# Creating an Oracle 12c R2 Physical Standby RAC Database from a Primary RAC database

A tutorial by Ahmed Baraka

Tutorial Objective

In this tutorial, you will configure Oracle Data Guard between two Oracle 12c R2 RAC databases. Specifically, you will create an Oracle 12c R2 physical standby RAC database from a primary RAC database.

The following diagram is a high level description of the environment that you will create:
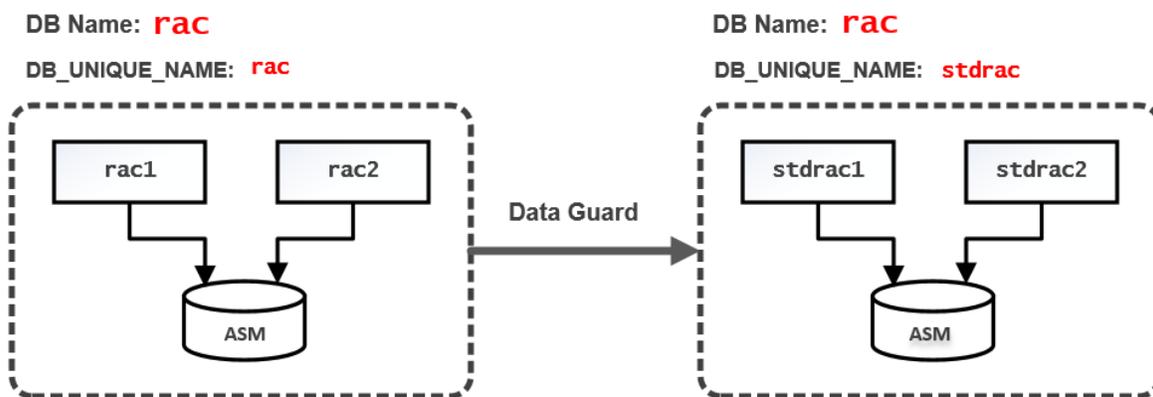
Required Knowledge

To understand the steps in this tutorial, you have to have the fundamental knowledge of the following:

- Oracle database administration
- Oracle RAC administration
- Oracle Data Guard
- Basics of Linux operating system

Tutorial Assumptions

The steps are described in this tutorial document with the following assumptions:

- The virtual machines have been created using Oracle VirtualBox software.

- You have four virtual machines. The first two machines are the nodes of the primary database. The other pair is the two nodes of the standby database.

- The first two machines are the cluster of the primary database. It has an Oracle 12c R2 RAC database installed in it.

- The other two machines are the cluster of the standby database. You have Oracle 12c R2 Grid Infrastructure software and Oracle 12c R2 database software installed in this cluster. You have only ASM configured in this cluster, but **no** database is created in it.
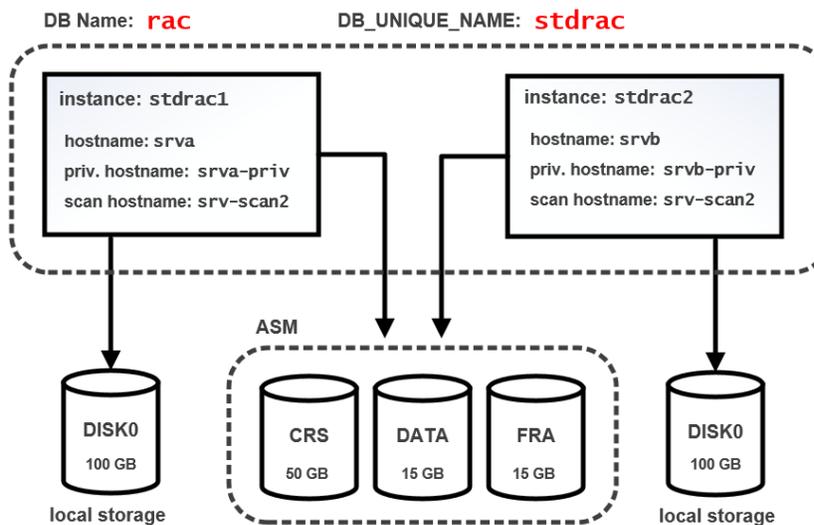
Practice Environment Architecture

The Primary Database Architecture

The following diagram shows the Primary database architecture:
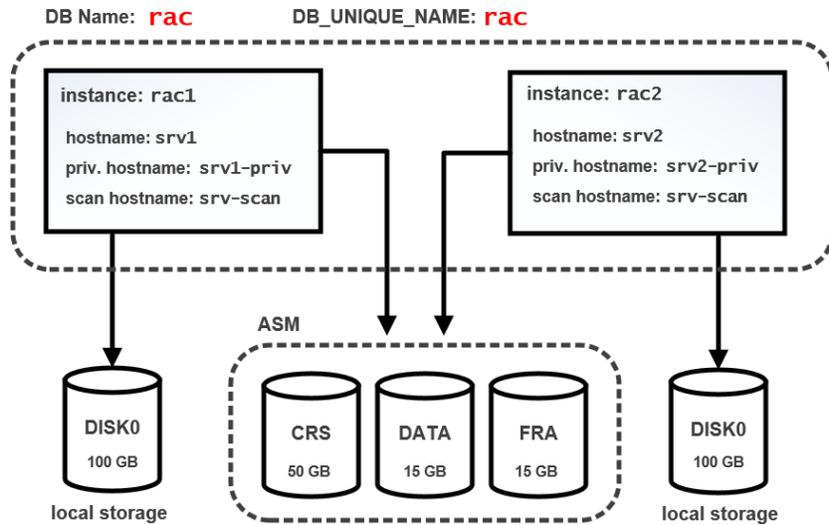
The Standby Database Architecture

The following diagram shows the Standby database architecture:

Note: Observe that the DB Name is the same in the two databases, but the DB_UNIQUE_NAME is different.

Required Software

Following is the required software to implement this tutorial:

- Putty
- Oracle VirtualBox 5.1
- Oracle Grid Infrastructure 12c R2
- Oracle Database 12c R2

Required Hardware

To implement this tutorial, you need either of the following:

- Two physical PCs with 16-GB memory, 280 GB free harddisk **each**.

OR

- One physical PC with 32-GB memory, half terabyte free harddisk.

Operating System Release

The virtual machines used to implement this tutorial have the following operating system installed in them:

- Oracle Linux 6.7 x86 64-bit

However, the steps described in this tutorial should work fine with any OS release supported by Oracle database.

Data Guard Configuration Specifications

The tutorial will create a physical standby database with the following specifications:

- Protection mode: Maximum Performance
- Fast-start failover: Disabled
- The management interface: SQL*Plus
- Standby database type: Physical Standby

What not covered by this tutorial

The tutorial will **not** cover the steps to perform the following:

- Create VirtualBox machines from scratch.
- Install Oracle 12c R2 Grid Infrastructure and Database software in the virtual machines.
- Create Oracle RAC database in the primary cluster.

Data Guard Considerations in RAC

This is an overview of the difference between configuring Data Guard in a non-cluster single-instance database and configuring Data Guard in RAC.

Follow are the most important considerations that you should take into your account when configuring Data Guard in RAC:

- When you use "duplicate database for standby from active database" command in RMAN, a single-instance RAC database is created. You have to manually set the RAC-specific parameters in the created instance including the INSTANCE_NUMBER and INSTANCE_NAME parameters, enable the other instance in the other node, and add the created database in the clusterware OCR as a new resource.

- The INSTANCE_NUMBER and INSTANCE_NAME parameters must be manually changed in the second standby instance.

- Either of the primary database or standby database could be a single-instance non-cluster database. This tutorial demonstrates the steps to configure Data Guard between two RAC databases.

- The redo log files in the standby redo log must reside in a location that is accessible by all of the standby database instances. Every standby database instance must be configured to archive the standby redo log to the same location. This tutorial suggests creating them in the FRA disk group in ASM.

- If OMF is not configured, LOG_ARCHIVE_FORMAT must be set. When this parameter is set, you must use the thread parameters (%t or %T) to uniquely identify the archived redo log files for each instance.

