Workshop Multitenant, Multimodel, In-Memory para la base de Datos Oracle

Parte 1 de 3

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential

Contenidos

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential

Requerimientos iniciales

- Clave SSH privada para acceder a la maquina cloud que contiene las bases de datos. Esta clave se proporciona junto con la documentación necesaria para el workshop.
- Cliente SSH para poder acceder al host de base de datos en el que se ejecutara el workshop
- IP de la maquina

Multitenant (1 hora)

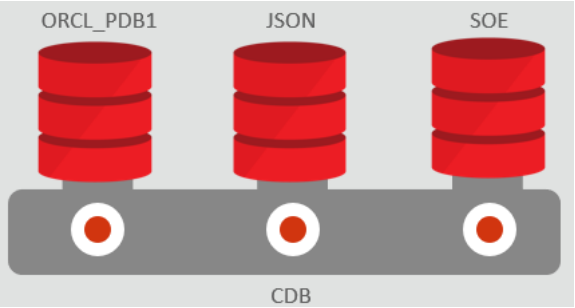
Acceso al contenedor, creación y administración de PDBs (15min)

A continuación, se describen los primeros pasos para acceder al contenedor de bases de datos y la creación de una Pluggable Database.

Accediendo al backend mediante ssh, se accede con usuario oracle para ejecutar el comando sqlplus.

```
[oracle@myoracledb ~]$
[oracle@myoracledb ~]$
[oracle@myoracledb ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Tue Dec 31 13:11:41 2019
Version 19.5.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c EE Extreme Perf Release 19.0.0.0.0 - Production
Version 19.5.0.0.0

--- Una vez hecho esto, podemos ver las PDBs disponibles.

SQL> show pdbs

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCL_PDB1                      READ WRITE NO
         4 JSON                           READ WRITE NO
         5 SOE                            READ WRITE NO
SQL>
```

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential



Crear una nueva PDB en un contenedor de bases de datos (CDB)

A continuación, se muestran los pasos para crear una nueva PDB.

En primer lugar, desde el host Linux que contiene la base de datos, se conecta a través de "sqlplus / as sysdba". Una vez hecho esto estaríamos en el Contenedor de bases de datos (CDB). Desde aquí podemos crear Pluggable Databases (PDB), que serían las bases de datos al uso.

```
$ sqlplus / as sysdba

create pluggable database PDB1 admin user pdbadmin identified by
WddFsdf_12_we2;
show pdbs;
alter pluggable database PDB1 open read write;
show pdbs;
alter session set container = PDB1;
ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY WddFsdf_12_we2
WITH BACKUP;
```

Desacoplar una PDB del contenedor de bases de datos

Las PDBs se pueden acoplar o desacoplar entre CDBs. Por ejemplo, se puede desconectar la PDB1 de una CDB1, y conectar la PDB1 a otra CDB2.
En el siguiente ejemplo se desacopla una PDB de un Contenedor.
Para ejecutar estas operativas, hay que tener en cuenta que las bases de datos Oracle en el cloud tienen por defecto activado *Transparent Data Encryption*. Teniendo en cuenta esto, ejecutaremos algunos comandos para ejecutar todas las acciones sin perder ese cifrado.

```
--- En primer lugar, hay que conectar al CDB. El keystore que contiene las
claves de cifrado normalmente se encuentra en autologin, no se pueden exportar
claves fuera de la BD si el estado del keystore está en autologin.

conn / as sysdba

--- una vez cerrado el keystore a nivel de CDB, se cerrará también en todas las
PDBs
```

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential

```
administer key management set keystore close;


---- abrir la keystore con password en lugar de autologin

administer key management set keystore open identified by "WddFsdf_12_we2";


show pdbs;

alter session set container=PDB1;

--- Abrimos el keystore en la PDB
administer key management set keystore open identified by "WddFsdf_12_we2";

--- Ya se puede exportar la clave de cifrado
administer key management export encryption keys with secret "WddFsdf_12_we2"
to '/home/oracle/PDB1.p12' identified by "WddFsdf_12_we2";

--- Una vez exportada la clave de cifrado, se desacopla la PDB de la CDB.

conn / as sysdba

alter pluggable database PDB1 close immediate;
alter pluggable database PDB1 unplug into '/home/oracle/PDB1.xml';

drop pluggable database PDB1 keep datafiles;

show pdbs;
```

Comprobar compatibilidad de una PDB para poder ser conectada a un contenedor

Una vez una PDB esta desconectada, es una buena práctica, si se quiere conectar a una CDB, comprobar la compatibilidad de la PDB dentro de la CDB de destino. En el caso de haber alguna incompatibilidad, este bloque de código nos informaría de ello.

```
set serveroutput on

DECLARE
   compatible BOOLEAN := FALSE;
BEGIN
   compatible := DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
        pdb_descr_file => '/home/oracle/PDB1.xml');
   if compatible then
      DBMS_OUTPUT.PUT_LINE('Is pluggable PDB1.xml compatible? YES');
   else DBMS_OUTPUT.PUT_LINE('Is pluggable PDB1.xml compatible? NO');
   end if;
END;
/
```

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential

Clonados, Resguardo y Recuperación de PDBs (15 min)

Conectar una PDB por medio de un clon desde una PDB desconectada (método as clone)

