Practice 16

Performing Recovery Part VI

Practice Target

In this practice you will perform further recovery scenarios in ORADB database.

Practice Overview

In this practice, you will implement the following recovery tasks:

- Recovery from losing the password file
- Recovery from losing temporary files
- Restoring from a password-encrypted backupset
- Restoring a database to a new host

Assumptions

This practice assumes the srv1 appliance is up and running and its database ORADB is running in OPEN state.

Pre-requisites

Take a snapshot of srv1 appliance. Name the snapshot "**Practice 16 Start**".

Caution!

Do not implement the practice without creating the snapshot as instructed by the previous step.

Recovery Scenario 19:

Recovery from Loss of the Password File

Scenario Target:

- In this scenario, you will recover from losing the password file.

Scenario Assumptions:

- Database is running in OPEN mode

Simulating the Loss

1. Start Putty and connect to `srv1` as `oracle`

2. Invoke SQL*Plus and login as sysdba through the listener.

Connecting successfully to this method proves that the password file exists.

```
sqlplus sys/oracle@oradb as sysdba
```

3. Delete the password file.

```
host rm -f /u01/app/oracle/product/12.2.0/db_1/dbs/orapwORADB
```

4. Exit from SQL*Plus and login as sysdba through the listener again.

The connection should be rejected and the following error is returned:

```
ORA-01017: invalid username/password; logon denied
```

```
sqlplus sys/oracle@oradb as sysdba
```

Recovery Actions

Perform the following actions to recreate the password file.

5. Exit from SQL*Plus and issue the following command to re-create the password file.

Observe that the other credentials that used to be there in the original file are lost.

```
orapwd file=$ORACLE_HOME/dbs/orapwORADB password=oracle entries=5 format=12
```

Note: the option "`format=12`" is used to force the utility to accept the simple password. In Oracle 12.2, the utility enforced password complexity.

6. Start SQL*Plus session and login as sysdba through the listener again to test the new password file.

```
sqlplus sys/oracle@oradb as sysdba
```

Recovery Scenario 20:

Recovery from Loss of the TEMPFILES

Scenario Target:

- In this scenario, you will recover from losing TEMPFILES

Scenario Assumptions:

- Database is running in OPEN mode

Simulating the Loss

7. In the SQL*Plus session, login as sysdba using password authentication.

```
conn / as sysdba
```

8. Retrieve the tempfiles in the database. Take a note of the retrieved full file name.

```
SELECT NAME FROM V$TEMPFILE;
```

9. Delete the tempfile.

```
host rm -f /u01/app/oracle/oradata/ORADB/datafile/<temp file name>
```

Recovery Actions

Perform the following actions to recreate the temp file.

10. In the SQL*Plus session, issue the following command to create a global temporary table.

Global temporary tables are stored in temporary tablespace.

You should receive the following error:

```
ORA-27041: unable to open file
```

```
CREATE GLOBAL TEMPORARY TABLE my_temp_table ON COMMIT PRESERVE ROWS AS SELECT * FROM
DBA_TABLES ;
```

11. Execute the following command to add a TEMPFILE to the temporary tablespace.

```
ALTER TABLESPACE temp ADD TEMPFILE;
SELECT NAME FROM V$TEMPFILE;
```

12. Make the missing tempfile offline.

```
ALTER DATABASE TEMPFILE '<missing temp file>' OFFLINE
```

13. Try creating the global temporary table now.

The command should succeed.

```
CREATE GLOBAL TEMPORARY TABLE my_temp_table ON COMMIT PRESERVE ROWS AS SELECT * FROM
DBA_TABLES;
```

Note: You may not be able to drop the lost TEMPFILE at this stage. If you want to delete it, you can restart the database then issue the command "`ALTER TABLESPACE DROP TEMPFILE '...';`"

Recovery Scenario 21:

Restoring from a Password-encrypted Backupset

Scenario Target:

- In this scenario, you will restore from a password-encrypted backupset.

Scenario Assumptions:

- Database is running in OPEN mode

Scenario Preparation

14. Invoke RMAN and login as target to `ORADB`

```
rman target /
```

15. Execute the following commands to produce password-encrypted backups for the `users` tablespace.

```
SET ENCRYPTION ON IDENTIFIED BY MyPassword ONLY;
BACKUP TABLESPACE USERS TAG 'PENCRYPTED_USERS';
```

16. Execute the following command.

```
ALTER SYSTEM SWITCH LOGFILE;
```

Recovery Actions

Perform the following actions to restore the `users` tablespace from the encrypted backupset.