Tutorial Overview

The tutorial implements the standard procedure to create a physical standby database. In high level, it goes through the following stages:

I. Prepare the primary database:

- o Enable archiving mode
- o Configure ARCHIVELOG DELETION POLICY in RMAN
- o Enable forced logging
- o Configure standby redo log files (SRL)
- o Set the related primary database initialization parameters
- o Enable Flashback Database
- o Set CONTROL_FILE_RECORD_KEEP_TIME parameter
- o Configure the tnsnames.ora file

II. Prepare the standby system:

- o Create the required directories
- o Create a static listener entry for the standby database
- o Create an init<SID>.ora file
- o Copy the password file from the primary system
- o Configure the tnsnames.ora file

III. Create the physical standby database:

- o Use RMAN to create the physical database online from the primary database
- o Set the standby database parameters
- o Start the Redo Apply
- o Verify the physical standby database is performing properly
- o Post creation steps

IV. Test and monitor the standby database.

Verifying the Environment

In the following steps, you will just verify that machines and the database are working and configured properly.

1. Make sure that every node can see all the other nodes in the network.

```
ping -c 3 srv1
ping -c 3 srv2
ping -c 3 srva
ping -c 3 srvb

ping -c 3 srv1-vip
ping -c 3 srv2-vip
ping -c 3 srva-vip
ping -c 3 srvb-vip

ping -c 3 srv-scan
ping -c 3 srv-scan2
```

2. In the standby cluster, verify that the clusterware services are running.

```
su - grid
crsctl check crs
```

3. In the standby cluster, verify that the clusterware resources are ONLINE.

```
crs_stat -t
```

4. In the standby cluster, checking the status of the necessary applications are running in all the cluster nodes.

```
srvctl status nodeapps -n srva
srvctl status nodeapps -n srvb
```

Preparing the Primary Database

5. Enable the archiving mode in the primary database.

```
# to make sure the database is working in archivelog mode:
su - oracle
sqlplus / as sysdba
SELECT LOG_MODE FROM V$DATABASE ;

# to know the location of the archive log files:
show parameter LOG_ARCHIVE_DEST_1

# to enable the ARCHIVELOG mode in the database (if it is disabled):
ALTER SYSTEM SET LOG_ARCHIVE_DEST_1 = 'LOCATION=USE_DB_RECOVERY_FILE_DEST' scope=BOTH;
# Note: because OMF is enabled, setting the LOG_ARCHIVE_FORMAT parameter has no effect.
srvctl stop database -d rac -o immediate
srvctl start database -d rac -o mount
sqlplus / as sysdba
ALTER DATABASE ARCHIVELOG;
srvctl stop database -d rac
srvctl start database -d rac
ALTER SYSTEM SWITCH LOGFILE;
SELECT NAME FROM V$ARCHIVED_LOG;
```

6. Enable forced logging option in the database.

```
SELECT FORCE_LOGGING FROM V$DATABASE;
ALTER DATABASE FORCE LOGGING;
SELECT FORCE_LOGGING FROM V$DATABASE;
```

7. Determine the number of required standby redo log file groups and create them.

    a. Obtain the number of redo log files and the size of each file.

       The number of required standby redo log files is determined by the following formula:

       `(maximum number of redo logfiles +1) * number of threads`

       Based on the output of the following SELECT statements, you need (4+1)*2=10 standby redo log file.

```
SELECT GROUP#, BYTES/1024/1024 MB FROM V$LOG ORDER BY 1,2;

col INSTANCE format a5
SELECT DISTINCT THREAD#, INSTANCE FROM GV$THREAD ORDER BY 1,2;
```

b. Create the required standby redo log files.

In normal operation, those files are not used by the primary database. They are used after switchover or failover and that is why you create them in the primary database now.

```
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 '+FRA' SIZE 50M ;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 '+FRA' SIZE 50M ;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 '+FRA' SIZE 50M ;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 '+FRA' SIZE 50M ;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 '+FRA' SIZE 50M ;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 '+FRA' SIZE 50M ;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 '+FRA' SIZE 50M ;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 '+FRA' SIZE 50M ;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 '+FRA' SIZE 50M ;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 '+FRA' SIZE 50M ;


# verify:
SELECT INST_ID, GROUP#,THREAD#,SEQUENCE#,ARCHIVED,STATUS
FROM GV$STANDBY_LOG ORDER BY 2,1;
```

