

Creación de un Multitenant Container Database 12c de forma Manual

Por Francisco Riccio 

Introducción

En algunas ocasiones hemos tenido la necesidad de crear bases de datos de forma manual, quizás por automatizar el proceso de creación o porque las políticas de seguridad de la compañía no permitían la ejecución de aplicaciones en entornos visuales y por ende no podíamos ejecutar el utilitario DBCA, etc.

El propósito de esta implementación es indicar el procedimiento a ejecutar para crear un Multitenant Container Database (CDB) de forma manual mediante scripts. Dicho procedimiento se asemeja en varios pasos a la creación de una base de datos como lo hacíamos en las anteriores versiones salvo por algunos pasos adicionales que debemos realizar.

Antes de iniciar con la implementación, recomiendo revisar el siguiente material con la finalidad de tener todos los conceptos sólidos sobre la arquitectura de Oracle Multitenant.

<https://docs.oracle.com/database/121/CNCPT/cdblogic.htm>

Implementación

El ambiente a implementar está configurado sobre 1 servidor con sistema operativo Oracle Linux 5 Update 10 x64 bits y Oracle Database 12.1.0.2.

El CDB a crear se llamará DEV.

Paso 1

Creamos una carpeta donde se almacenarán los control files, data files y redo logs de nuestro CDB. Posteriormente configuramos la variable ORACLE_SID y crearemos el archivo PFILE.

```
$ mkdir -p /u02/oradata/DEV
$ export ORACLE_SID=DEV
$ cd $ORACLE_HOME/dbs
```

A continuación se presenta el contenido del archivo initDEV.ora (PFILE).

```
*.db_name=DEV  
*.control_files='/u02/oradata/DEV/control01.ctl','/u02/oradata/DEV/control02.ctl','/u02/oradata/DEV/control03.ctl'  
*.db_block_size=8192  
*.memory_target=800M  
*.undo_tablespace='UNDOTBS1'  
*.enable_pluggable_database=true
```

El parámetro **ENABLE_PLUGGABLE_DATABASE** es importante que esté configurado con el valor de TRUE, de esta manera indicamos que es una base de datos de tipo CDB.

El valor por defecto del parámetro es FALSE.

Nota: Si contamos con una base de datos CDB y modificamos el parámetro ENABLE_PLUGGABLE_DATABASE al valor de FALSE o dicho parámetro lo retiramos del Parameter File conseguiremos el error: "ORA-65093: multitenant container database not set up properly".

Por ende la base de datos CDB debe tener este parámetro con el valor de TRUE en todo momento.

Una vez creado el PFILE procedemos a subir nuestro CDB a NOMOUNT.

```
$ sqlplus / as sysdba
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Sun May 24 22:37:07 2015
```

```
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
Connected to an idle instance.
```

```
SQL> startup nomount  
ORACLE instance started.
```

```
Total System Global Area 272629760 bytes  
Fixed Size 2923336 bytes  
Variable Size 213910712 bytes  
Database Buffers 50331648 bytes  
Redo Buffers 5464064 bytes  
SQL> █
```

Paso 2

Creamos una carpeta para almacenar el Container SEED. En nuestro caso se ubicará en la ruta: /u02/oradata/DEV/pdbseed.

```
$ mkdir /u02/oradata/DEV/pdbseed
```

Posteriormente procedemos a crear el CDB.
Se adjunta el script de creación.

```
create database DEV
maxlogfiles 16
maxlogmembers 3
maxdatafiles 1024
datafile '/u02/oradata/DEV/system01.dbf' size 700M reuse
        autoextend on next 200M maxsize unlimited
        extent management local
SYSAUX
datafile '/u02/oradata/DEV/sysaux01.dbf' size 500M reuse
        autoextend on next 200M maxsize unlimited
smallfile default temporary tablespace TEMP
tempfile '/u02/oradata/DEV/temp01.dbf' size 500M reuse
        autoextend on next 200M maxsize 5G
smallfile undo tablespace UNDOTBS1
datafile '/u02/oradata/DEV/undotbs01.dbf' size 500M reuse
        autoextend on next 200M maxsize 5G
CHARACTER SET AL32UTF8
NATIONAL CHARACTER SET AL16UTF16
LOGFILE
group 1 ('/u02/oradata/DEV/redo01.log') SIZE 100M,
```

group 2 ('/u02/oradata/DEV/redo02.log') SIZE 100M,

group 3 ('/u02/oradata/DEV/redo03.log') SIZE 100M

USER SYS IDENTIFIED BY oracle USER SYSTEM IDENTIFIED BY oracle

ENABLE PLUGGABLE DATABASE

SEED

file_name_convert=('/u02/oradata/DEV/system01.dbf', '/u02/oradata/DEV/pdbseed/system01.dbf',

'/u02/oradata/DEV/sysaux01.dbf', '/u02/oradata/DEV/pdbseed/sysaux01.dbf',

'/u02/oradata/DEV/temp01.dbf', '/u02/oradata/DEV/pdbseed/temp01.dbf',

'/u02/oradata/DEV/undotbs01.dbf', '/u02/oradata/DEV/pdbseed/undotbs01.dbf');

```
SQL> create database DEV
 2  maxlogfiles 16
 3  maxlogmembers 3
 4  maxdatafiles 1024
 5  datafile '/u02/oradata/DEV/system01.dbf' size 700M reuse
 6      autoextend on next 200M maxsize unlimited
 7      extent management local
 8  SYSAUX
 9  datafile '/u02/oradata/DEV/sysaux01.dbf' size 500M reuse
10      autoextend on next 200M maxsize unlimited
11  smallfile default temporary tablespace TEMP
12  tempfile '/u02/oradata/DEV/temp01.dbf' size 500M reuse
13      autoextend on next 200M maxsize 5G
14  smallfile undo tablespace UNDOTBS1
15  datafile '/u02/oradata/DEV/undotbs01.dbf' size 500M reuse
16      autoextend on next 200M maxsize 5G
17  CHARACTER SET AL32UTF8
18  NATIONAL CHARACTER SET AL16UTF16
19  LOGFILE
20  group 1 ('/u02/oradata/DEV/redo01.log') SIZE 100M,
21  group 2 ('/u02/oradata/DEV/redo02.log') SIZE 100M,
22  group 3 ('/u02/oradata/DEV/redo03.log') SIZE 100M
23  USER SYS IDENTIFIED BY oracle USER SYSTEM IDENTIFIED BY oracle
24  ENABLE PLUGGABLE DATABASE
25  SEED file_name_convert=('/u02/oradata/DEV/system01.dbf', '/u02/oradata/DEV/pdbseed/system01.dbf',
26                          '/u02/oradata/DEV/sysaux01.dbf', '/u02/oradata/DEV/pdbseed/sysaux01.dbf',
27                          '/u02/oradata/DEV/temp01.dbf', '/u02/oradata/DEV/pdbseed/temp01.dbf',
28                          '/u02/oradata/DEV/undotbs01.dbf', '/u02/oradata/DEV/pdbseed/undotbs01.dbf');
```

Database created.

Las opciones:

- **ENABLE PLUGGABLE DATABASE**, permite crear el container ROOT en conjunto con los CONTROL FILES, REDO LOG Y DATAFILES durante la fase de OPEN.
- **SEED**, permite indicar donde se almacenarán los data files para el container SEED. Si utilizamos OMF o el parámetro PDB_FILE_NAME_CONVERT podemos ignorar esta opción.