17. Try restoring `users` tablespace without providing the password.

The command fails and returns the error: "`ORA-19913: unable to decrypt backup`"

```
RESTORE TABLESPACE USERS;
```

18. Issue the following commands to restore and recover the `users` tablespace.

```
SET DECRYPTION IDENTIFIED BY MyPassword;

ALTER TABLESPACE USERS OFFLINE;

RESTORE TABLESPACE USERS;

RECOVER TABLESPACE USERS;

ALTER TABLESPACE USERS ONLINE;
```

Recovery Scenario 22:

Restoring a Database on a New Host

Scenario Target:

- In this scenario, you will restore an entire database from a database backupset on a new host.

To avoid building up another machine from scratch for this scenario, you will drop the database from the appliance and restore it again into the same machine.

Note: although the scenario assumes that we are restoring the database into a new server because the source server becomes unavailable (unplanned migration), nearly the same procedure can be applied for planned migration.

Scenario Assumptions:

- Database on the source machine is not available
- The source database backup files are available.
- The Keystore file (if any) has been copied to its default location in the destination server.
- The database will be restored in different directories than their original directories in the source server.

Scenario Preparation

19. Invoke RMAN and login as target to `ORADB`

```
rman target /
```

20. Obtain the `DBID` and take a note of it.

21. Produce backupsets of the entire database, archive logs, and control files in the shared folder.

Observe that you can take backup of all the named input files using a single `BACKUP` command and define a `FORMAT` for each input file type.

```
BACKUP
FORMAT '/media/sf_extdisk/backup/ORADB%U.bck'
DATABASE TAG 'NEW_HOST_DB'
CURRENT CONTROLFILE TAG 'NEW_HOST_CTL'
FORMAT '/media/sf_extdisk/backup/ORADBCTL.bck'
SPFILE TAG 'NEW_HOST_SPFILE'
FORMAT '/media/sf_extdisk/backup/ORADBSPFILE.bck'
PLUS ARCHIVELOG TAG 'NEW_HOST_ARC'
FORMAT '/media/sf_extdisk/backup/ORADBARC%U.bck';
```

22. Make sure the backup pieces are saved in the intended location

```
HOST 'ls -al /media/sf_extdisk/backup/';
```

23. Exit from RMAN

Note: In real life scenario, you have to copy the Keystore file as well. For security reasons, do not include them with the backup files.

24. In the VM window, login as `oracle`, start `dbca` utility and delete `ORADB` database

```
dbca
```

25. Just to make sure that the source FRA and data file directories will not be used by our procedure code, delete them.

```
rm -r /u01/app/oracle/fra/ORADB/ORADB
rm -r /u01/app/oracle/oradata/ORADB
```

26. Create the directories where the datafiles and FRA files will be saved in the destination server. Create a directory under `oracle` home to save temporary files in it and the directory to save the audit files.

```
mkdir /u01/app/oracle/fra2
mkdir -p /u01/app/oracle/datafile/ORADB
mkdir -p /u01/app/oracle/admin/ORADB/adump
mkdir ~/temp
```

Recovery Prerequisites:

In real life scenario, before you proceed with the restore procedure, you should perform the following tasks on the destination host:

- Make the backup files are available to the destination host.
- Make the Keystore file (if any) available to the destination host and copy it to its default location.
- Install the same Oracle software in the destination server as the software that used to be running in the source server. It should be of the same release and architecture.
- Configure the operating system users and permissions.
- Configure the operating system environment variables.
- The directories where the database files will be located should be created and `oracle` user should have write permission on them.

As the database will be restored into the same machine, all the above tasks are already implemented.

Recovery Actions

Perform the following actions to restore the database from the backup files.

27. Create a PFILE for the restored database from the backed up SPFILE

a. Invoke RMAN and login as target to the local instance

```
rman target /
```

b. Start the instance in `NOMOUNT` mode

```
STARTUP FORCE NOMOUNT
```

c. Execute the following run block.

- Replace the *<dbid>* with the database DBID value

```
RUN{
 SET DBID <dbid>;
 RESTORE SPFILE TO PFILE '/home/oracle/temp/ORADBpfile.ora' FROM
'/media/sf_extdisk/backup/ORADBSPFILE.bck';
}
```

28. Edit the directory-based initialization parameters so that they point to the correct directories in the destination host. You do not need to perform this step if the directory structure in the destination is the same as the directory structure in the source.