8. In the primary database cluster, configure the `tnsnames.ora` file in all its nodes: `srv1` and `srv2`.

   Do not copy the code from the PDF file. Copy it from the attached `tnsnames.ora` file.

```
su - oracle
vi /u01/app/oracle/product/12.2.0/db_1/network/admin/tnsnames.ora

RAC =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = srv-scan)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rac.localdomain)
    )
  )

STDRAC =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = srva)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = stdrac.localdomain))
  )
```

9. In the standby cluster node `srva`, add the connection descriptors below to the `tnsnames.ora` file.

```
su - oracle
vi /u01/app/oracle/product/12.2.0/db_1/network/admin/tnsnames.ora

RAC =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = srv-scan)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rac.localdomain)
    )
  )

# (UR=A) setting is only needed in the standby instance
STDRAC =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = srva)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = stdrac.localdomain) (UR=A))
  )
```

10. In the standby cluster node `srvb`, add the connection descriptors below to the `tnsnames.ora` file.

```
su - oracle
vi /u01/app/oracle/product/12.2.0/db_1/network/admin/tnsnames.ora

RAC =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = srv-scan)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rac.localdomain)
    )
  )

STDRAC =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = srvb)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = stdrac.localdomain))
  )
```

11. Test the connection configuration in all machines:

```
tnsping rac
tnsping stdrac
```

12. In the primary database, set the initialization parameters required by the Data Guard configuration.

```
sqlplus / as sysdba

show parameter DB_UNIQUE_NAME

# the value of this parameter is case-sensitive:
ALTER SYSTEM SET LOG_ARCHIVE_CONFIG='DG_CONFIG=(rac,stdrac)' SCOPE=BOTH SID='*';

ALTER SYSTEM SET LOG_ARCHIVE_DEST_2= 'SERVICE=stdrac ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=stdrac' SCOPE=BOTH SID='*';

# should be EXCLUSIVE
show parameter REMOTE_LOGIN_PASSWORDFILE

# FAL = fetch archive log
ALTER SYSTEM SET FAL_SERVER='stdrac' SCOPE=BOTH SID='*';

# No need to set DB_FILE_NAME_CONVERT if the standby system has the disk group names or
#  the same datafile destination directory structure
show parameter DB_FILE_NAME_CONVERT

show parameter STANDBY_FILE_MANAGEMENT
ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT='AUTO' SCOPE=BOTH SID='*';
```

```
# set the maximum number of ARCn processes that can be created ( default 4):
ALTER SYSTEM SET LOG_ARCHIVE_MAX_PROCESSES=8 SCOPE=BOTH SID='*';

# Get list of directory-dependent parameters and create the directories in the standby
server:
SELECT NAME, VALUE FROM V$PARAMETER WHERE upper(VALUE) LIKE upper('%/rac/%') AND NAME LIKE
'%dest';
```

13. Enable Flashback Database (optional but recommended)

```
ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET =2880 scope=BOTH;
ALTER DATABASE FLASHBACK ON;
```

14. Set CONTROL_FILE_RECORD_KEEP_TIME parameter.

This parameter controls the number of days in which the control file records should be retained.

```
show parameter CONTROL_FILE_RECORD_KEEP_TIME
ALTER SYSTEM SET CONTROL_FILE_RECORD_KEEP_TIME=30 scope=BOTH SID='*';
```

15. In all nodes, if `sqlnet.ora` file is not configured, create it and add the following setting in it.

```
vi /u01/app/oracle/product/12.2.0/db_1/network/admin/sqlnet.ora
NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
```

Prepare the Standby System

16. On the standby server `srva`, create the directories pointed at by the parameters. As ASM is being used, the directories under ASM diskgroups should also be created.

```
su - grid
asmcmd
cd DATA
mkdir RAC
mkdir STDRAC
ls

-- in srva and srvb
su - oracle
mkdir -p /u01/app/oracle/admin/stdrac
mkdir /u01/app/oracle/admin/stdrac/adump
mkdir /u01/app/oracle/admin/stdrac/cdump
```

17. In `srva`, as `oracle`, create a pfile named as `init<SID>.ora` in the default directory. Just put only DB_NAME in it.