En el siguiente ejemplo se muestra cómo crear un clon partiendo de los metadatos y los ficheros de datos de la PDB desconectada y también se vuelve a conectar la PDB inicial por el método “nocopy”, es decir, tomando los metadatos y ficheros de datos originales de la PDB y volviéndola a conectar a la CDB.

```
conn / as sysdba

create pluggable database PDB1_clone as clone using '/home/oracle/PDB1.xml';

## Plug the unplugged DB with nocopy method
create pluggable database PDB1_nocopy using '/home/oracle/PDB1.xml' NOCOPY
TEMPFILE REUSE;

alter pluggable database PDB1_nocopy open read write;

--- Aquí, debería dar un Warning, falta por importar las claves de cifrado que
exportamos en el ejercicio anterior. Con la siguiente sentencia se comprueban
los errores de acoplamiento de la PDB:

select name,cause,type,status,message,action from pdb_plug_in_violations;


show pdbs

alter session set container = PDB1_nocopy;

select file_name from dba_data_files; --- con esta query vemos los DBFs

--- Tomar nota de los datafiles de esta query, luego se utilizarán para simular
una pérdida de datos y recuperación mediante un backup de PDB.

Ejemplo:
SQL> select file_name from dba_data_files
  2  ;

FILE_NAME
--------------------------------------------------------------------------------
-
+DATA/ORCL_FRA3TF/9BC7919806F1646CE0530214010AC0F4/DATAFILE/system.299.1029320935

+DATA/ORCL_FRA3TF/9BC7919806F1646CE0530214010AC0F4/DATAFILE/sysaux.300.1029320935

+DATA/ORCL_FRA3TF/9BC7919806F1646CE0530214010AC0F4/DATAFILE/undotbs1.298.1029320935
```

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential

```
--- Se abre el keystore y se importan las claves de cifrado
administer key management set keystore open identified by "WddFsdf_12_we2";

ADMINISTER KEY MANAGEMENT IMPORT  KEYS WITH SECRET WddFsdf_12_we2 FROM
'/home/oracle/PDB1.p12' IDENTIFIED by WddFsdf_12_we2 with backup;

--- Reiniciar PDB, ya deberia abrir sin errores relativos al cifrado.

conn / as sysdba

alter pluggable database PDB1_nocopy close immediate;

alter pluggable database PDB1_nocopy open read write;

show pdbs
```

Crear una nueva PDB tomando como referencia otra PDB.

En el siguiente ejercicio, se crea un clon de una PDB por el metodo “from PDB”, el cual toma como referencia la PDB indicada después del from y crea una copia exacta.

```
create pluggable database PDB1 from PDB1_nocopy keystore identified by
WddFsdf_12_we2;

alter pluggable database PDB1 open read write;

show pdbs;
```

Backup y recuperacion PDBs

A continuación, se simula la pérdida de datos de un datafile por procedimientos externos a la base de datos. El ejercicio consiste en hacer un backup de una PDB usando RMAN (Recovery Manager), borrar un datafile desde ASM y recuperar la PDB con el datafile perdido.

```
$ rman target=/

RMAN> BACKUP PLUGGABLE DATABASE PDB1_NOCOPY;
```

Simulación de pérdida de datos:

```
--- Entrar en la PDB en la que queremos simular la perdida de datos

$ sqlplus / as sysdba
show pdbs

alter session set container=PDB1_NOCOPY;
```

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential

```
select file_name from dba_data_files;

--- cerrar PDB

alter pluggable database PDB1_NOCOPY close immediate;

--- Salir de sqlplus

--- Comprobar el usuario en el que nos encontramos, si es Oracle, salimos al
usuario opc para después conectar al usuario grid

[oracle@myoracledb ~]$ id
uid=101(oracle) gid=1001(oinstall)
groups=1001(oinstall),1002(dbaoper),1003(dba),1006(asmdba)
[oracle@myoracledb ~]$ sudo -u grid bash
[grid@myoracledb oracle]$

--- Una vez conectado al usuario grid, se entra en ASMCMD para borrar un
datafile (ejemplo de pérdida de datos)

[grid@myoracledb ~]$ asmcmd
ASMCMD> ls
DATA/
RECO/
ASMCMD> rm
+DATA/ORCL_FRA3TF/9BC7919806F1646CE0530214010AC0F4/DATAFILE/system.299.1029320935
ASMCMD> exit

--- Se vuelve a cambiar el usuario a Oracle y se intenta abrir la PDB

[grid@myoracledb ~]$   exit
logout
[oracle@myoracledb ~]$

--- Se intenta abrir la PDB en la que hemos simulado la perdida de datos

sqlplus / as sysdba

SQL> alter pluggable database PDB1_NOCOPY open;
alter pluggable database PDB1_NOCOPY open
*
ERROR at line 1:
ORA-01157: cannot identify/lock data file 56 - see DBWR trace file
ORA-01110: data file 56:
'+DATA/ORCL_FRA3TF/9BC7919806F1646CE0530214010AC0F4/DATAFILE/system.299.1029320
9
35'

--- Vemos que falla, hay que recuperar desde backup
```

Una vez hecho el backup, se ejecuta el siguiente código en RMAN para recuperar la BD.

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential

```
$ rman target=/

RUN {
  RESTORE PLUGGABLE DATABASE PDB1_NOCOPY;
  RECOVER PLUGGABLE DATABASE PDB1_NOCOPY;
  ALTER PLUGGABLE DATABASE PDB1_NOCOPY open;
}
```

Limpiar el entorno

```
sqlplus / as sysdba

alter pluggable database PDB1_NOCOPY close immediate;
drop pluggable database PDB1_NOCOPY including datafiles;
alter pluggable database PDB1_clone close immediate;
drop pluggable database PDB1_clone including datafiles;
```

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential