STANDBY DATABASE SETUP

PRODUCTION AND STANDBY ENVIRONMENT :

➢ PRE-CHECKS :

- Database Information :

```
set lines 200 pages 2000
col host_name for a20
select name,open_mode,log_mode,database_role,instance_name,host_name from
v$database,v$instance;
```

- Check space on the production and standby server for backup and archive logs.

- Check Standby server configuration Like CPU,MEMORY,It should be the same as Production server.

➢ SETUP LISTENER AND TNS ENTRIES

PRODUCTION ENVIRONMENT :

Location: $ORACLE_HOME/network/admin

→ **listener.ora file** || NO CHANGE

```
LISTENER =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1522))
    (ADDRESS = (PROTOCOL = TCP)(HOST = <ip_or_host>)(PORT = 1522))
  )
```

→ **tnsnames.ora** || ADD DR's TNS Entry

```
<DB_NAME>=
 (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = <ip_or_host>)(PORT = 1522))
  (CONNECT_DATA =
   (SERVER = DEDICATED)
   (SERVICE_NAME = <DB_NAME>)
  )
 )



<DR_DB_NAME> =
                (DESCRIPTION =
                 (ADDRESS = (PROTOCOL = TCP)(HOST = <ip_or_host>)(PORT = 1535))
                   (CONNECT_DATA =
                   (SERVER = DEDICATED)
                   (SERVICE_NAME = <DR_DB_NAME>)
                  )
                 )
```

STANDBY ENVIRONMENT :

Location: $ORACLE_HOME/network/admin

→ **listener.ora** file

```
LISTENER =
   (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1535))
      (ADDRESS = (PROTOCOL = TCP)(HOST = <ip_or_host>)(PORT = 1535))
   )

SID_LIST_LISTENER =
 (SID_LIST =
(SID_DESC =
     (ORACLE_HOME = <oracle_home_location>)
     (SID_NAME = <DR_DB_NAME>)
   )
 )
```

→ **tnsnames.ora**

```
<DB_NAME>=
 (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST =<ip_or_host>)(PORT = 1522))
   (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SERVICE_NAME = <DB_NAME>)
   )
 )


<DR_DB_NAME> =
                (DESCRIPTION =
                  (ADDRESS = (PROTOCOL = TCP)(HOST = <ip_or_host>)(PORT = 1535))
                    (CONNECT_DATA =
                    (SERVER = DEDICATED)
                    (SERVICE_NAME = <DR_DB_NAME>)
                   )
                  )
```

➢ VERIFY TNSPING FROM PRODUCTION AND STANDBY SERVERS

→ FROM PRODUCTION

tnsping **<prod_db_name>**
tnsping **<standby_db_name>**

→ FROM STANDBY

tnsping **<prod_db_name>**
tnsping **<standby_db_name>**

NOTE : tnsping should be successful from PRODUCTION to STANDBY
FROM STANDBY to PRODUCTION We can done it by later || BUT BOTH ARE MUST

STEPS : ON PRODUCTION ENVIRONMENT

1] Take RMAN Full Backup of Production Database, Also take Backup of the control file for a standby database.

backup as compressed backupset database format
'**<location>**/RMAN_FULLBackup_%U_%T' plus archivelog format
'**<location>**/arch_backup_%U_%T'
backup current controlfile for standby format '***<location>***/control_%d_%u_%s';
backup spfile format '**<location>**/spfile_new_%U_%T';

2] Transfer the standby control file, password file And Full backup on STANDBY SIDE.

Password File :
cd $ORACLE_HOME/dbs
scp ***<file_name>*** ***<ip>***:$ORACLE_HOME/dbs

Rename it to the standby database name.
The username is required to be SYS and password to be the same on Primary and Standby.
The password file name must match the instance name/SID used at the standby site,not the DB_NAME.

Standby Control File :
cd ***<location>***
scp ***<file_name>*** ***<ip>:<location>***

RMAN Full Backup :
cd ***<backup_location>***
scp ***<files>*** ***<ip>:<location>***

3] SET DR PARAMETERS

a] Check LOGGING IF ENABLE then no need to change

```
select force_logging from v$database;
```

To Enable : ALTER DATABASE FORCE LOGGING;

Why force logging?
If force logging is enabled, all the database changes will be logged in redo log files, even for nologging operations.The above means that if FORCE LOGGING is enabled at a higher level, the NOLOGGING at a lower level has no effect. That means that even for a nologging table redo information may still be logged.
This feature can be enabled at TWO levels:

– Database level
– Tablespace level

Database level:

If the force logging is enabled at a database level, all the operations happening in the database are logged.

b] Configure a Standby Redo Log on Primary

```
set lines 180
col MEMBER for a60
select b.thread#, a.group#, a.member, b.bytes FROM v$logfile a, v$log b WHERE
a.group# = b.group#;
```

```
 GROUP#     MEMBER                                                     BYTES
---------- ------------------------------------------------------------ ----------
         3 /u01/oracle/oradata/ORCL/redo03.log                        209715200
         2 /u01/oracle/oradata/ORCL/redo02.log                        209715200
         2 /u01/oracle/oradata/ORCL/redo01.log                        209715200
```

Since we have 3 online redo log file groups, we need to create 4(3+1) Standby redo log file groups. Standby Redo logs files come into picture only when protection mode is Maximum Availability and Maximum Protection.

```
alter database add standby logfile group 4 ('<location>/redo04.log') size 200m;
alter database add standby logfile group 5 ('<location>/redo05.log') size 200m;
alter database add standby logfile group 6 ('<location>/redo06.log') size 200m;

select * from v$logfile;
```

```
    GROUP# STATUS  TYPE    MEMBER                                          IS_  CON_ID
---------- ------- ------- ----------------------------------------------- ---  ----------
         3         ONLINE  /u01/oracle/oradata/ORCL/redo03.log           NO    0
         2         ONLINE  /u01/oracle/oradata/ORCL/redo02.log           NO    0
         1         ONLINE  /u01/oracle/oradata/ORCL/redo01.log           NO    0
         4         STANDBY /u01/oracle/oradata/ORCL/redo04.log           NO    0
         5         STANDBY /u01/oracle/oradata/ORCL/redo05.log           NO    0
         6         STANDBY /u01/oracle/oradata/ORCL/redo06.log           NO    0
         7         STANDBY /u01/oracle/oradata/ORCL/redo07.log           NO    0
```

select a.group#, a.member, b.bytes from v$logfile a, v$standby_log b where a.group# = b.group#;

```
GROUP#     MEMBER                                                      BYTES
---------- ------------------------------------------------------------ ----------
         4 /u01/oracle/oradata/ORCL/redo04.log                         209715200
         5 /u01/oracle/oradata/ORCL/redo05.log                         209715200
         6 /u01/oracle/oradata/ORCL/redo06.log                         209715200
         7 /u01/oracle/oradata/ORCL/redo07.log                         209715200
```

4] VERIFY ARCHIVE MODE ENABLED ON PRIMARY

archive log list;

5] SET PRIMARY DATABASE INITIALIZATION PARAMETERS

a] ALTER SYSTEM SET DB_UNIQUE_NAME='**<PROD_UNIQUE_NAME>**' scope=spfile;

b] ALTER SYSTEM SET LOG_ARCHIVE_CONFIG='DG_CONFIG=(**<PROD_UNIQUE_NAME>,<DR_UNIQUE_NAME>**)' scope=both;

c] ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=<archive_location> VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=**<PROD_UNIQUE_NAME>'** scope=both;

d] ALTER SYSTEM SET LOG_ARCHIVE_DEST_2='SERVICE=**<DR_UNIQUE_NAME>** NOAFFIRM ASYNC VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=**<DR_UNIQUE_NAME**>' scope=both;

e] ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_1=ENABLE scope=both;

f] ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE scope=both;

g] ALTER SYSTEM SET LOG_ARCHIVE_FORMAT='%t_%s_%r.arc' SCOPE=SPFILE;

h] ALTER SYSTEM SET LOG_ARCHIVE_MAX_PROCESSES=30 scope=both;

i] ALTER SYSTEM SET REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE SCOPE=SPFILE;

note: The FAL_CLIENT database initialization parameter is no longer required from 11gR2

j] ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=AUTO;
k] create pfile='**<location>**/init**<dbname>**.ora' from spfile;

Now do some changes in pfile which is created in step k]