RMAN will create the full spfile.

```
su - oracle
echo 'DB_NAME=rac' > $ORACLE_HOME/dbs/initstdrac1.ora
```

18. In `srv1`, copy the password file from the primary database system to the machines `srva` and `srvb`.

The password file is saved in the ASM. Use `asmcmd` utility to take a copy of it into a staging file in the file system. Then copy the staging file to the standby systems.

```
# obtain the full path name of the current password file:
su - grid
asmcmd pwget --dbuniquename rac

# take a copy of it in the file system:
asmcmd pwcopy +DATA/RAC/PASSWORD/pwdrac.256.961023093 /tmp/orapwstdrac

# copy the staging file to the default directory in the standby cluster nodes:
su -
scp /tmp/orapwstdrac srva:/u01/app/oracle/product/12.2.0/db_1/dbs/orapwstdrac1
scp /tmp/orapwstdrac srvb:/u01/app/oracle/product/12.2.0/db_1/dbs/orapwstdrac2

# in srva change the ownership of the file to oracle
su -
chown oracle:oinstall /u01/app/oracle/product/12.2.0/db_1/dbs/orapwstdrac1

# in srvb change the ownership of the file to oracle
ssh srvb
chown oracle:oinstall /u01/app/oracle/product/12.2.0/db_1/dbs/orapwstdrac2
exit
```

19. In srva, create a static entry in the local listener to the local standby database instance.

Do not copy the code from the PDF file. Copy it from the attached file.

```
su – grid
cp $TNS_ADMIN/listener.ora $TNS_ADMIN/listener.ora.backup
vi $TNS_ADMIN/listener.ora

# add the following to the file:
SID_LIST_LISTENER=
   (SID_LIST=
       (SID_DESC=
          (GLOBAL_DBNAME=stdrac.localdomain)
          (SID_NAME=stdrac1)
          (ORACLE_HOME=/u01/app/oracle/product/12.2.0/db_1)
        )
     )

# restart the listener:
srvctl stop listener
srvctl start listener
lsnrctl services | grep stdrac
```

Creating the Standby Database

20. (optional) Open a new Putty session and connect to the primary machines as `oracle` and use it to monitor the alter log file of the primary database. This may help you to diagnose any issue you may face when creating the standby database.

```
# in srv1:
tail -f /u01/app/oracle/diag/rdbms/rac/rac1/trace/alert_rac1.log

# in srv2:
tail -f /u01/app/oracle/diag/rdbms/rac/rac2/trace/alert_rac2.log
```

21. In the standby machine `srva`, start up the standby instance

```
# in srva:
su - oracle
export ORACLE_SID=stdrac1

# now start up the instance
sqlplus '/ as sysdba'
STARTUP NOMOUNT pfile='/u01/app/oracle/product/12.2.0/db_1/dbs/initstdrac1.ora'
exit
```

22. Create the standby database from the primary database using RMAN DUPLICATE command.

The command can be run from the standby database, i.e. the data will be pulled from the primary, or it can be run from the primary, where the data will be pushed to the standby.

The code below command is to be run on the standby.

Note: If the DUPLICATE command fails after it shutsdown the auxiliary instance, you must shutdown then start up the standby instance again. You also should drop the created SPFILE before you can run the same DUPLICATE command again.

Note: In my implementation, when I executed the DUPLICATE command, RMAN looked for the SPFILE in the dbs directory under the Oracle home, not in its real location in the ASM diskgroup. It returned the following errors:

```
ORA-19505: failed to identify file "../dbs/spfilerac*.ora"
ORA-27037: unable to obtain file status
```

The root cause of the errors was note clear to me and I had to make copy of the SPFILE in the dbs directory as a workaround to the issue.

```
# in srva:
rman

CONNECT TARGET sys/oracle@rac;
CONNECT AUXILIARY sys/oracle@stdrac;
run {
 allocate channel prmy1 type disk;
 allocate channel prmy2 type disk;
 allocate auxiliary channel stby1 type disk;
 DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE
 SPFILE
 set 'db_unique_name'='stdrac'
 set instance_name='stdrac1'
 set instance_number='1'
 set db_create_online_log_dest_1='+FRA'
 set db_create_online_log_dest_2='+DATA'
 set db_recovery_file_dest='+FRA'
 set audit_file_dest='/u01/app/oracle/admin/stdrac/adump'
 set core_dump_dest='/u01/app/oracle/admin/stdrac/cdump'
 nofilenamecheck;
}
```

