Deploying Oracle RAC Database 12c on Red Enterprise Linux 7
Best Practices

Roger Lopez, Principal Software Engineer

Version 1.1
March 2015
Comments and Feedback

In the spirit of open source, we invite anyone to provide feedback and comments on any reference architectures. Although we review our papers internally, sometimes issues or typographical errors are encountered. Feedback allows us to not only improve the quality of the papers we produce, but allows the reader to provide their thoughts on potential improvements and topic expansion to the papers. Feedback on the papers can be provided by emailing refarch-feedback@redhat.com. Please refer to the title within the email.

Staying In Touch

Join us on some of the popular social media sites where we keep our audience informed on new reference architectures as well as offer related information on things we find interesting.

Like us on Facebook:
https://www.facebook.com/rhrefarch

Follow us on Twitter:
https://twitter.com/RedHatRefArch

Plus us on Google+:
https://plus.google.com/u/0/b/114152126783830728030/refarch-feedback@redhat.com
# Table of Contents

1 Executive Summary ........................................................................................................... 1

2 Reference Architecture Environment ................................................................................ 2
   2.1 Reference Architecture Overview ................................................................................ 2
   2.2 Network Topology ........................................................................................................ 3
   2.3 Hardware Details ........................................................................................................... 4
   2.4 File System Layout & Disk Space Details ..................................................................... 5
   2.5 Storage Layout ............................................................................................................ 7
   2.6 Swap Space .................................................................................................................. 7
   2.7 Security: Firewall Settings ......................................................................................... 8
   2.8 Security: SELinux ......................................................................................................... 9

3 Reference Architecture Configuration Details .................................................................... 10
   3.1 Setting OS Hostname .................................................................................................. 10
   3.2 Network Configuration .............................................................................................. 11
      3.2.1 Configuring /etc/resolv.conf file ........................................................................ 11
      3.2.2 Public Network Configuration ............................................................................ 11
      3.2.3 Configure SCAN via DNS .................................................................................. 14
      3.2.4 Configure Virtual IP (VIP) via DNS .................................................................. 16
      3.2.5 Private Network Configuration ......................................................................... 17
   3.3 OS Configuration ....................................................................................................... 20
      3.3.1 Using the Red Hat Subscription Manager ............................................................ 20
      3.3.2 NTP Configuration ............................................................................................. 22
      3.3.3 Oracle Database 12c Release 1 (12.1) Package Requirements ............................. 23
      3.3.4 Configuring Security-Enhanced Linux (SELinux) ............................................... 25
      3.3.5 Configuring Firewall Settings ............................................................................ 25
      3.3.6 Setting Virtual Memory ...................................................................................... 28
      3.3.7 Setting Shared Memory (SHMMAX, SHMALL, SHMMNI) .................................... 29
      3.3.8 Setting Semaphores (SEMMSL, SEMMNI, SEMMNS) ....................................... 31
      3.3.9 Ephemeral Network Ports ................................................................................... 31
      3.3.10 Optimizing Network Settings .......................................................................... 32
      3.3.11 Setting NOZEROCONF .................................................................................... 32
      3.3.12 Disabling the avahi-daemon service ................................................................. 32
      3.3.13 Increasing synchronous I/O Requests ............................................................... 33
      3.3.14 Increasing File Handles ..................................................................................... 33
1 Executive Summary

IT organizations face challenges of optimizing Oracle database environments to keep up with the ever increasing workload demands and evolving security risks. This reference architecture provides a step-by-step deployment procedure with the latest best practices to install and configure an Oracle Real Application Clusters (RAC) Database 12c Release 1 (12.1) with Oracle Automatic Storage Management (ASM). It is suited for system, storage, and database administrators deploying Oracle RAC Database 12c Release 1 (12.1) on Red Hat Enterprise Linux 7. It is intended to provide a Red Hat | Oracle reference architecture that focuses on the following tasks:

- Deploying Oracle Grid Infrastructure 12c Release 1 (12.1.0.2.0)
- Deploying Oracle RAC Database Software 12c Release 1 (12.1.0.2.0)
- Deploying an Oracle RAC Database 12c Release 1 (12.1.0.2.0) with shared SAN disks
- Using Oracle ASM disks with udev rules
- Securing the Oracle RAC Database 12c environment with SELinux

---

1 Oracle Database 12c – supported on version 12.1.0.2.0 or higher on Red Hat Enterprise Linux 7
2 Reference Architecture Environment

This section focuses on the components used during the deployment of Oracle RAC Database 12c with Oracle Automatic Storage Management (ASM) on Red Hat Enterprise Linux 7 x86_64 in this reference architecture.

