Practice

Using Oracle Data Pump

Practice Target

In this practice, you will implement the common tasks included in using Oracle Data Pump utilities.

Practice Overview

In this practice, you will perform the following tasks:

- Export `SOE` schema from `PDB1` that is running in `srv1` to the database that is running in `winsrv`
- Use Network option to directly move `SOE` schema from `srv1` to `winsrv`.

Assumptions

- This practice assumes that `srv1` is up and running from the **CDB** snapshot.
- The vm `winsrv` is up and running

A. Using Data Pump Utilities to Move SOE Schema from srv1 to winsrv using Dump Files

In this section of the practice, you will use Data Pump to move SOE Schema from srv1 to winsrv. Then you will connect Swingbench to the database running in winsrv.

1. Open Putty and connect to srv1 as oracle

2. In srv1, invoke SQL*Plus and connect to orawindb as HR to test the connection to the database in winsrv

```
sqlplus hr/ABcd##1234@orawindb
```

As a preparation for the migration, we would like to know the total size of SOE schema. We need to know this information so that we make sure that there is enough space in the target database system.

3. Submit the following query to calculate the total size of the objects owned by SOE

   DBA_SEGMENTS provides information about all the objects that consume data from the disk in the database.

```
conn / as sysdba
ALTER SESSION SET CONTAINER=PDB1;
SELECT SUM(BYTES/1024/1024) MB FROM DBA_SEGMENTS WHERE OWNER='SOE';
```

In the target database, we would like to create a tablespace dedicated to SOE schema. For performance reasons, make the initial tablespace size slightly more than the total SOE schema size.

4. Connect to the database in winsrv as SYSTEM.

```
conn system/ABcd##1234@orawindb
```

5. Make sure the OMF is enabled in the database.

   If OMF is enabled, there is no need to set the datafile in the CREATE TABLESPACE statement.

```
show parameter DB_CREATE_FILE_DEST
```

6. Create a tablespace to accommodate the SOE schema in winsrv. Make sure its size is slightly larger than the total SOE data size.

   For easy configuration, we could create a tablespace with the same name as the one in the source database. We create a tablespace with a different name to practice tablespace remapping.

```
CREATE TABLESPACE WINSOE DATAFILE SIZE 1G AUTOEXTEND ON NEXT 100M MAXSIZE 30G;
```

After preparing the tablespace, we need to create the destination database user. Normally, we need the destination user to have the same privileges as the source database user. In the following steps, you will generate the code to re-create SOE in the destination database and grant the required privileges to it.

7. Run the following statements in srv1 to obtain the code to re-create SOE.

DBMS_METADATA is a built-in package that we can use to generate DDL to re-create any object in the database.

```
conn SYSTEM/ABcd##1234@pdb1

set linesize 180
set head off
set pages 0
set long 9999999
SELECT DBMS_METADATA.GET_DDL('USER', 'SOE') || ';' DDL FROM DUAL;
SELECT DBMS_METADATA.GET_GRANTED_DDL('ROLE_GRANT','SOE') || ';' FROM DUAL;
SELECT DBMS_METADATA.GET_GRANTED_DDL('SYSTEM_GRANT','SOE') || ';' FROM DUAL;
SELECT DBMS_METADATA.GET_GRANTED_DDL('OBJECT_GRANT','SOE') || ';' FROM DUAL;
set head on
set pages 15
```

8. Copy the code generated by the previous step. Perform the following actions on it:

a. Paste it into a new text file.

b. Remove the extra characters in it.

c. Change the default tablespace name to WINSOE, which is the name of the tablespace that we created in the destination database.

d. Add the semi colon ";" to the end of the statements.

e. Regarding the "CREATE USER" statement, the code creates the user with the same password as the original SOE user. If that is fine for your case, you just need to remove the carriage return (which is equivalent to the [Enter] character) from the generated characters that represent the password (so that the characters after the VALUES clause becomes in a single line). If you do not remove those characters, the code will generate the error "ORA-02153: invalid VALUES password string".

Alternatively, you can remove the "VALUES" and replace it with the password.

The code should look like the following code:

```
conn system/ABcd##1234@orawindb

CREATE USER "SOE" IDENTIFIED BY VALUES
'S:913D67ECFC81951C215EB0CD80B6C0E3EBFBCB0EF3AABAED958BEB68BA01;T:6D8243E092475373DD9FCB1
DCA68E44EDF937F38F8D40F19ED631B6327FE6CF9B38CFE87A64345A5E1C1363EB4AFFE6A33FAE407578B9CCE
F90F436808A96E90F0FB9F5282642B155DCAAAB25293713E'
DEFAULT TABLESPACE "WINSOE"
 TEMPORARY TABLESPACE "TEMP";

GRANT "RESOURCE" TO "SOE";
GRANT ALTER SYSTEM TO "SOE";
GRANT CREATE SESSION TO "SOE";
GRANT ALTER SESSION TO "SOE";
GRANT CREATE TABLE TO "SOE";
GRANT CREATE VIEW TO "SOE";
```

```
GRANT CREATE SEQUENCE TO "SOE";
GRANT CREATE PROCEDURE TO "SOE";
GRANT ANALYZE ANY TO "SOE";
GRANT ANALYZE ANY DICTIONARY TO "SOE";
GRANT CREATE JOB TO "SOE";
GRANT MANAGE SCHEDULER TO "SOE";
GRANT SELECT ON "SYS"."V_$PARAMETER" TO "SOE";
GRANT EXECUTE ON "SYS"."DBMS_LOCK" TO "SOE";
```

