Excellent — you’re asking for a **real-world advanced Oracle 19c Data Guard with RAC** setup.  
You have:

* **Primary:** 2-node RAC with ASM (+DATA, +FLASH) **already running**.
* **Goal:** Set up **three** 2-node RAC standby databases, each with ASM.

This is a **big ask**, so I’ll give you a **full professional-grade plan**, with a **clear structure**:

# 🏛 High-Level Architecture:

|  | **PRIMARY SITE** | **STANDBY 1** | **STANDBY 2** | **STANDBY 3** |
| --- | --- | --- | --- | --- |
| Cluster | RAC (2 nodes) | RAC (2 nodes) | RAC (2 nodes) | RAC (2 nodes) |
| ASM | +DATA, +FLASH | +DATA, +FLASH | +DATA, +FLASH | +DATA, +FLASH |
| DB Mode | ARCHIVELOG | ARCHIVELOG | ARCHIVELOG | ARCHIVELOG |
| Transport | SYNC/ASYNC depending on distance | ASYNC | ASYNC | ASYNC |
| Broker | Will configure with **DGMGRL** |  |  |  |

# 📜 ****Full Step-by-Step Setup****

# 1. ****Pre-Requisites**** (Primary RAC and all 3 Standby RACs)

✔️ Software Installations:

* Oracle Grid Infrastructure 19c — installed and clustered (CRS)
* Oracle RDBMS 19c — installed

✔️ ASM Setup:

* Disk groups created: +DATA (for database files) and +FLASH (for FRA, archivelogs)

✔️ Network:

* Public, Private (Interconnect), and VIP addresses configured.
* SCAN listener working.

✔️ Configure Static Listener (Mandatory for Duplicate): On all nodes (Primary + Standbys) **add static entry** in listener.ora:

bash

CopyEdit

SID\_LIST\_LISTENER =

(SID\_LIST =

(SID\_DESC =

(GLOBAL\_DBNAME = yourdb\_DGMGRL)

(ORACLE\_HOME = /u01/app/oracle/product/19.0.0/dbhome\_1)

(SID\_NAME = yourdb1) # on node 1

)

(SID\_DESC =

(GLOBAL\_DBNAME = yourdb\_DGMGRL)

(ORACLE\_HOME = /u01/app/oracle/product/19.0.0/dbhome\_1)

(SID\_NAME = yourdb2) # on node 2

)

)

Reload listener:

bash

CopyEdit

lsnrctl reload

# 2. ****Primary Database Configuration****

✔️ Enable Forced Logging:

sql

CopyEdit

ALTER DATABASE FORCE LOGGING;

✔️ Check ARCHIVELOG mode:

sql

CopyEdit

ARCHIVE LOG LIST;

If not in archivelog, enable it (you said primary is ready, so probably done).

✔️ Create Standby Redo Logs (SRLs):

Must match (or be higher than) the number of primary online redo logs × number of standby sites. Example:

sql

CopyEdit

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 500M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 500M;

-- Repeat so each thread has 4+ SRLs

**Tip:** SRLs must be same size as online redo logs!

✔️ TNS entries (tnsnames.ora): Make entries for all databases — primary and all three standbys.

Example for primary tnsnames.ora:

bash

CopyEdit

PRIMDB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = prim-scan.example.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = primdb)

)

)

STBY1DB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = stby1-scan.example.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = stby1db)

)

)

-- similarly for STBY2DB, STBY3DB

✔️ Configure LOG\_ARCHIVE\_CONFIG:

sql

CopyEdit

ALTER SYSTEM SET LOG\_ARCHIVE\_CONFIG='DG\_CONFIG=(PRIMDB,STBY1DB,STBY2DB,STBY3DB)';

✔️ Setup LOG\_ARCHIVE\_DEST\_n for each standby:

sql

CopyEdit

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_2='SERVICE=STBY1DB ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=stby1db';

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_3='SERVICE=STBY2DB ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=stby2db';

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_4='SERVICE=STBY3DB ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=stby3db';

✔️ Enable Flashback Database (recommended):

sql

CopyEdit

ALTER DATABASE FLASHBACK ON;

✔️ Create Password File:

Make sure passwordless authentication is configured using orapwd.

bash

CopyEdit

orapwd file=$ORACLE\_HOME/dbs/orapwprimdb password=Oracle123 entries=10 force=y format=12

Copy password file to all standby nodes' dbs directory.