Parameters :

*.control_files
*.audit_file_dest
*.db_file_name_convert
*.db_unique_name
*.log_file_name_convert

Now Transfer PFILE to STANDBY SIDE.

scp **<location>**/init***<dbname>***.ora <ip>:$ORACLE_HOME/dbs/

STEPS : ON STANDBY ENVIRONMENT

Create required directories

```
mkdir -p /u01/app/oracle/admin/<DR_DB_NAME>/adump
mkdir -p /u01/oracle/oradata/<DR_DB_NAME>
mkdir -p /u01/oracle/archive/<DR_DB_NAME>
```

1. START THE DATABASE IN NOMOUNT USING PFILE

```
export ORACLE_SID=<STANDBY_SID>
export ORACLE_HOME=<ORACLE_HOME_PATH>
export PATH=$ORACLE_HOME/bin:$PATH

sqlplus / as sysdba

startup nomount pfile='$ORACLE_HOME/dbs/initdbname.ora';
```

2. RESTORE DATABASE

```
rman target /

restore standby controlfile from '/home/oracle/Control_XE_bt1vmp5f_381';

sql 'alter database mount standby database';
```

```
catalog start with '/backup/RMAN_BKP';
```

3. CREATE .sh FILE TO RUN IN NOHUP UTILITY

```
export ORACLE_SID=<STANDBY_SID>
export ORACLE_HOME=<ORACLE_HOME_PATH>
export PATH=$ORACLE_HOME/bin:$PATH
tdt=`date "+%d%m%Y"`
export tdt

rman target / cmdfile '<location>/restore.sql' log <location>/restore_$tdt.log
```

.SQL FILE

```
run {
allocate channel c1 device type disk;
allocate channel c2 device type disk;
restore database;
release channel c1;
release channel c2;
}
```

RUN THE FILE IN NOHUP

```
cd <location>
nohup sh restore.sh &
```

4. CHECK STATUS

```
set lines 234 pages 300
select SID,INPUT_BYTES/1024/1024/1024,OUTPUT_BYTES/1024/1024/1024,STATUS
from v$rman_status where status='RUNNING' AND operation like '%RESTORE%';
```

ON PRODUCTION :

1] Stop and Start the RFS

```
alter system set log_archive_dest_state_3='DEFER' sid='*' scope=both;
```

```
alter system set log_archive_dest_state_3='ENABLE' sid='*' scope=both;
```

ON STANDBY :

2] Check RFS is started or not IF Started the start recovery

```
select process,pid,status,thread#,sequence# from v$managed_standby;

alter  database recover managed standby database disconnect from session;
```

3] CHECK MAX SEQUENCE BOTH ON PRODUCTION AND STANDBY

```
select max(sequence#) from v$archived_log where applied='YES';
```

On Standby Database **SEQUENCE** should be same as production database.

Command to check DR SYNC status :

```
set lines 456 pages 456
select TO_CHAR(sysdate, 'DD-MM-YYYY HH24:MI:SS') from dual;

select name,open_mode,log_mode from v$database;

select max(sequence#) from v$archived_log where applied='YES';
select inst_id,process,pid,STATUS,CLIENT_PROCESS,BLOCK#,THREAD#,sequence# from
gv$managed_standby;

SELECT ARCH.THREAD# "THREAD", ARCH.SEQUENCE# "LAST SEQUENCE RECEIVED",
APPL.SEQUENCE# "LAST SEQUENCE APPLIED", (ARCH.SEQUENCE# - APPL.SEQUENCE#)
"DIFFERENCE"
FROM (SELECT THREAD# ,SEQUENCE# FROM GV$ARCHIVED_LOG WHERE
(THREAD#,FIRST_TIME ) IN (SELECT THREAD#,MAX(FIRST_TIME) FROM GV$ARCHIVED_LOG
GROUP BY THREAD#)) ARCH,
(SELECT THREAD# ,SEQUENCE# FROM GV$LOG_HISTORY WHERE (THREAD#,FIRST_TIME ) IN
(SELECT THREAD#,MAX(FIRST_TIME) FROM GV$LOG_HISTORY GROUP BY THREAD#)) APPL
WHERE ARCH.THREAD# = APPL.THREAD# ORDER BY 1;
```