9. Login to `orawindb` as `SYS` and run the code prepared in the preceding step.

```
conn sys/ABcd##1234@orawindb as sysdba

-- run the code produced from the preceding step
```

10. Grant unlimited quota to `SOE` on the `WINSOE` tablespace.

```
ALTER USER SOE QUOTA UNLIMITED ON WINSOE ;
```

Now we need to create a directory object in the source system to save the dump file produced by the Data Pump export utility in it.

11. Get the directory path pointed by the prebuilt directory object `DATA_PUMP_DIR`. By default, Oracle Data Pump export utility produces its dump file in this directory.

The local users in the PDBs do not have privileges to access this directory object.

```
conn / as sysdba
SELECT DIRECTORY_PATH FROM DBA_DIRECTORIES WHERE DIRECTORY_NAME='DATA_PUMP_DIR';
```

12. Create a directory in the staging folder for saving the dump files in it.

```
host mkdir /media/sf_staging/dump
```

13. Create a directory object that is linked to the new directory.

Because the source database is a PDB, we must create the directory object in that PDB.

```
ALTER SESSION SET CONTAINER=PDB1;
CREATE DIRECTORY DUMPDIR AS '/media/sf_staging/dump';
```

14. Grant the required privileges to `SOE` on the created directory.

```
GRANT READ, WRITE ON DIRECTORY DUMPDIR TO SOE;
```

Before issuing the export job, we would like to know how much is the total size of the dump file that will be produced by the Data Pump Export utility. We normally do that when migrating large data.

15. Exit from SQL*Plus

```
quit
```

16. Run the Data Pump Export utility so that it estimates the total size of the migration job dump file.

The code invokes the Data Pump export utility in the default mode (`SCHEMA` mode).

```
expdp SOE/ABcd##1234@pdb1 DIRECTORY=DUMPDIR ESTIMATE_ONLY=Y ESTIMATE=STATISTICS
```

Now we are ready to invoke the export job. In real life scenarios, if the target is to migrate the system to a new server, we would disallow application sessions to the database before starting the job.

17. Run the Data Pump utility to execute the export job.

The `LOGTIME` and `METRICS` are optional but recommended.

```
expdp SOE/ABcd##1234@pdb1 DIRECTORY=DUMPDIR DUMPFILE=SOE.dmp LOGFILE=SOE.log LOGTIME=ALL
METRICS=YES
```

18. Check out the contents of the log file.

It contains copy of the command output. Observe the password is not shown in the output.

```
cat /media/sf_staging/dump/SOE.log
```

19. Check out the size of the produced dump file.

```
ls -alh /media/sf_staging/dump/SOE.dmp
```

The export job is concluded. We are ready now to import the dump file into the destination database.

20. In the hosting PC, move the produced dump file from the shared folder configured in `srv1` to the shared folder configured in `winsrv`. If they are the same, you do not have to do anything in this step.

21. Login to `winsrv` from its VirtualBiox window and open a command-prompt window.

We need to login to the GUI interface of `winsrv` because, for some reason, creating a directory object into a Windows-based machine from a Putty session does not work!

22. Create a local temporary folder and copy the `SOE.dmp` file to it.

In Windows-based vms, the shared folder is normally mapped to Z drive. Open the File Explorer in `winsrv` to verify that the Z drive is correctly mapped to the shared folder.

Oracle database engine in Windows cannot access files in directory objects pointed to a network shared folder.

```
mkdir D:\temp
copy Z:\SOE.dmp D:\temp
```

23. Login to `orawindb` as `sys` and create a directory object that points to the dump subfolder in the shared folder. Grant access on the directory to `SOE`.

```
sqlplus sys/ABcd##1234@orawindb as sysdba
CREATE OR REPLACE DIRECTORY DUMPDIR AS 'D:\temp';
GRANT READ, WRITE ON DIRECTORY DUMPDIR TO SOE;
```

24. Exit from SQL*Plus

```
exit
```

25. Run the Data Pump import utility to import the `SOE.dmp` file into the destination schema.

Observe the remapping parameter in the command.

```
impdp SOE/ABcd##1234@orawindb DIRECTORY=DUMPDIR DUMPFILE=SOE.dmp LOGFILE=SOEimport.log
REMAP_TABLESPACE=SOETBS:WINSOE LOGTIME=ALL METRICS=YES
```

The utility returns the following error:

```
ORA-39082: Object type PACKAGE BODY:"SOE"."ORDERENTRY" created with compilation
warnings
```

This is a common error returned by import utility when importing PL/SQL packages. To resolve this issue, we simply need to compile the packages.