# 3. ****Prepare Standby Databases****

On **all standby clusters**, prepare:

✔️ Create necessary directory structures:

bash

CopyEdit

mkdir -p /u01/app/oracle/admin/stby1db/adump

mkdir -p /u01/app/oracle/oradata/stby1db

(Use +DATA in ASM, so adjust accordingly.)

✔️ Create the ASM directories:

sql

CopyEdit

ALTER DISKGROUP DATA ADD DIRECTORY '+DATA/STBY1DB';

ALTER DISKGROUP FLASH ADD DIRECTORY '+FLASH/STBY1DB';

✔️ Prepare an empty init.ora file (temporary for RMAN duplicate):

On first node (node1) of each standby:

Example /u01/app/oracle/product/19.0.0/dbhome\_1/dbs/initstby1db1.ora

bash

CopyEdit

DB\_NAME=primdb

DB\_UNIQUE\_NAME=stby1db

INSTANCE\_NUMBER=1

CLUSTER\_DATABASE=TRUE

REMOTE\_LOGIN\_PASSWORDFILE=EXCLUSIVE

FAL\_SERVER=primdb

FAL\_CLIENT=stby1db

DB\_FILE\_NAME\_CONVERT='+DATA/PRIMDB/','+DATA/STBY1DB/'

LOG\_FILE\_NAME\_CONVERT='+FLASH/PRIMDB/','+FLASH/STBY1DB/','+DATA/PRIMDB/','+DATA/STBY1DB/'

STANDBY\_FILE\_MANAGEMENT=AUTO

✔️ Create SPFILE manually from init.ora (after duplication) or modify later.

# 4. ****RMAN Active Duplicate****

Login to **Primary node** and perform duplicate for each standby one by one.

Example for Standby1:

bash

CopyEdit

rman target sys/Oracle123@primdb auxiliary sys/Oracle123@stby1db

DUPLICATE TARGET DATABASE FOR STANDBY

FROM ACTIVE DATABASE

DORECOVER

SPFILE

SET DB\_UNIQUE\_NAME='stby1db'

SET INSTANCE\_NUMBER='1'

SET DB\_FILE\_NAME\_CONVERT='+DATA/PRIMDB/','+DATA/STBY1DB/'

SET LOG\_FILE\_NAME\_CONVERT='+FLASH/PRIMDB/','+FLASH/STBY1DB/','+DATA/PRIMDB/','+DATA/STBY1DB/'

SET FAL\_SERVER='primdb'

SET FAL\_CLIENT='stby1db'

SET STANDBY\_FILE\_MANAGEMENT='AUTO'

NOFILENAMECHECK;

Repeat the **RMAN DUPLICATE** for Standby2 and Standby3.

# 5. ****Register Instances to CRS (srvctl)****

Once the duplicate is successful on standby clusters:

On each standby node:

bash

CopyEdit

srvctl add database -d stby1db -o /u01/app/oracle/product/19.0.0/dbhome\_1 -p +DATA/STBY1DB/PARAMETERFILE/spfilestby1db.ora -r PHYSICAL\_STANDBY -n stby1db1,stby1db2

srvctl add instance -d stby1db -i stby1db1 -n node1

srvctl add instance -d stby1db -i stby1db2 -n node2

Repeat for stby2db, stby3db.

Start all standby databases:

bash

CopyEdit

srvctl start database -d stby1db

srvctl start database -d stby2db

srvctl start database -d stby3db

# 6. ****Enable Managed Recovery on Standby****

Connect to each standby DB and:

sql

CopyEdit

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;

Optionally for Active Data Guard (if you want to open read-only while applying):

sql

CopyEdit

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;

ALTER DATABASE OPEN;

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT;

# 7. ****(Optional but Recommended) Configure DGMGRL Broker****

Enable Broker on all databases:

sql

CopyEdit

ALTER SYSTEM SET DG\_BROKER\_START=TRUE SCOPE=BOTH;

Create Broker configuration:

bash

CopyEdit

dgmgrl sys/Oracle123@primdb

CREATE CONFIGURATION 'DG\_CONFIG' AS PRIMARY DATABASE IS 'primdb' CONNECT IDENTIFIER IS 'primdb';

ADD DATABASE 'stby1db' AS CONNECT IDENTIFIER IS 'stby1db' MAINTAINED AS PHYSICAL;

ADD DATABASE 'stby2db' AS CONNECT IDENTIFIER IS 'stby2db' MAINTAINED AS PHYSICAL;

ADD DATABASE 'stby3db' AS CONNECT IDENTIFIER IS 'stby3db' MAINTAINED AS PHYSICAL;

ENABLE CONFIGURATION;

Check status:

bash

CopyEdit

SHOW CONFIGURATION;

SHOW DATABASE verbose primdb;

# 🚩 Final Checks:

| **Checkpoint** | **Command** |
| --- | --- |
| Archive shipping | select sequence#,applied from v$archived\_log order by sequence#; |
| Managed recovery running | select process, status from v$managed\_standby; |
| DGMGRL health | show configuration; |

# ⚡ Quick Summary:

✔️ Primary RAC 2-node ready.  
✔️ Three Standby RACs (each 2-node) built with **RMAN Active Duplicate**.  
✔️ ASM with +DATA and +FLASH used everywhere.  
✔️ Static listeners needed.  
✔️ SRLs created.  
✔️ TNS entries maintained.  
✔️ Data Guard Broker optionally configured.

# 📢 Important Recommendations:

* Use **ASYNC** transport for remote sites.
* Always enable **Flashback** if possible.
* Monitor via **OEM 13c**.
* Plan **switchover/failover drills** periodically.

Would you like me to also prepare ready-to-use **scripts** (like RMAN duplicate script, DGMGRL commands, srvctl commands) that you can just plug and run? 🚀  
(If yes, I'll prepare a **real copy-paste practical set** for you!) 🔥

**Would you like me to continue and give you the ready-to-go full script set as well?** 🎯

4o

##### You said:

yes please

###### ChatGPT said:

Perfect!  
Since you're doing a **serious, production-grade** multi-standby RAC setup, I'll give you **FULL, ready-to-run scripts**.  
You can adjust minor things like DB name, paths, SCAN hostnames as per your environment.

Here we go! 🚀

# 🛠️ ****Script 1: Primary Database Preparation****

**Run on Primary RAC DB (one node):**

sql

CopyEdit

-- Force Logging

ALTER DATABASE FORCE LOGGING;

-- Flashback (Optional but Recommended)

ALTER DATABASE FLASHBACK ON;

-- Standby Redo Logs (minimum 4 per thread recommended)

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 500M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 500M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 500M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 500M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 500M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 500M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 500M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 500M;

-- Log Archive Config

ALTER SYSTEM SET LOG\_ARCHIVE\_CONFIG='DG\_CONFIG=(PRIMDB,STBY1DB,STBY2DB,STBY3DB)' SCOPE=BOTH;

-- Destination for each standby

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_2='SERVICE=stby1db ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=stby1db' SCOPE=BOTH;

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_3='SERVICE=stby2db ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=stby2db' SCOPE=BOTH;

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_4='SERVICE=stby3db ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=stby3db' SCOPE=BOTH;

-- Standby file management

ALTER SYSTEM SET STANDBY\_FILE\_MANAGEMENT=AUTO SCOPE=BOTH;

# 🛠️ ****Script 2: Listener Static Entry****

**listener.ora** — Add static entry on **ALL nodes (Primary + Standbys):**

bash

CopyEdit

SID\_LIST\_LISTENER =

(SID\_LIST =

(SID\_DESC =

(GLOBAL\_DBNAME = primdb\_DGMGRL)

(ORACLE\_HOME = /u01/app/oracle/product/19.0.0/dbhome\_1)

(SID\_NAME = primdb1)

)

(SID\_DESC =

(GLOBAL\_DBNAME = primdb\_DGMGRL)

(ORACLE\_HOME = /u01/app/oracle/product/19.0.0/dbhome\_1)

(SID\_NAME = primdb2)

)

)

(Similarly on standby for stby1db1, stby1db2 etc.)

👉 Then reload:

bash

CopyEdit

lsnrctl reload

# 🛠️ ****Script 3: TNSNAMES Entries****

Update **$ORACLE\_HOME/network/admin/tnsnames.ora** on ALL nodes:

bash

CopyEdit

PRIMDB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = prim-scan.example.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = primdb)

)

)

STBY1DB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = stby1-scan.example.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = stby1db)

)

)

STBY2DB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = stby2-scan.example.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = stby2db)

)

)

STBY3DB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = stby3-scan.example.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = stby3db)

)

)

# 🛠️ ****Script 4: RMAN Active Duplicate Standby 1****

**From Primary Node, run:**

bash

CopyEdit

rman target sys/Oracle123@primdb auxiliary sys/Oracle123@stby1db

DUPLICATE TARGET DATABASE

FOR STANDBY

FROM ACTIVE DATABASE

DORECOVER

SPFILE

SET DB\_UNIQUE\_NAME='stby1db'

SET INSTANCE\_NUMBER='1'

SET DB\_FILE\_NAME\_CONVERT='+DATA/PRIMDB/','+DATA/STBY1DB/'

SET LOG\_FILE\_NAME\_CONVERT='+FLASH/PRIMDB/','+FLASH/STBY1DB/','+DATA/PRIMDB/','+DATA/STBY1DB/'

SET FAL\_SERVER='primdb'

SET FAL\_CLIENT='stby1db'

SET STANDBY\_FILE\_MANAGEMENT='AUTO'

NOFILENAMECHECK;

✅ Repeat similarly for Standby2 (stby2db) and Standby3 (stby3db)!

Just change DB\_UNIQUE\_NAME and FAL\_CLIENT entries accordingly.

# 🛠️ ****Script 5: Add Standby DB to CRS (srvctl)****

After duplicate succeeds, register database with CRS on **standby clusters**:

bash

CopyEdit

srvctl add database -d stby1db -o /u01/app/oracle/product/19.0.0/dbhome\_1 -p +DATA/STBY1DB/PARAMETERFILE/spfilestby1db.ora -r PHYSICAL\_STANDBY -n stby1db1,stby1db2

srvctl add instance -d stby1db -i stby1db1 -n node1

srvctl add instance -d stby1db -i stby1db2 -n node2

srvctl start database -d stby1db

✅ Repeat similarly for stby2db, stby3db.

# 🛠️ ****Script 6: Start Managed Recovery****

On each standby:

sql

CopyEdit

-- SQLPLUS login

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT;

(If you want Active Data Guard — open database read-only while applying — add open steps.)

# 🛠️ ****Script 7: Data Guard Broker (DGMGRL)****

Enable DGMGRL on all (primary and standby):

sql

CopyEdit

ALTER SYSTEM SET DG\_BROKER\_START=TRUE SCOPE=BOTH;

Then from DGMGRL prompt:

bash

CopyEdit

dgmgrl sys/Oracle123@primdb

CREATE CONFIGURATION 'DG\_CONFIG' AS PRIMARY DATABASE IS 'primdb' CONNECT IDENTIFIER IS 'primdb';

ADD DATABASE 'stby1db' AS CONNECT IDENTIFIER IS 'stby1db' MAINTAINED AS PHYSICAL;

ADD DATABASE 'stby2db' AS CONNECT IDENTIFIER IS 'stby2db' MAINTAINED AS PHYSICAL;

ADD DATABASE 'stby3db' AS CONNECT IDENTIFIER IS 'stby3db' MAINTAINED AS PHYSICAL;

ENABLE CONFIGURATION;

SHOW CONFIGURATION;

# ✅ ****After Success, Validate with:****

bash

CopyEdit

-- DGMGRL

show configuration;

show database verbose primdb;

show database verbose stby1db;

show database verbose stby2db;

show database verbose stby3db;

-- SQLPLUS

select database\_role, open\_mode from v$database;

select process, status, thread#, sequence# from v$managed\_standby;

# 🎯 Notes:

| **Key Point** | **Tip** |
| --- | --- |
| Duplicate via RMAN | Fastest, Active no downtime |
| Broker (DGMGRL) | Makes switchover/failover 1 command |
| Diskgroups | +DATA for datafiles, +FLASH for FRA/Archives |
| Static Listeners | Mandatory for duplicate |
| TNS Setup | Must exist before duplicate |
| Monitoring | Use OEM 13c or v$dataguard\_stats |

# 📋 Quick Table: What To Replace In Scripts

| **Placeholder** | **Replace With** |
| --- | --- |
| prim-scan.example.com | Your Primary RAC SCAN VIP |
| stby1-scan.example.com | Standby1 RAC SCAN VIP |
| stby2-scan.example.com | Standby2 RAC SCAN VIP |
| stby3-scan.example.com | Standby3 RAC SCAN VIP |
| Oracle123 | SYS password |
| +DATA/PRIMDB/ | ASM paths for data |
| +FLASH/PRIMDB/ | ASM paths for archive logs |

# ✨ ****All ready!****

If you follow above **properly**, you'll have a **full production-grade Oracle 19c 2-node RAC Primary + 3 x 2-node RAC Standbys** setup fully working.

Would you also like me to prepare:

* A **switchover and failover commands set** ready-to-run via DGMGRL?
* **Health-check SQL queries** to regularly monitor your primary + standbys?

👉 I can prepare those too if you want! 🚀  
(Just say "Yes, give me switchover + healthcheck scripts too") 🔥

4o

##### You said:

"Yes, give me switchover + healthcheck scripts too

###### ChatGPT said:

Awesome! 🔥  
You’re building a **true enterprise-class** Data Guard setup — let's make it full-proof!  
Here’s what I’ll give you next:

* **Part 1:** Switchover Steps (using DGMGRL)
* **Part 2:** Health Check Scripts (SQL and DGMGRL)
* **Bonus:** Easy Auto-Check Script you can schedule via cron/job!

# 🛠️ ****Part 1: Switchover Steps (Primary <-> Standby)****

✅ **Important Preparation before Switchover:**

From **any server** where DGMGRL is configured:

bash

CopyEdit

dgmgrl sys/Oracle123@primdb

### 🔵 Check Broker Status

bash

CopyEdit

SHOW CONFIGURATION;

SHOW DATABASE VERBOSE primdb;

SHOW DATABASE VERBOSE stby1db;

👉 All must show: SUCCESS  
If any show Warning, **fix first**.

✅ **Actual Switchover Steps**

## 1️⃣ Switchover Primary ➔ Standby1

bash

CopyEdit

-- Connect DGMGRL

dgmgrl sys/Oracle123@primdb

-- Switchover Command

SWITCHOVER TO stby1db;

-- (DGMGRL will automatically change roles and restart DBs)

-- Verify

SHOW CONFIGURATION;

SHOW DATABASE VERBOSE stby1db;

SHOW DATABASE VERBOSE primdb;

✨ Now stby1db is **new Primary**, primdb becomes **new Standby**.

✅ **Post Switchover Cleanup:**

You should now:

* Update TAF services if needed (optional)
* Check replication lag
* Monitor for few mins

## 2️⃣ Switchover Back Standby1 ➔ Primary

From **stby1db** node (now acting as Primary):

bash

CopyEdit

dgmgrl sys/Oracle123@stby1db

SWITCHOVER TO primdb;

SHOW CONFIGURATION;

✅ Done!

✅ **If needed to Switchover to Standby2 or Standby3:**  
Just change to stby2db or stby3db in above.

bash

CopyEdit

SWITCHOVER TO stby2db;

# 🛠️ ****Part 2: Health Check Scripts****

You should regularly run these:

## 🔵 1. Quick SQL Check from SQL\*Plus:

sql

CopyEdit

COL DATABASE\_ROLE FOR A20

COL OPEN\_MODE FOR A20

SELECT NAME, DATABASE\_ROLE, OPEN\_MODE, PROTECTION\_MODE, SWITCHOVER\_STATUS FROM V$DATABASE;

## 🔵 2. Managed Recovery Process Status

sql

CopyEdit

SELECT PROCESS, STATUS, THREAD#, SEQUENCE# FROM V$MANAGED\_STANDBY ORDER BY THREAD#, PROCESS;

👉 MRP0 should be running on standby.

## 🔵 3. Lag Monitoring (Apply / Transport Lag)

sql

CopyEdit

SELECT NAME, VALUE, UNIT FROM V$DATAGUARD\_STATS WHERE NAME IN ('transport lag', 'apply lag');

✅ Should show minimal lag (seconds only under normal conditions).

## 🔵 4. DGMGRL Configuration Health Check

bash

CopyEdit

dgmgrl sys/Oracle123@primdb

SHOW CONFIGURATION;

SHOW DATABASE VERBOSE primdb;

SHOW DATABASE VERBOSE stby1db;

SHOW DATABASE VERBOSE stby2db;

SHOW DATABASE VERBOSE stby3db;

✅ All should show SUCCESS and no errors.

# 🎁 ****Bonus: Scheduled Health Check Script****

You can put this in a simple shell script to **auto-email** status daily!

**Example: /home/oracle/dg\_healthcheck.sh**

bash

CopyEdit

#!/bin/bash

export ORACLE\_HOME=/u01/app/oracle/product/19.0.0/dbhome\_1

export PATH=$ORACLE\_HOME/bin:$PATH

export ORACLE\_SID=primdb1

LOGFILE=/tmp/dg\_healthcheck\_$(date +%F).log

dgmgrl sys/Oracle123@primdb <<EOF > $LOGFILE

show configuration;

show database verbose primdb;

show database verbose stby1db;

show database verbose stby2db;

show database verbose stby3db;

exit

EOF

mail -s "Daily Data Guard Health Check $(date +%F)" yourteam@example.com < $LOGFILE

✅ Schedule via cron:

bash

CopyEdit

crontab -e

0 7 \* \* \* /home/oracle/dg\_healthcheck.sh

(Adjust ORACLE\_HOME/ORACLE\_SID/path accordingly.)

# 🚀 Now you have:

* Switchover made super simple
* Full health check SQL and shell scripts
* Cron scheduling to auto-report every morning

# ⚡ Final Tips:

| **Task** | **Command** |
| --- | --- |
| Force Role Change (Force Failover) | FAILOVER TO stby1db; (use with care) |
| Reinstate Failed Primary after Failover | REINSTATE DATABASE primdb; |
| Broker Logs | $ORACLE\_HOME/rdbms/log/drc\*.log |
| Check Redo Apply LAG | v$dataguard\_stats |
| Monitor from OEM 13c | DG monitoring available |

# 🎯 Conclusion

With these in place, your setup will be **rock-solid** and **production-ready**.  
You’ll be ready for any switchover or DR drill at any time confidently. 😎

Would you also like:

* **"Failover full script" (if primary crashes)?**
* **"Reinstate failed primary after failover" steps?**  
  👉 I can give that ready too if you want!

(Just say: "Yes, give me failover + reinstate scripts too!") 🔥

4o

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

Excellent, you are asking a very **real-world DBA scenario**:

"**Primary RAC 2-node** (already running) ➔ Setup **three Standby RAC 2-node** (each with ASM storage) using Oracle 19c Data Guard."

This is a **serious enterprise-level setup** — usually done with **Active Data Guard** or **Standard Physical Standby**.

I’ll explain this step-by-step very systematically assuming:

* **Primary RAC 2-node**: Already running (CDB+PDB or Non-CDB).
* **All nodes (Primary + Standby) use ASM** for storage.
* **Oracle Grid Infrastructure and RDBMS binaries are installed on standby clusters.**
* **All networking (SCAN, VIP, public, private interconnect)** is properly set.
* **Data Guard Broker (DGMGRL)** will be used later for management.
* **Listeners are running on all nodes**.

**🔥 Master Plan for Setup**

**1. Prerequisites on Standby Sites**

✔️ **Install Oracle Grid Infrastructure 19c** (for RAC) — if not already installed.  
✔️ **Install Oracle RDBMS 19c binaries** (software-only install).  
✔️ **Setup ASM Diskgroups** (+DATA, +FRA) matching primary diskgroup structure.  
✔️ **Create necessary directories** (like dbs, admin, adump) under ORACLE\_HOME.  
✔️ **Ensure Network** (public/private/SCAN VIP) is working properly.

**2. Configure Password File for Standby RAC nodes**

🔹 From primary (node1), copy the password file to standby nodes:

bash

CopyEdit

$ cd $ORACLE\_HOME/dbs

$ scp orapw<DB\_UNIQUE\_NAME> standby1\_node1:$ORACLE\_HOME/dbs/

$ scp orapw<DB\_UNIQUE\_NAME> standby2\_node1:$ORACLE\_HOME/dbs/

$ scp orapw<DB\_UNIQUE\_NAME> standby3\_node1:$ORACLE\_HOME/dbs/

✅ Ensure REMOTE\_LOGIN\_PASSWORDFILE=EXCLUSIVE is set.

**3. Configure listener.ora and tnsnames.ora on ALL nodes**

🔹 tnsnames.ora example (same file across primary and all standby nodes):

text

CopyEdit

PRIM\_DB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = prim-scan)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = primdb)

)

)

STDBY1\_DB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = standby1-scan)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = standby1db)

)

)

STDBY2\_DB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = standby2-scan)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = standby2db)

)

)

STDBY3\_DB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = standby3-scan)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = standby3db)

)

)

✅ Test connectivity using:

bash

CopyEdit

$ tnsping PRIM\_DB

$ tnsping STDBY1\_DB

$ tnsping STDBY2\_DB

$ tnsping STDBY3\_DB

**4. Enable FORCE LOGGING on Primary**

Connect to **Primary Database**:

sql

CopyEdit

ALTER DATABASE FORCE LOGGING;

🔹 Verify:

sql

CopyEdit

SELECT FORCE\_LOGGING FROM V$DATABASE;

**5. Enable Archivelog Mode (if not already enabled)**

On **Primary**:

bash

CopyEdit

$ srvctl stop database -d <db\_unique\_name>

$ sqlplus / as sysdba

STARTUP MOUNT;

ALTER DATABASE ARCHIVELOG;

ALTER DATABASE OPEN;

✅ Verify:

sql

CopyEdit

ARCHIVE LOG LIST;

**6. Configure Primary for Standby Redo Log Groups**

* Add **standby redo logs** (must be **at least** equal to number of online redo groups × number of RAC instances + 1).
* For RAC, you need thread-specific SRLs.

Example:

sql

CopyEdit

-- For thread 1

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 11 ('+DATA') SIZE 500M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 12 ('+DATA') SIZE 500M;

-- For thread 2

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 GROUP 21 ('+DATA') SIZE 500M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 GROUP 22 ('+DATA') SIZE 500M;

**7. Prepare Initialization Parameter File for Standbys**

🔹 Create a PFILE from SPFILE:

sql

CopyEdit

CREATE PFILE='/tmp/initprimdb.ora' FROM SPFILE;

🔹 Modify parameters (to create **standby-specific** parameters):

Mandatory parameters to change:

text

CopyEdit

DB\_NAME=primdb -- SAME

DB\_UNIQUE\_NAME=standby1db -- DIFFERENT FOR EACH STANDBY

LOG\_ARCHIVE\_CONFIG='DG\_CONFIG=(primdb,standby1db,standby2db,standby3db)'

LOG\_ARCHIVE\_DEST\_1='LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=standby1db'

LOG\_ARCHIVE\_DEST\_2='SERVICE=primdb LGWR ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=primdb'

FAL\_SERVER=primdb

FAL\_CLIENT=standby1db

DB\_FILE\_NAME\_CONVERT='/primdb/','/standby1db/' -- Adjust path names if different

LOG\_FILE\_NAME\_CONVERT='/primdb/','/standby1db/'

STANDBY\_FILE\_MANAGEMENT=AUTO

✅ Similarly prepare for standby2db and standby3db (modify DB\_UNIQUE\_NAME and FAL\_CLIENT).

**8. Create Directories on Standby ASM for Datafiles/Redo**

Use **ASMCMD** or SQL to create directories if needed:

bash

CopyEdit

$ asmcmd

ASMCMD> mkdir +DATA/STDBY1DB/DATAFILE

ASMCMD> mkdir +DATA/STDBY1DB/ONLINELOG

ASMCMD> mkdir +FRA/STDBY1DB

**9. Start NOMOUNT Stage on All Standby Instances**

On each standby node:

bash

CopyEdit

export ORACLE\_SID=<instance\_name>

sqlplus / as sysdba

STARTUP NOMOUNT PFILE='/tmp/initstandby1db.ora';

**10. RMAN Duplicate (Active) for each Standby RAC**

From standby1 Node1:

bash

CopyEdit

rman target sys@primdb auxiliary sys@standby1db

RMAN> DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE

DORECOVER

SPFILE

SET DB\_UNIQUE\_NAME='standby1db'

SET LOG\_FILE\_NAME\_CONVERT='/primdb/','/standby1db/'

SET DB\_FILE\_NAME\_CONVERT='/primdb/','/standby1db/'

NOFILENAMECHECK;

✅ Similarly for standby2db and standby3db.

**11. Convert Standby to RAC**

After duplicate is done:

* Create instance-specific parameter files and add them to CRS:

bash

CopyEdit

srvctl add database -d standby1db -o $ORACLE\_HOME -p +DATA/STDBY1DB/PARAMETERFILE/spfile.ora -r PHYSICAL\_STANDBY

srvctl add instance -d standby1db -i standby1db1 -n standby1-node1

srvctl add instance -d standby1db -i standby1db2 -n standby1-node2

srvctl start database -d standby1db

✅ Do for standby2db and standby3db as well.

**12. Register All Standbys in Data Guard Broker (optional but recommended)**

Enable Broker on all DBs:

sql

CopyEdit

ALTER SYSTEM SET DG\_BROKER\_START=TRUE;

Connect using dgmgrl:

bash

CopyEdit

dgmgrl sys@primdb

DGMGRL> CREATE CONFIGURATION 'DGConfig' AS PRIMARY DATABASE IS 'primdb' CONNECT IDENTIFIER IS 'primdb';

DGMGRL> ADD DATABASE 'standby1db' AS CONNECT IDENTIFIER IS 'standby1db' MAINTAINED AS PHYSICAL;

DGMGRL> ADD DATABASE 'standby2db' AS CONNECT IDENTIFIER IS 'standby2db' MAINTAINED AS PHYSICAL;

DGMGRL> ADD DATABASE 'standby3db' AS CONNECT IDENTIFIER IS 'standby3db' MAINTAINED AS PHYSICAL;

DGMGRL> ENABLE CONFIGURATION;

**13. Final Verification**

✅ Validate by:

sql

CopyEdit

SELECT DATABASE\_ROLE, OPEN\_MODE FROM V$DATABASE;