23. Verify the instance is running fine.

```
sqlplus / as sysdba

SELECT NAME, DB_UNIQUE_NAME, DATABASE_ROLE, OPEN_MODE FROM GV$DATABASE;

COL DATAFILE FORMAT A50
SELECT INST_ID, NAME DATAFILE FROM GV$DATAFILE;

SELECT INST_ID,GROUP#,THREAD#,SEQUENCE#,ARCHIVED,STATUS FROM GV$STANDBY_LOG ORDER BY
1,2,3;

# you will notice the SRL members within the groups got multiplexed:
SELECT TYPE, count(*) FROM GV$LOGFILE GROUP BY TYPE;
```

24. Remove the multiplexed copy of each SRL group on the standby.

```
COLUMN MEMBER FORMAT A50
SELECT GROUP#, MEMBER FROM V$LOGFILE WHERE TYPE='STANDBY' ORDER BY 1,2;

# drop all the members in DATA diskgroup:
# use the following query to generate drop statements:
set linesize 150
set pagesize 22
select 'ALTER DATABASE DROP STANDBY LOGFILE MEMBER ''' || MEMBER || ''';'
from V$LOGFILE WHERE TYPE='STANDBY' and member like '+DATA%';
```

Note: you will receive ORA-00261 error when dropping some standby redo log files because the standby redo log on concern is being used by the transport service. To drop it, on the **primary** database **nodes**, switch the current logfile by running the following command a few times in each node.

```
alter system switch logfile;
```

Verify the SRL group multiplexed members were removed:

```
SELECT GROUP#, MEMBER FROM V$LOGFILE WHERE TYPE='STANDBY' ORDER BY 1,2;
```

25. Set the necessary initialization parameters for the standby database `stdrac`.

```
ALTER SYSTEM SET FAL_SERVER=rac SCOPE=BOTH SID='*';
ALTER SYSTEM SET FAL_CLIENT=stdrac SCOPE=BOTH SID='*';
ALTER SYSTEM SET LOG_ARCHIVE_CONFIG='DG_CONFIG=(rac,stdrac)' SID='*';
ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=AUTO SID='*';
ALTER SYSTEM SET LOG_ARCHIVE_DEST_2='SERVICE=rac ASYNC DB_UNIQUE_NAME=rac
VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)' SCOPE=BOTH SID='*';
```

26. As `oracle` in `srva`, add the standby database resource in the Clusterware OCR

```
srvctl add database -d stdrac -o $ORACLE_HOME -role physical_standby -startoption mount -
diskgroup DATA,FRA
srvctl add instance -d stdrac -i stdrac1 -n srva.localdomain
srvctl add instance -d stdrac -i stdrac2 -n srvb.localdomain
```

27. Relocate the SPFILE used by the standby database from the default file system directory to ASM.

    a. Create a copy of the initialization parameter file in ASM

```
sqlplus / as sysdba
show parameter SPFILE
CREATE PFILE FROM SPFILE;

CREATE SPFILE='+DATA/STDRAC/PARAMETERFILE/spfilestdrac.ora' From PFILE;
```

    b. Modify PFILE to use new SPFILE in ASM

```
echo "SPFILE='+DATA/STDRAC/PARAMETERFILE/spfilestdrac.ora'" >
$ORACLE_HOME/dbs/initstdrac1.ora

rm $ORACLE_HOME/dbs/spfilestdrac1.ora

ssh srvb "echo \"SPFILE='+DATA/STDRAC/PARAMETERFILE/spfilestdrac.ora'\" >
$ORACLE_HOME/dbs/initstdrac2.ora"
ssh srvb "cat $ORACLE_HOME/dbs/initstdrac2.ora"
```

    c.   Update OCR with the new SPFILE location -

```
srvctl modify database -d stdrac -p +DATA/STDRAC/PARAMETERFILE/spfilestdrac.ora
```

    d.   Restart the standby instance

```
srvctl status instance -d stdrac -i stdrac1
srvctl start instance -d stdrac -i stdrac1
srvctl status database -d stdrac
```

    e.   Connect to the database and verify that the SPFILE parameter has been successfully set

```
show parameter SPFILE
```