2.1 Reference Architecture Overview

A pictorial representation of the environment used in this reference environment is shown in Figure 2.1.1: Reference Architecture Overview

![Reference Architecture Overview](image-url)
2.2 Network Topology

The network topology used in this reference environment consists of two public switches using link aggregation to connect the two switches together (Public Switch A and Public Switch B), creating a single logical switch. A similar link aggregation is done for private switches (Private Switch A and Private Switch B), creating a single logical switch. Ethernet device `ens10f0` on the server connects to Public Switch A, while Ethernet device `ens10f1` on the server connects to Public Switch B. Ethernet devices `ens10f0` and `ens10f1` are bonded together as a bond device, `bond0`, providing high availability for the public network traffic. Ethernet device `ens10f2` on each server connects to Private Switch A, while Ethernet device `ens10f3` on each server connects to Private Switch B. Ethernet devices `ens10f2` and `ens10f3` take advantage of Oracle's Highly Available Internet Protocol (HAIP) for Oracle's private interconnect. HAIP can load balance Ethernet traffic for up to for Ethernet devices. Due to the use of Oracle's HAIP, no bond device is created for private Ethernet devices `ens10f2` and `ens10f3` on each node within the Oracle RAC Database cluster. Figure 2.2.1: Network Topology shows the pictorial representation of the network topology.
2.3 Hardware Details

The following are the hardware requirements to properly install Oracle RAC Database 12c on a x86_64 system:

- Minimum of 4 GB of RAM for the installation of both Oracle Grid Infrastructure and Oracle Database

- The minimum of three Network Interface Cards (NIC) with the usage of direct attach storage or fibre channel storage; however, four NICs are recommended.

- Red Hat Enterprise Linux 7.x Server x86_64 with kernel 3.10.0-123 or higher

- Console access that supports 1024 x 768 resolution to ensure correct display of Oracle’s Universal Installer (OUI).

- All nodes within the Oracle RAC Database environment require the same chip architecture. This reference architecture uses 64-bit processors on all nodes within the cluster.

Table 2.3.1: Server Details specifies the hardware for the server used within this reference environment. This hardware meets the minimum requirements for properly installing Oracle Database 12c Release 1 (12.1) on a x86_64 system.

<table>
<thead>
<tr>
<th>Server Hardware</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle RAC 12c Release 1 Server (db-oracle-node1, db-oracle-node2) [2 x HP ProLiant DL370 G6 Server]</td>
<td>Red Hat Enterprise Linux 7 kernel 3.10.0-123.el7.x86_64</td>
</tr>
<tr>
<td></td>
<td>2 Socket, 8 Core (16 cores) Intel(R) Xeon(R) CPU W5580 @ 3.20 GHz</td>
</tr>
<tr>
<td></td>
<td>48 GB of memory, DDR3 4096 MB @ 1333 MHz DIMMs</td>
</tr>
<tr>
<td></td>
<td>2 x NetXen NX3031 1/10-Gigabit Network Interface Cards (NICs) for public network</td>
</tr>
<tr>
<td></td>
<td>2 x NetXen NX3031 1/10-Gigabit Network Interface Cards (NICs) for private network</td>
</tr>
<tr>
<td></td>
<td>1 x Qlogic ISP2532 8GB Fibre Channel Dual Port HBA³</td>
</tr>
</tbody>
</table>

Table 2.3.1: Server Details

2 Preparing your Cluster, Oracle Documentation - [https://docs.oracle.com/database/121/TDPRC/preparing.htm#TDPRC122](https://docs.oracle.com/database/121/TDPRC/preparing.htm#TDPRC122)

3 Due to hardware limitations, this reference architecture uses only one dual port HBA. It is recommended to have at least two Fibre Channel Single Port HBAs for high availability.
Table 2.3.2: Switch Details specifies the Fibre Channel and Ethernet switches used within this reference environment.

<table>
<thead>
<tr>
<th>Switch Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fibre Channel</td>
</tr>
<tr>
<td>Public Network</td>
</tr>
<tr>
<td>Private Network (Private VLANs)</td>
</tr>
</tbody>
</table>

Table 2.3.2: Switch Details

Table 2.3.3: Storage Details specifies the storage used for storing Oracle data files within this reference environment.

