

General guide

Introduction

Modern enterprises demand high availability (HA), scalability, and disaster recovery (DR) capabilities from their database systems.

Oracle Database, particularly when deployed with Real Application Clusters (RAC) and Data Guard, provides a comprehensive suite of features to address these requirements.

This internship project simulated a complete Oracle environment in a virtualized setup, covering system preparation, clustered database deployment, and disaster recovery configuration.

The work was conducted entirely on a personal laptop using Oracle VirtualBox and interconnected RHEL 7.8 virtual machines, creating a cost-effective yet realistic environment to explore advanced Oracle Database Administration (DBA) concepts. The project can be divided into three major phases: VM setup, RAC installation and DataGuard Installation

Content Table

Table des matières

Introduction	1
Content Table	2
Virtual Machine Setup	3
Create Virtual Machines (VMs)	3
Operating System Configuration:	4
RAC and Database Installation Guide	8
Pre-Install and Kernel Parameters	8
Storage Setup	8
Software Preparation	8
Install Grid Infrastructure (Clusterware)	9
Install Oracle Database Home	9
Create ASM Disk Group	9
Create the RAC Database	10
Oracle Data Guard Installation and Configuration	11
Primary Database Configuration	11
Standby Database Initialization	12
Managed Recovery Process (MRP)	14
Data Guard Broker Configuration	14
Fast Start Failover (FSFO)	15

Virtual Machine Setup

The foundation of the project was the creation of a reproducible virtual environment. A dedicated RHEL 7.8 “gold image” VM was built, configured, and cloned to serve as the basis for all nodes.

Create Virtual Machines (VMs)

Open Oracle VirtualBox.

Create a new VM with type: Red Hat Enterprise Linux (64-bit).

Example name: RAC (serves as a master template).

- Configure resources:
- Memory: 8192 MB
- Processors: 2-4 Network adapters:
- Adapter 1 → Internal Network (private interconnect).
- Adapter 2 → Bridged (public network). Attach RHEL

ISO and start installation.

Install Red Hat Enterprise Linux 7.8.

After installation, reboot.

Install Guest Additions for better VM integration (clipboard, shared folders).

Eject Guest Additions before shutdown.

Template VM

This VM will be used as a base image. Clone it to create db1 and db2.

Operating System Configuration:

Hostname and networking:

Configure hostnames and IPs:

```
hostnamectl set-hostname db1.db.com
```

```
reboot -h now
```

Edit /etc/hosts for both nodes to include rac network configuration:

```
127.0.0.1 localhost localhost.localdomain
```

Public IPs

```
192.168.1.101 db1.db.com db1
```

```
192.168.1.102 db2.db.com db2
```

Private Interconnect

```
192.168.0.101 db1-priv.db.com db1-priv
```

```
192.168.0.102 db2-priv.db.com db2-priv
```

Virtual IPs

```
192.168.1.111 db1-vip.db.com db1-vip
```

```
192.168.1.112 db2-vip.db.com db2-vip
```

SCAN Addresses

```
192.168.1.121 db-scan.db.com db-scan
```

```
192.168.1.122 db-scan.db.com db-scan
```

```
192.168.1.123 db-scan.db.com db-scan
```

Disabling IPv6

Edit /etc/sysctl.conf:

```
net.ipv6.conf.all.disable_ipv6 = 1 net.ipv6.conf.default.disable_ipv6 =  
1 net.ipv6.conf.lo.disable_ipv6 = 1
```

Its also recommended to disable firewall and apply the changes through
`/sbin/sysctl -p`

1. Users and groups setup:

```
groupadd dba -g 1600 useradd -g dba -G dba,vboxsf -s  
/bin/bash oracle -u 1601 useradd -g dba -G dba,vboxsf -s  
/bin/bash grid -u 1602 echo "password" | passwd --stdin  
oracle echo "password" | passwd --stdin grid  
echo "%dba ALL=(ALL:ALL) NOPASSWD: ALL" >> /etc/sudoers
```