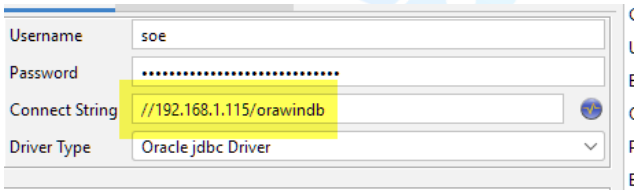
26. Login to `orawindb` and compile the invalid package.

If you have so many invalid PL/SQL packages, you can compile them all using the script `@$ORACLE_HOME/rdbms/admin/utlrp.sql`

```
sqlplus SOE/ABcd##1234@orawindb
ALTER PACKAGE "SOE"."ORDERENTRY" COMPILE;
```

27. In the hosting PC, open a command-prompt window and start Swingbench.

28. Change the connection string to connect to `orawindb` as follows. Test the connection. It should be successful.



29. Start the Swingbench sessions.

The sessions should operate as normal. By this step, the migration is concluded.

30. Stop Swingbench sessions and exit from Swingbench.

B. Using Data Pump Utilities to Move SOE Schema from srv1 to winsrv via Network Links

In this section of the practice, you will use Data Pump to move SOE schema from srv1 to winsrv using Network Links.

31. In winsrv, login as sys and re-create the SOE schema.

```
sqlplus sys/ABcd##1234@orawindb as sysdba

DROP USER SOE CASCADE;

CREATE USER "SOE" IDENTIFIED BY ABcd##1234
DEFAULT TABLESPACE "WINSOE" TEMPORARY TABLESPACE "TEMP"
QUOTA UNLIMITED ON WINSOE;

GRANT "RESOURCE" TO "SOE";
GRANT ALTER SYSTEM TO "SOE";
GRANT CREATE SESSION TO "SOE";
GRANT ALTER SESSION TO "SOE";
GRANT CREATE TABLE TO "SOE";
GRANT CREATE VIEW TO "SOE";
GRANT CREATE SEQUENCE TO "SOE";
GRANT CREATE PROCEDURE TO "SOE";
GRANT ANALYZE ANY TO "SOE";
GRANT ANALYZE ANY DICTIONARY TO "SOE";
GRANT CREATE JOB TO "SOE";
GRANT MANAGE SCHEDULER TO "SOE";
GRANT SELECT ON "SYS"."V_$PARAMETER" TO "SOE";
GRANT EXECUTE ON "SYS"."DBMS_LOCK" TO "SOE";
GRANT READ, WRITE ON DIRECTORY DUMPDIR TO SOE;
```

32. Grant CREATE DATABASE LINK privilege to SOE.

```
GRANT CREATE DATABASE LINK TO SOE;
```

33. Login as SOE and run the following code to create a database link that points to SOE in srv1.

SOEsrv1 name was added earlier in the course in
D:\oracle\product\19.0.0\db_1\network\admin\tnsnames.ora

```
conn SOE/ABcd##1234@orawindb
CREATE DATABASE LINK PDB1.LOCALDOMAIN
 CONNECT TO SOE IDENTIFIED BY ABcd##1234 USING 'soesrv1';

-- verify: the query should return the current date:
SELECT SYSDATE FROM DUAL@PDB1.LOCALDOMAIN ;
```

34. Exit from SQL*Plus

```
exit
```

35. In `winsrv`, invoke the Data Pump import utility to migrate `SOE` from `srv1` to `winsrv`.

Observe that there is no need to set a dump file but there is a need to set the `DIRECTORY`.

In this code example, we did not use a tns name to connect to the database. This means the utility will connect to the local database pointed by the environment variable `ORACLE_SID`

```
impdp SOE/ABcd##1234 NETWORK_LINK=PDB1.LOCALDOMAIN DIRECTORY=DUMPDIR LOGFILE=SOE_net.log
REMAP_TABLESPACE=SOETBS:WINSOE LOGTIME=ALL METRICS=YES
```

Cleanup

36. In `winsrv`, drop the `SOE` schema and its tablespace.

```
sqlplus SYSTEM/ABcd##1234@orawindb
DROP USER SOE CASCADE;
DROP TABLESPACE winsoe INCLUDING CONTENTS AND DATAFILES;
```

37. Delete the dump directory contents in `D:\temp` folder

38. Delete the dump directory contents in the shard folders.

39. Create vm Snapshots for `srv1` and `winsrv` and delete the old snapshots so that the changes are saved in the vms.

Summary

- Data Pump utilities can be used to reliably move data from one Oracle database to another across different platforms.
- Data movement can be done via dump files or directly via network links.