

Using a Physical Standby Database for Read/Write Testing and Reporting

We can temporarily open a physical standby database for read/write, reporting.

How it's work:

- We must implement Flash back and create a restore point on the Standby database and open the standby database for read write operation.
- After completion of the read write operation, we flash back the standby database to a restore point.
- When the database is flashed back, Data automatically synchronizes with the primary database, without the need to re-create the physical standby database from a backup copy of the primary database.

Important:

While the database is activated, it is not receiving redo data from the primary database and cannot provide disaster protection. It is recommended that there be at least two physical standby databases participating in the configuration so that the primary database remains protected against data loss.

Workshop:

Complete bellow mention steps on Standby Database

Step 1 Set up a flash recovery area in Standby Database.

```
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE=20G;  
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='/DB/fback';
```

Step 2 Cancel Redo Apply and create a guaranteed restore point on Physical Standby database.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;  
SQL> CREATE RESTORE POINT before_application_patch GUARANTEE FLASHBACK  
DATABASE;
```

Help: find current scn number (SELECT CURRENT_SCN FROM V\$DATABASE;)

Complete bellow mention Step on primary database

Step 3 Archive the current log file. On the primary database, switch logs so the SCN of the restore point will be archived on the physical standby database:

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

Note: When using standby redo log files, this step is essential to ensure the database can be properly flashed back to the restore point.

Step 4 Defer the archival of redo data to the destination associated with the physical standby database that will be opened. For example:

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=DEFER;
```

Complete bellow mention step on Standby

Step 5 Activate the physical standby database:

```
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;
```

Note: In the event of a disaster on the primary machine, you need to make the standby database active as soon as possible. In this example, it is assumed that the primary database server is not available and that the standby database needs to be activated as the primary database immediately. The standby database will need to be activated in its current state - no other redo from the primary database will be applied as it is assumed to be down

Step 6 If the physical standby database has been opened read-only since the instance was started, perform this step. Otherwise, skip to step 7. Enter the following statement to shut down and restart the physical standby database:

```
SQL> SHUTDOWN IMMEDIATE;  
SQL> STARTUP MOUNT FORCE;
```

7. Set the protection mode to maximum performance and open the database for read/write access:

```
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE;  
SQL> ALTER DATABASE OPEN;
```

Use the activated database for reporting and testing.

Step 8 Finally Revert the activated database back to a physical standby database.

```
SQL> SHUTDOWN IMMEDIATE;  
SQL> STARTUP MOUNT FORCE;  
SQL> FLASHBACK DATABASE TO RESTORE POINT before_application_patch;  
SQL> ALTER DATABASE CONVERT TO PHYSICAL STANDBY;  
SQL> STARTUP MOUNT FORCE;
```

Step 9 put standby database in recover mode

Step 10 Reenable archiving to the physical standby database destination.
On the primary database, issue the following statement to reenale archiving to the physical standby database:
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE;