDB3') parallel 8;

```
SYS@shocdb1 31-AUG-22>create pluggable database SHOPDB3 from SHOPDB2 storage unlimited
  2 tempfile reuse file_name_convert=('SHOPDB2','SHOPDB3') parallel 8;

Pluggable database created.

SYS@shocdb1 31-AUG-22>
```

The location of the datafiles for

The Pluggable database: SHOPDB2 → /u01/app/oracle/oradata/SHOCDB1/SHOPDB2/

The Pluggable database: SHOPDB3 → /u01/app/oracle/oradata/SHOCDB1/SHOPDB3/

Note: here we can see that the pluggable databases SHOPDB2 and SHOPDB3 datafile's naming convention is not using OMF (oracle managed files) and even the GUIDs is not there)

```
SYS@shocdb1 31-AUG-22>select name from v$datafile;

NAME
-----
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_system_kjzyld81_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_sysaux_kjzy2j5z_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_undotbs1_kjzy3999_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_system_kjzy5sh2_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_sysaux_kjzy5scj_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_users_kjzy3bdx_.dbf
/u01/app/oracle/oradata/SHOCDB1/datafile/ol_mf_undotbs1_kjzy5sh8_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_system_kjzyhq30_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_sysaux_kjzyhq38_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_undotbs1_kjzyhq39_.dbf
/u01/app/oracle/oradata/SHOCDB1/E79387E71AAD6720E0536538A8C001D5/datafile/ol_mf_users_kjzyjlcx_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB2/system_kjzyhq30_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB2/sysaux_kjzyhq38_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB2/undotbs1_kjzyhq39_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB2/users_kjzyjlcx_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB3/system_kjzyhq30_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB3/sysaux_kjzyhq38_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB3/undotbs1_kjzyhq39_.dbf
/u01/app/oracle/oradata/SHOCDB1/SHOPDB3/users_kjzyjlcx_.dbf

19 rows selected.

SYS@shocdb1 31-AUG-22>
```


So now Finally when we login into the container database i.e., SHOCDB1

We can see that it has below PDBS

The pluggable database: PDB\$SEED, SHOPDB1, SHOPDB2

```
[oracle@SHOMAC1 ~]$ . oraenv
ORACLE_SID = [shocdb1] ? shocdb1
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@SHOMAC1 ~]$
[oracle@SHOMAC1 ~]$ sqlplus "/" as sysdba"

SQL*Plus: Release 12.2.0.1.0 Production on Fri Sep 2 16:18:15 2022

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SYS@shocdb1 02-SEP-22>show pdbs

   CON_ID CON_NAME                                OPEN MODE  RESTRICTED
-----
   2 PDB$SEED                                READ ONLY  NO
   3 SHOPDB1                                READ WRITE NO
   4 SHOPDB2                                READ WRITE NO

SYS@shocdb1 02-SEP-22>
SYS@shocdb1 02-SEP-22>
```

Step to drop pluggable database PDB

alter pluggable database SHOPDB2 close;

drop pluggable database SHOPDB2 including datafiles;

```
SYS@shocdb1 31-AUG-22>alter pluggable database SHOPDB2 close ;

Pluggable database altered.

SYS@shocdb1 31-AUG-22>drop pluggable database SHOPDB2 including datafiles;

Pluggable database dropped.

SYS@shocdb1 31-AUG-22>
```