Paso 3

Finalizado el paso 2 debemos modificar el parámetro oculto: "_oracle_script" al valor de TRUE a nivel de sesión.

```
SQL> select open_mode from v$database;
```

```
OPEN_MODE  
-----  
READ WRITE
```

```
SQL> alter session set "_ORACLE_SCRIPT"=TRUE;
```

```
Session altered.
```

Luego debemos cerrar y abrir el PDB SEED.

```
SQL> alter pluggable database pdb$seed close;
```

```
Pluggable database altered.
```

```
SQL> alter pluggable database pdb$seed open read only;
```

```
Pluggable database altered.
```

Procedemos a ejecutar el siguiente script:

```
export PATH=$ORACLE_HOME/bin:$ORACLE_HOME/perl/bin/:$PATH
```

```
SQL> @?/rdbms/admin/catcdb.sql
```

El script permite instalar todos los componentes (APEX, Oracle OLAP, Oracle Database Vault, Oracle Label Security, etc) disponibles en el CDB.

El CDB debería contar con todos los componentes instalados para evitar el error ORA-1917.

Mayor detalle en: My Oracle Support (MOS) Nota: 1962134.1 (ORA-1917: user or role 'PDB_DBA' does not exist When Trying to Create Pluggable Database (PDB))

Nota 1: El BUG 17033183 apareció en la primera versión del Release 12c; en el cual el script catcdb.sql no venía incluido como parte de la instalación del software de Oracle Database.

Mayor detalle en: My Oracle Support (MOS) Nota: 17033183.8 (Bug 17033183 - Withdrawn fix - superseded by 17316776).

Nota 2: La redefinición de la variable PATH está documentado en: My Oracle Support (MOS) Nota: 1964089.1 (ORA-01917 While Creating Pluggable Database).

Una vez ejecutado el script, nuestro CDB estaría listo para albergar PDBs como se verá a continuación.

Paso 4

Es importante que tengamos presente las siguientes recomendaciones:

- El tablespace permanente y temporal del CDB deben ser de uso exclusivo.
- Asignar un nuevo tablespace permanente y temporal por defecto a cada PDB de uso exclusivo.

A continuación se creará un PDB de ejemplo con la finalidad de cumplir con las recomendaciones previamente indicadas.

```
create pluggable database PDB_DESA
admin user friccio identified by oracle roles=(DBA)
file_name_convert=('/u02/oradata/DEV/pdbseed','/u02/oradata/DEV/PDB_DESA');

SQL> create pluggable database PDB_DESA
admin user friccio identified by oracle roles=(DBA)
file_name_convert=('/u02/oradata/DEV/pdbseed','/u02/oradata/DEV/PDB_DESA');

Pluggable database created.
```

Creamos los tablespace permanente y temporal para configurarlos como defectos del PDB.

```
SQL> alter session set container=PDB_DESA;

Session altered.

SQL> create tablespace DATA datafile '/u02/oradata/DEV/PDB_DESA/DATA_01.dbf' size 100M;

Tablespace created.

SQL> create temporary tablespace TEMP_DESA tempfile '/u02/oradata/DEV/PDB_DESA/TEMP_DESA_01.dbf' size 100M;

Tablespace created.

SQL> alter database default tablespace DATA;

Database altered.

SQL> alter pluggable database PDB_DESA default tablespace DATA;

Pluggable database altered.

SQL> alter pluggable database PDB_DESA default temporary tablespace TEMP_DESA;

Pluggable database altered.
```

Conclusión

Este material ha permitido entregar todos los pasos necesarios para crear un Multitenant Container Database (CDB) desde la consola SQL Plus vía scripts demostrando lo simple y práctico del proceso.

A pesar de ello, es importante aclarar que el uso del utilitario DBCA para la creación de CDB y NO-CDB debería ser nuestra primera opción para evitar errores humanos en el procedimiento.

Publicado por Ing. Francisco Riccio. Es un IT Architect en IBM Perú e instructor de cursos oficiales de certificación Oracle. Está reconocido por Oracle como un Oracle ACE y certificado en productos de Oracle Application & Base de Datos.

e-mail: francisco@friccio.com

web: www.friccio.com