<table>
<thead>
<tr>
<th>Storage Hardware</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP StorageWorks MSA2324fc Dual Controller Array</td>
<td>24 x 146 GB 15K SAS Hard disks</td>
</tr>
</tbody>
</table>

Table 2.3.3: Storage Details

2.4 File System Layout & Disk Space Details

The following is the disk space requirements for properly installing Oracle Database 12c Release 1 (12.1) software for this reference environment.

<table>
<thead>
<tr>
<th>Software</th>
<th>Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Grid Infrastructure Home (includes software files)</td>
<td>12 GB</td>
</tr>
<tr>
<td>Oracle Database Home Enterprise Edition (includes software files and data files)</td>
<td>12 GB</td>
</tr>
<tr>
<td>/tmp</td>
<td>1 GB</td>
</tr>
</tbody>
</table>

Table 2.4.1: Disk Space Requirements

NOTE: The actual amount of disk space consumed for Oracle Grid Infrastructure Home and Oracle Database Home Enterprise Edition may vary.
Table 2.4.2: File System Layout specifies the file system layout for the server used in this reference environment. The layout ensures the disk space requirements to properly install the Oracle Grid Infrastructure and Oracle Database software for Oracle RAC Database 12c Release 1 (12.1).

<table>
<thead>
<tr>
<th>File System Layout</th>
<th>Disk Space Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>15 GB</td>
</tr>
<tr>
<td>/dev/shm</td>
<td>24 GB</td>
</tr>
<tr>
<td>/boot</td>
<td>248 MB</td>
</tr>
<tr>
<td>/home</td>
<td>8 GB</td>
</tr>
<tr>
<td>/tmp</td>
<td>4 GB</td>
</tr>
<tr>
<td>/u01</td>
<td>50 GB</td>
</tr>
<tr>
<td>/usr</td>
<td>5 GB</td>
</tr>
<tr>
<td>/var</td>
<td>8 GB</td>
</tr>
</tbody>
</table>

Table 2.4.2: File System Layout

Oracle RAC Database 12c Release 1 (12.1) recommends three volumes each of at least 4 GB in size to store the Oracle Cluster Registry (OCR), voting disks, and the Oracle Grid Infrastructure Management Repository (GIMR) within an Oracle ASM disk group with the use of normal redundancy. The OCR manages the Oracle Clusterware and Oracle RAC Database 12c Release 1 configuration information. The voting disk manages any information pertaining to the node membership. GIMR is a mandatory installation with the release of Oracle Grid Infrastructure 12c Release 1 (12.1.0.2). Via Oracle's documentation, it enables the Cluster Health Monitor, Oracle Database QoS Management, Rapid Home Provisioning, and provides a historical metric repository that simplifies viewing of past performance and diagnosis issues. While the size of the Oracle data files varies for each solution, the following are the Oracle data file sizes used for this reference environment.

<table>
<thead>
<tr>
<th>Volume</th>
<th>Volume Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Database Volume 1 (db1)</td>
<td>100 GB</td>
</tr>
<tr>
<td>Oracle Database Volume 2 (db2)</td>
<td>100 GB</td>
</tr>
<tr>
<td>Fast Recovery Area (fra)</td>
<td>200 GB</td>
</tr>
<tr>
<td>Oracle Redo Log Volume (redo)</td>
<td>10 GB</td>
</tr>
<tr>
<td>OCR &amp; Voting Disks (ocrvote1)</td>
<td>4 GB</td>
</tr>
<tr>
<td>OCR &amp; Voting Disks (ocrvote2)</td>
<td>4 GB</td>
</tr>
<tr>
<td>OCR &amp; Voting Disks (ocrvote3)</td>
<td>4 GB</td>
</tr>
</tbody>
</table>

Table 2.4.3: Oracle OCR, Voting Disk, & Data File Sizes
2.5 Storage Layout

Table 2.5.1: Storage Disk Layout for Reference Architecture shows the storage disk layout for each volume.