28. Set the correct INSTANCE_NAME and INSTANCE_NUMBER for the instances in SPFILE

```
ALTER SYSTEM SET INSTANCE_NUMBER=1 SCOPE=SPFILE SID='stdrac1';
ALTER SYSTEM SET INSTANCE_NUMBER=2 SCOPE=SPFILE SID='stdrac2';
ALTER SYSTEM SET INSTANCE_NAME='stdrac1' SCOPE=SPFILE SID='stdrac1';
ALTER SYSTEM SET INSTANCE_NAME='stdrac2' SCOPE=SPFILE SID='stdrac2';
```

29. Start the second instance in the standby database.

```
srvctl start instance -d stdrac -i stdrac2
srvctl status database -d stdrac
```

30. Start the apply process in the standby database.

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;
```

31. Check out the Data Guard configuration in each database:

```
SELECT INST_ID, DATABASE_ROLE, DB_UNIQUE_NAME INSTANCE, OPEN_MODE, PROTECTION_MODE,
PROTECTION_LEVEL FROM GV$DATABASE;

SELECT INST_ID, DATABASE_MODE, RECOVERY_MODE, PROTECTION_MODE
FROM GV$ARCHIVE_DEST_STATUS
WHERE DEST_NAME ='LOG_ARCHIVE_DEST_1';
```

32. Check on the redo transport services.

    a. On the standby, list the SRL files:

```
ALTER SESSION SET NLS_DATE_FORMAT ='DD-MON-RR HH24:MI:SS';
SELECT INST_ID, SEQUENCE#, APPLIED, FIRST_TIME, NEXT_TIME
FROM GV$ARCHIVED_LOG ORDER BY 2,1,4;
```

    b. On the primary database, switch the current logfile :

```
ALTER SYSTEM SWITCH LOGFILE;
```

    c. On the standby database, list the SRL file again and observe the changes in the returns rows:

```
SELECT INST_ID, SEQUENCE#, APPLIED, FIRST_TIME, NEXT_TIME
FROM GV$ARCHIVED_LOG ORDER BY 2,1,4;
```

33. In the standby database, retrieve Data Guard statistics below.

```
set linesize 120
col START_TIME format a20
col ITEM format a20

SELECT TO_CHAR(START_TIME, 'DD-MON-RR HH24:MI:SS') START_TIME, ITEM , SOFAR, UNITS
FROM V$RECOVERY_PROGRESS
WHERE ITEM IN ('Active Apply Rate', 'Average Apply Rate', 'Redo Applied');
```

34. In the standby database, retrieve the transport lag and the apply lag. Transport lag represents the data that will be lost in case of disaster.

```
col NAME for a13
col VALUE for a13
col UNIT for a30
set LINES 132

SELECT NAME, VALUE, UNIT, TIME_COMPUTED
FROM V$DATAGUARD_STATS WHERE NAME IN ('transport lag', 'apply lag');
```

35. In the standby system, check the state of the standby database instance resources.

You should see their states are "INTERMEDIATE".

```
su - grid
crsctl stat res -t
```

Post Creation Steps

36. Configure `ARCHIVELOG DELETION POLICY` in RMAN in both systems

```
rman target /
CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;
```

37. On the standby database, enable the Flashback Database:

```
sqlplus sys/oracle@stdrac as sysdba
SELECT LOG_MODE,FLASHBACK_ON FROM V$DATABASE;
ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET =2880;
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
ALTER DATABASE FLASHBACK ON;
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;
```

Testing the Data Guard Functionality

38. On the primary database, create a new tablespace.

```
CREATE TABLESPACE TEST;
```

39. Check out the tablespace in the standby database. Connect to each instance and check out the tablespace.

```
SELECT NAME FROM V$TABLESPACE WHERE NAME ='TEST';
SELECT NAME FROM V$DATAFILE WHERE TS# IN (SELECT TS# FROM V$TABLESPACE WHERE NAME ='TEST');
```

Next Steps

In real life scenario, after creating a physical standby database, consider implementing the following points:

- Configure the Data Guard Broker

- Consider the need to change the protection mode

- Put on a mechanism to purge the archived redo logs on the standby