Open Physical Standby database for testing and re-enable Dataguard afterwards

This document is based on "Data Guard Scenarions" at http://oracle-documentation.web.cern.ch/oracle-documentation/10gr2doc/server.102/b14239/scenarios.htm#CIHIABD

- **Note** : when Standby is open, it does not receive redo data from primary and cannot provide disaster protection

Open standby in read/write mode

- On Standby - stop all except one database instance
    - srvctl stop instance -d ` echo ${ORACLE_SID%[0-9]}` -i ` echo $ORACLE_SID` ## on the nodes to be stopped

- On Standby - redo apply and create restore point

```
STANDBY> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
STANDBY> ALTER DATABASE FLASHBACK ON; -- case you have readonly tablespaces, bug 6150680, f
STANDBY> CREATE RESTORE POINT before_stop GUARANTEE FLASHBACK DATABASE;
```

- On Primary - switch logs so the SCN of restore point is archived in the standby
    - PRIMARY> ALTER SYSTEM ARCHIVE LOG CURRENT;

- On Primary and standby - disable archive redo shipping
    - PRIMARY> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_x=DEFER SID='*'; ## change x by the correct archiver process #
    - STANDBY> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_x=DEFER SID='*'; ## change x by the correct archiver process #

- On Standby - Activate and open it
    - STANDBY> ALTER DATABASE ACTIVATE STANDBY DATABASE;
    - STANDBY> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE;
    - STANDBY> ALTER DATABASE OPEN;

- On Standby - start other instances, if necessary
    - srvctl start instance -d ` echo ${ORACLE_SID%[0-9]}` -i ` echo $ORACLE_SID` ## on the nodes to be started

- Set parameters to avoid jobs and streams to run
    - STANDBY> ALTER SYSTEM set JOB_QUEUE_PROCESSES=0 sid='*' scope='memory'; -- avoid that jobs run from standby
    - STANDBY> ALTER SYSTEM set PARALLEL_MAX_SERVERS=0 sid='*' scope='memory'; -- avoid that streams start

Flashback database and resume standby

- On Standby - stop all except one database instance
    - srvctl stop instance -d ` echo ${ORACLE_SID%[0-9]}` -i ` echo $ORACLE_SID` ## on the nodes to be stopped

- On Standby, put in mount mode and flashback database
    - STANDBY> STARTUP MOUNT FORCE;
    - STANDBY> FLASHBACK DATABASE TO RESTORE POINT before_stop;
    - STANDBY> ALTER DATABASE CONVERT TO PHYSICAL STANDBY;
    - STANDBY> STARTUP MOUNT FORCE;

Standby is not much behind primary

- (check archive retention on /etc/inittab of pdb-backup, usually 1 day), one can just let oracle automatically fetch the gap
    - `STANDBY> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;`
    - `STANDBY> ALTER DATABASE FLASHBACK OFF; -- case you have readonly tablespaces, bug 6150680, fixed 11.1.0.7`

- Drop restore point:
    - `STANDBY>  DROP RESTORE POINT before_stop;`

- Re-enable the archive shipping on both ends:
    - `PRIMARY>  ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_x=ENABLE SID='*';` ## change x by the correct archiver process #
    - `STANDBY>  ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_x=ENABLE SID='*';` ## change x by the correct archiver process #

- On Standby - start other instances, if necessary
    - `srvctl start instance -d ` echo ${ORACLE_SID%[0-9]}`  -i ` echo $ORACLE_SID`` ## on the nodes to be started

---

Standby several days behind primary

- Standby would not be able to fetch, do a incremental backup of primary or fetch from backup the missing archive logs

- **recover db standby DB using backup and refresh the controlfile** as detailed in RecoveryStandbyGaps

- On Standby - stop all except one database instance
    - `srvctl stop instance -d ` echo ${ORACLE_SID%[0-9]}`  -i ` echo $ORACLE_SID`` ## on the nodes to be stopped

- Restart Flashback Database:
    - `STANDBY> ALTER DATABASE FLASHBACK OFF;`
    - `STANDBY> ALTER DATABASE FLASHBACK ON;`

- Start Redo Apply on the physical standby database and drop restore point
    - `STANDBY> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;`
    - `STANDBY> DROP RESTORE POINT before_stop;`

- Remove the incremental backups from the standby system:
    - `RMAN_STANDBY> DELETE BACKUP device type disk TAG 'FOR STANDBY';`

- Manually remove the incremental backups from the primary system:
    - `rm /ORA/dbs00/oracle/backup/ForStandby_*`

- Re-enable the archive shipping on both ends:
    - `PRIMARY>  ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_x=ENABLE SID='*';` ## change x by the correct archiver process #
    - `STANDBY>  ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_x=ENABLE SID='*';` ## change x by the correct archiver process #

- On Standby - start other instances, if necessary
    - `srvctl start instance -d ` echo ${ORACLE_SID%[0-9]}`  -i ` echo $ORACLE_SID`` ## on the nodes to be started

Dataguard checks

```
SQL> alter session set nls_date_format='dd-mm-yyyy hh24:mi:ss';
SQL> select thread#, sequence#,  first_time, next_time, applied from v$archived_log order b

SQL> select * from v$dataguard_status;
SQL> alter system archive log current;
```

Limitations of Flashback Database

Because Flashback Database works by undoing changes to the datafiles that exist at the moment that you run the command, it has the following limitations:

- Flashback Database can only undo changes to a datafile made by an Oracle database. It cannot be used to repair media failures, or to recover from accidential deletion of datafiles.
- You cannot use Flashback Database to undo a shrink datafile operation.
- If the database control file is restored from backup or re-created, all accumulated flashback log information is discarded. You cannot use FLASHBACK DATABASE to return to a point in time before the restore or re-creation of a control file.
- When using Flashback Database with a target time at which a NOLOGGING operation was in progress, block corruption is likely in the database objects and datafiles affected by the NOLOGGING operation. For example, if you perform a direct-path INSERT operation in NOLOGGING mode, and that operation runs from 9:00 to 9:15 on April 3, 2005, and you later need to use Flashback Database to return to the target time 09:07 on that date, the objects and datafiles updated by the direct-path INSERT may be left with block corruption after the Flashback Database operation completes.
- shrinking a datafile or dropping a tablespace can prevent flashing back the Database

This topic: PSSGroup > OpenDataGuardForTest
Topic revision: r15 - 2009-02-11 - MiguelAnjo