<table>
<thead>
<tr>
<th>Virtual Diskgroup Name</th>
<th>Volume Name</th>
<th>Volume Size</th>
<th>RAID Group Type</th>
<th>Hard Drive Count</th>
<th>Hot Spares Available</th>
<th>Size of Virtual Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>vd01</td>
<td>db1</td>
<td>100 GB</td>
<td>Raid 10</td>
<td>8</td>
<td>1</td>
<td>586 GB</td>
</tr>
<tr>
<td>vd02</td>
<td>db2</td>
<td>100 GB</td>
<td>Raid 10</td>
<td>8</td>
<td>1</td>
<td>586 GB</td>
</tr>
<tr>
<td>vd03</td>
<td>fra</td>
<td>200 GB</td>
<td>Raid 5</td>
<td>5</td>
<td>0</td>
<td>586 GB</td>
</tr>
<tr>
<td>vd04</td>
<td>redo</td>
<td>10 GB</td>
<td>Raid 1</td>
<td>2</td>
<td>0</td>
<td>146 GB</td>
</tr>
<tr>
<td>vd04</td>
<td>ocrvote1</td>
<td>4 GB</td>
<td>Raid 1</td>
<td>2</td>
<td>0</td>
<td>146 GB</td>
</tr>
<tr>
<td>vd04</td>
<td>ocrvote2</td>
<td>4 GB</td>
<td>Raid 1</td>
<td>2</td>
<td>0</td>
<td>146 GB</td>
</tr>
<tr>
<td>vd04</td>
<td>ocrvote3</td>
<td>4 GB</td>
<td>Raid 1</td>
<td>2</td>
<td>0</td>
<td>146 GB</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>146 GB</td>
</tr>
</tbody>
</table>

NOTE: The Hot Spare Available is a Global Hot Spare that can be applied to any virtual disk group in case of failure.

2.6 Swap Space

Swap space is determined by the amount of RAM found within the system. The following table displays the swap space recommendation. This reference environment allocates 16 GB of RAM for swap space.

<table>
<thead>
<tr>
<th>RAM</th>
<th>Swap Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 GB up to 16 GB</td>
<td>Equal to the size of RAM</td>
</tr>
<tr>
<td>Greater than 16 GB</td>
<td>16 GB of RAM</td>
</tr>
</tbody>
</table>

Table 2.6.1: Recommended Swap Space

NOTE: When calculating swap space, ensure not to include RAM assigned for hugepages. More information on hugepages can be found in Section 4.1.5 Enabling HugePages.
2.7 Security: Firewall Settings

Red Hat Enterprise Linux 7 introduces the dynamic firewall daemon, `firewalld`. `firewalld` provides a dynamically managed firewall with support for network/firewall zones to define the trust level of network connections or interfaces. `firewalld` is the default firewall service in Red Hat Enterprise Linux 7, however, `iptables` service is still available. It is important to note that with the `iptables` service, every single change means flushing all the old rules and reading all of the new rules from the `/etc/sysconfig/iptables`. With `firewalld`, there is no re-creating of all the rules, only the differences are applied. Consequently, `firewalld` can change the setting during runtime without existing connections being lost. For the purposes of this reference architecture, `firewalld` is used and is the preferred method of implementing firewall rules. This section focuses on providing the details required to run `firewall-cmd` successfully for an Oracle Database environment. Table 2.7.1: Firewall Port Settings lists the enabled ports in this reference environment.

<table>
<thead>
<tr>
<th>Port</th>
<th>Services</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>ssh</td>
<td>TCP</td>
<td>Secure Shell (SSH)</td>
</tr>
<tr>
<td>1521</td>
<td>-</td>
<td>TCP</td>
<td>Oracle Transparent Network Substrate (TNS) Listener default port</td>
</tr>
<tr>
<td>5500</td>
<td>-</td>
<td>TCP</td>
<td>EM Express 12c default port</td>
</tr>
</tbody>
</table>

*Table 2.7.1: Firewall Port Settings*

---

4 Linux man pages - `man (1) firewalld`
2.8 Security: SELinux

Starting with Oracle 11g Release 2 version 11.2.0.3, SELinux is supported for Oracle database environments. The system in this reference environment runs with SELinux enabled and set to ENFORCING mode. **Table 2.8.1: SELinux Packages** lists the required SELinux packages.

<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>selinux-policy</td>
<td>3.12.1-153</td>
</tr>
<tr>
<td>selinux-policy-targeted</td>
<td>3.12.1-153</td>
</tr>
</tbody>
</table>

**Table 2.8.1: SELinux Packages**
3 Reference Architecture Configuration Details

This reference architecture focuses on the deployment of Oracle RAC Database 12c Release 1 (12.1) with Oracle Automatic Storage Management (ASM) on Red Hat Enterprise Linux 7 x86_64. The configuration is intended to provide a comprehensive Red Hat | Oracle solution. The key solution components covered within this reference architecture consists of:

- Red Hat Enterprise Linux 7
- Oracle Grid Infrastructure 12c Release 1 (12.1.0.2.0)
- Oracle Database 12c Release 1 Software Installation (12.1.0.2.0)
- Deploying an Oracle Database 12c Release 1 (12.1.0.2.0) with shared SAN disks
- Enabling Security-Enhanced Linux (SELinux)
- Configuring Device Mapper Multipathing
- Using udev Rules instead of Oracle ASMLib

3.1 Setting OS Hostname

A unique host name is required for the installation of Oracle RAC Database 12c Release 1 (12.1) The host names within this reference environment is: db-oracle-node1 and db-oracle-node2. To set the host name, please follow the instructions below.