2. Local Yum Repository

If u dont plan to use internet connectivity install the yum
repository manually through the following steps:

1. Mount the RHEL ISO:

```
mkdir /cdrom mount  
/dev/cdrom /cdrom
```

2. Create a repository file:

```
cp /cdrom/media.repo /etc/yum.repos.d/ chmod 777  
/etc/yum.repos.d/media.repo mv /etc/yum.repos.d/redhat.repo  
/etc/yum.repos.d/redhat.repo.old
```

3. Update the system:

```
yum update -y
```

3. Package Installation

1. Install required packages:

```
yum install -y compat-openssl10 ksh libnsl sysstat xterm
```

2. Install Oracle preinstall package:

```
curl -o oracle-database-preinstall-19c-1.0-1.el7.x86_64.rpm \
```

```
https://yum.oracle.com/repo/OracleLinux/OL7/latest/x86_64/getPackage/oracle-database-preinstall-19c-1.0-1.el7.x86_64.rpm
```

```
rpm -i oracle-database-preinstall-19c-1.0-1.el7.x86_64.rpm 3.
```

Adjust limits in /etc/security/limits.conf: grid soft stack

10240

```
oracle soft stack 10240 grid
```

```
soft nofile 4096
```

```
grid hard nofile 63536 oracle
```

```
soft nofile 4096 oracle hard
```

```
nofile 63536
```

4. Oracle ASM Configuration

1. Install ASMLIB:

```
yum install -y kmod-oracleasm rpm -i oracleasm-lib-2.0.12-1.el7.x86_64.rpm rpm -i oracleasm-support-2.1.11-2.el7.x86_64.rpm
```

2. Configure **ASM**:

```
oracleasm configure -i systemctl
```

```
enable oracleasm
```

```
systemctl start oracleasm
```

5. Directory Structure

Prepare installation directories, for example mine were: `mkdir`

```
-p /dbi/oracle/V19BaseDatabase mkdir -p
```

```
/dbi/oracle/V19Database
```

```
mkdir -p /dbi/oracle/V19BaseGrid mkdir -
```

```
p /dbi/oracle/V19Grid
```

```
chown oracle:dba /dbi/oracle/V19Database
```

```
/dbi/oracle/V19BaseDatabase
```

```
chown grid:dba /dbi/oracle/V19Grid /dbi/oracle/V19BaseGrid
```

```
chmod -R 775 /dbi
```

6. Time Synchronization

Enable and configure **NTP** services, it makes sure our **RAC**

clusters are synchronized: `systemctl enable chronyd`

```
systemctl restart chronyd chronyc -a 'burst 4/4' chronyc -a
```

```
makestep
```

Now our VM is properly configured for **Grid** and **Database** installation next step would be cloning it twice and editing the ips and hostnames accordingly.

RAC and Database Installation Guide

Pre-Install and Kernel Parameters

Install pre-requisites:

```
yum install oracle-database-preinstall-19c-1.0-1.el7.x86_64.rpm
```

Storage Setup

- Create shared virtual disks:
 - **20 GB** for ASM data.
 - **5 GB** for OCR.
- Create ASM disks:

```
fdisk -l | grep /dev/sd | printf
```

```
"o\n\n\np\n1\n\n\n\nw\n\n" | sudo fdisk /dev/sdb
```

```
printf "o\n\n\np\n1\n\n\n\nw\n\n" | sudo fdisk
```

```
/dev/sdc oracleasm createdisk ASMDATA1
```

```
/dev/sdb1 oracleasm createdisk ASMOCR /dev/sdc1
```

```
oracleasm scandisks
```

```
oracleasm listdisks
```

Software Preparation

- Unzip Grid and Database software on Node 1.
- Install cluster verification utility package: `rpm -i cvuqdisk-1.0.10-1.rpm`
- Optionally run Cluster Verification Utility:

```
./runcluvfy.sh stage -pre crsinst -n db1,db2
```


Install Grid Infrastructure (Clusterware)

1. **Unzip Grid software as grid user:** `unzip -qq GRID_SOFTWARE_ZIP -d /dbi/oracle/V19Grid`
2. **Install CVUQDISK as root:** `cd /dbi/oracle/V19Grid/cv/rpm`
`CVUQDISK_GRP=dba; export CVUQDISK_GRP` `rpm -i cvuqdisk-1.0.10-1.rpm`
3. **Run Grid Setup (as grid user):**
`/dbi/oracle/V19Grid/gridSetup.sh`
4. **Add entry in /etc/oratab:**
`+ASM1:/dbi/oracle/V19Grid:Y`

Install Oracle Database Home

1. **Unzip Oracle Database software (as oracle user):**
`unzip -qq ODB_ZIP -d /dbi/oracle/V19Database/`
2. **Run Database Installer:**
`/dbi/oracle/V19Database/runInstaller`

Create ASM Disk Group

1. **Connect to ASM instance:**
`. oraenv`
`+ASM1`
`sqlplus / as sysasm`

2. Create and mount ASM diskgroup:

```
CREATE DISKGROUP ASMDATA1 EXTERNAL REDUNDANCY  
DISK
```

```
 '/dev/oracleasm/disks/ASMDATA1';
```

```
SELECT STATE, NAME FROM V$ASM_DISKGROUP;
```

```
ALTER DISKGROUP ASMDATA1 MOUNT;
```

```
COL name FOR A10;
```

```
COL compatibility FOR A15;
```

```
SELECT name, compatibility FROM v$asm_diskgroup;
```

```
ALTER DISKGROUP ASMDATA1 SET ATTRIBUTE  
'compatible.asm' = '19.0';
```

Create the RAC Database

1. Run **DBCA** as oracle user to create the database.
2. Add entry in /etc/oratab:

```
ora19c1:/dbi/oracle/V19Database:Y
```

Oracle Data Guard Installation and Configuration

Before configuring Data Guard, both servers were prepared with the following prerequisites:

- Oracle Linux replaced by **Red Hat Enterprise Linux (RHEL) 7.8** for standardization across the cluster and DR environments.
- Oracle 19c Enterprise Edition installed on both primary (db1) and standby (db2).
- Hostname resolution configured in /etc/hosts.
- Firewalls disabled, SELinux set to permissive.
- SSH connectivity verified between primary and standby.
- Both databases configured with: ◦ **ARCHIVELOG mode** ◦ **Force Logging enabled** ◦ **Flashback Database enabled** ◦

DB_UNIQUE_NAME parameter set

You can use the Main RAC VM we setup previously and clone it

Primary Database Configuration

On the primary node (db1):

1. Ensure the database runs in ARCHIVELOG mode: **ALTER DATABASE ARCHIVELOG;**
2. Enable Force Logging:
ALTER DATABASE FORCE LOGGING;
3. Enable Flashback:
ALTER DATABASE FLASHBACK ON;

4. Set a unique database identifier:

```
ALTER SYSTEM SET DB_UNIQUE_NAME='orap' SCOPE=SPFILE;
```

5. Configure listener.ora and tnsnames.ora to allow service resolution between db1 and db2.

Standby Database Initialization

On the standby node (db2):

1. Copy the PFILE from the primary and adjust instancespecific parameters (e.g., DB_UNIQUE_NAME, CONTROL_FILES, LOG_ARCHIVE_DEST).