a. Open the recovered PFILE with the vi editor

```
vi /home/oracle/temp/ORADBpfile.ora
```

b. Set the following parameters:

```
*.audit_file_dest='/u01/app/oracle/admin/ORADB/adump'
*.control_files='/u01/app/oracle/datafile/ORADB/control1.ctl','/u01/app/oracle/fra2/co
ntrol2.ctl'
*.db_create_file_dest='/u01/app/oracle/datafile/ORADB'
*.db_recovery_file_dest='/u01/app/oracle/fra2'
```

c. Save the file and exit from vi

29. Start SQL*Plus and connect to the local instance as sysdba. Create SPFILE from the PFILE.

```
sqlplus / as sysdba
CREATE SPFILE FROM PFILE='/home/oracle/temp/ORADBpfile.ora';
```

30. Invoke RMAN and connect as target to the local instance.

Note: If you are restoring the database into a new machine for testing purposes only, then do not connect to the recovery catalog database in the destination host because the restored database has the same DBID as the source databse.

```
rman target /
```

31. Start the database in NOMOUNT state again.

The target of this step is to let the instance starts using the SPFILE.

```
SHUTDOWN
STARTUP NOMOUNT
```

32. Restore the control files from the backup of the control file.

```
RESTORE CONTROLFILE FROM '/media/sf_extdisk/backup/ORADBCTL.bck';
```

33. Mount the database.

```
SHUTDOWN
STARTUP MOUNT
```

34. Catalog the backup files.

```
CATALOG START WITH '/media/sf_extdisk/backup/';
```

35. Verify that the backup files of the database and archive logs are registered.

```
LIST BACKUP OF DATABASE TAG 'NEW_HOST_DB';
LIST BACKUP OF ARCHIVELOG ALL TAG 'NEW_HOST_ARC';
```

36. Run the following query to build up the code to set the new redo log files. Copy the output of the query, paste it in a notepad, and put the destination format of each redo log member in the *three dots* placement.

Place one set of the redo group members in the new location of the datafiles (`/u01/app/oracle/datafile/ORADB/`) and the other set in the new FRA location (`/u01/app/oracle/fra2/`). Give the members any name of your convenience.

```
SELECT GROUP#, 'ALTER DATABASE RENAME FILE ''' || MEMBER || ''' TO ''...'';' CODE FROM
V$LOGFILE ORDER BY 1;
```

37. Execute the commands edited in the previous step.

You could run those statements in the run block that you are going to run in the next step. Personally, I prefer to run them beforehand just to make sure there is no issue in them.

Note: observe that the destination file names are static; which means they do not follow the OMF. If you want to have OMF generated redo log groups, drop the redo log groups (using `ALTER DATABASE DROP LOGFILE GROUP` command) and create them again (using `ALTER DATABASE ADD LOGFILE GROUP <n> SIZE <m>` command). The generated files will follow your OMF settings.

38. To prevent RMAN from trying to access backup files and archive logs that are not available, it is recommended to crosscheck RMAN repository and delete the expired entries. Execute the following commands.

```
CROSSCHECK BACKUPSET;
CROSSCHECK COPY DEVICE TYPE DISK;
DELETE EXPIRED BACKUPSET;
DELETE EXPIRED COPY;
```

39. List the backup of archive logs and obtain the "`Next SCN`" value of the last displayed archive log.

```
LIST BACKUP OF ARCHIVELOG ALL;
```

40. Restore the database to its new location by executing the following run block. Replace the `<obtained SCN>` with the SCN number obtained from the previous step.

The code targets at creating all the datafiles in a common location. If you want to specify different location for specific tablespace or data file, use the commands "`SET NEWNAME FOR TABLESPACE`" and "`SET NEWNAME FOR TABLESPACE`".

```
RUN
{
 SET NEWNAME FOR DATABASE TO '/u01/app/oracle/datafile/ORADB/%U';
 # redo log files renamed already in a previous step
 SET UNTIL SCN <obtained SCN>;
 RESTORE DATABASE;
 SWITCH DATAFILE ALL;
 RECOVER DATABASE;
}
```

41. Disable the BCT and enable it again.

Without doing this step, the database tries to create the BCT file in its old directory.

```
ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
```

42. Open the database with `RESETLOGS` option.

```
ALTER DATABASE OPEN RESETLOGS;
```

43. (optional) Retrieve the new database file names.

```
SELECT NAME FROM V$DATAFILE;
```

Clean Up

44. Delete the backup files.

```
host "rm -f /media/sf_extdisk/backup/*.*";
```

45. Shutdown `srv1` and restore it to the snapshot "**Practice 16 Start**". Start `srv1`.

46. Delete the snapshot "Practice 16 Start".

Summary

In this practice, you implemented the following recovery tasks:

- Recovery from losing the password file
- Recovery from losing temporary files
- Restoring from a password-encrypted backupset
- Restoring a database to a new host