2. Start the standby in NOMOUNT mode:

```
STARTUP NOMOUNT PFILE='/tmp/init.ora';
```

3. Use RMAN to duplicate the primary to the standby:

```
rman TARGET sys@orap AUXILIARY sys@oras
```

```
RMAN> DUPLICATE TARGET DATABASE FOR STANDBY FROM  
ACTIVE DATABASE DORECOVER;
```

4. Create standby control files and configure standby redo logs to support real-time apply.

On **both primary and standby**, configure Data Guard-related parameters in the SPFILE:

- **Primary (db1 – orap)**

```
ALTER SYSTEM SET
```

```
LOG_ARCHIVE_CONFIG='DG_CONFIG=(orap,oras)'
```

```
SCOPE=BOTH;
```

```
ALTER SYSTEM SET
```

```
LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FI
```

```
LE_DEST VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
```

```
DB_UNIQUE_NAME=orap';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_2='SERVICE=oras
```

```
ASYNCH VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
```

```
DB_UNIQUE_NAME=oras';
```

```
ALTER SYSTEM SET FAL_SERVER=oras;
```

```
ALTER SYSTEM SET FAL_CLIENT=orap;
```

```
ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=AUTO;
```

- **Standby (db2 – oras)**

```
ALTER SYSTEM SET
```

```
LOG_ARCHIVE_CONFIG='DG_CONFIG=(orap,oras)'
```

```
SCOPE=BOTH;
```

```
ALTER SYSTEM SET
```

```
LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FI
```

```
LE_DEST VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
```

```
DB_UNIQUE_NAME=oras';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_2='SERVICE=orap
```

```
ASYNCH VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
```

```
DB_UNIQUE_NAME=orap';
```

```
ALTER SYSTEM SET FAL_SERVER=orap;
```

```
ALTER SYSTEM SET FAL_CLIENT=oras;
```

```
ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=AUTO;
```

Managed Recovery Process (MRP)

On the standby, enable real-time redo apply:

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
USING CURRENT LOGFILE DISCONNECT FROM SESSION;
```

Data Guard Broker Configuration

To simplify management, Oracle Data Guard Broker was enabled:

1. Enable Broker on both databases:

```
ALTER SYSTEM SET DG_BROKER_START=TRUE;
```

2. Configure broker from dgmgrl:

```
DGMGRL> CREATE CONFIGURATION dg_config AS PRIMARY  
DATABASE IS orap CONNECT IDENTIFIER IS orap;
```

```
DGMGRL> ADD DATABASE oras AS CONNECT IDENTIFIER IS oras  
MAINTAINED AS PHYSICAL;
```

```
DGMGRL> ENABLE CONFIGURATION;
```

3. Validate configuration and perform a test switchover.

```
DGMGRL> switchover to 'orap';
Performing switchover NOW, please wait...
Operation requires a connection to database "orap"
Connecting ...
Connected to "orap"
Connected as SYSDBA.
New primary database "orap" is opening...
Operation requires start up of instance "ora" on database "oras"
Starting instance "ora"...
Connected to an idle instance.
ORACLE instance started.
Connected to "oras"
Database mounted.
Database opened.
Connected to "oras"
Switchover succeeded, new primary is "orap"
DGMGRL> █
```

Figure 1: Switchover testing

Fast Start Failover (FSFO)

For maximum availability:

- Protection mode set to **Maximum Availability**.
- Log transport switched to **SYNC**.

```
DGMGRL> edit database orap set property 'LogXptMode'='sync';
Property "LogXptMode" updated
DGMGRL> edit database oras set property 'LogXptMode'='sync';
Property "LogXptMode" updated
DGMGRL> edit configuration set protection mode as maxavailability;
Succeeded.
DGMGRL> enable configuration;
Enabled.
DGMGRL>
```

- Observer process configured to monitor automatic failover.

```
DGMGRL> show observer;

Configuration - ora

  Primary:          orap
Active Target:      oras

Observer "db2.db.com" - Master

  Host Name:          db2.db.com
  Last Ping to Primary: 2 seconds ago
  Last Ping to Target:  2 seconds ago

Observer "db1.db.com" - Backup

  Host Name:          db1.db.com
  Last Ping to Primary: 1 second ago
  Last Ping to Target:  0 seconds ago

DGMGRL>
```