Set the host name via the use of the `hostnamectl` command. An example of setting `db-oracle-node1` host name is shown below.

```
# hostnamectl set-hostname db-oracle-node1.cloud.lab.eng.bos.redhat.com
```

Verify the status:

```
# hostnamectl status
Static hostname: db-oracle-node1.cloud.lab.eng.bos.redhat.com
  Icon name: computer
  Chassis: n/a
  Machine ID: eb00fab21b644a45ba9d2532554f7b49
  Boot ID: 503421cddcd54634ba926daca7e08cc
  Operating System: Red Hat Enterprise Linux
  CPE OS Name: cpe:/o:redhat:enterprise_linux:7.0:GA:server
    Kernel: Linux 3.10.0-123.el7.x86_64
  Architecture: x86_64
```

---

6 Oracle Grid Infrastructure 12c (12.1.0.2.0) or higher supported version with Red Hat Enterprise Linux 7
7 Oracle Database 12c (12.1.0.2.0) or higher supported version with Red Hat Enterprise Linux 7
8 No Oracle ASMLib for Red Hat Enterprise Linux 7 is available at the time of this writing. My Oracle Support Article ID: 1089399.1
3.2 Network Configuration

The network configuration focuses on the proper setup of public and private network interfaces along with the DNS configuration for the Single Client Access Name (SCAN). The public bonded network interface provides an Oracle environment with high availability in case of a network interface failure. The High Availability Internet Protocol (HAIP) provides the private network interfaces with failover and load balancing across each private network interface. SCAN provides the Oracle RAC Database 12c Release 1 (12.1) environment a single name that can be used by any client trying to access an Oracle Database within the cluster.

3.2.1 Configuring /etc/resolv.conf file

The resolver is a set of routines in the C library that provides access to the Internet Domain Name System (DNS). The resolver configuration file contains information that is read by the resolver routines the first time they are invoked by a process. The file is designed to be human readable and contains a list of keywords with values that provide various types of resolver information.\(^9\) The /etc/resolv.conf file for this reference environment consists of two configuration options: nameserver and search. The search option is used to search for a host name that is part of a particular domain. The nameserver option is the IP address of the name server the systems (db-oracle-node1, db-oracle-node2) must query. If more than one nameserver is listed, the resolver library queries them in order. An example of the /etc/resolv.conf file used on each node of the reference environment is shown below.

```
cat /etc/resolv.conf
search cloud.lab.eng.bos.redhat.com
nameserver 10.19.143.247
nameserver 10.19.143.248
nameserver 10.19.255.2
```

3.2.2 Public Network Configuration

The public network configuration consists of two network interfaces bonded together to provide high availability. The example below shows how to bond physical interfaces ens10f0 and ens10f1 with a bond device labeled bond0. NetworkManager will be used to setup the bond device.

**NOTE:** NetworkManager is optional to setup bonding or Ethernet device interfaces, however, it is the method used within this reference environment.

Check the status of NetworkManager:

```
# systemctl status NetworkManager.service
NetworkManager.service - Network Manager
Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled)
Active: active (running) since Tue 2014-10-28 11:28:22 CDT; 1 day 22h ago
Main PID: 1989 (NetworkManager)
CGroup: /system.slice/NetworkManager.service
   └─1989 /usr/sbin/NetworkManager –no-daemon
```

\(^9\) Linux man pages – man resolv.conf
Enable NetworkManager to start upon boot time, if not already:

```bash
# systemctl enable NetworkManager.service
```

Startup the NetworkManager text user interface (TUI) `nmtui` to setup the bonded devices:

```bash
# nmtui
```

1. Select **Edit a connection**, and press **OK**.
2. Select **Add**, hit Enter.
3. Within the **New Connection** window, highlight **Bond**, and press **Create**.
4. Change **Profile Name** if desired, otherwise within Device type **bond0**
5. Select **Add**, and within the **New Connection** window, select **Ethernet**, and press **Create**.
6. Within the window, change **Profile name** if desired, and enter the appropriate network device, i.e. ens10f0 and click **OK**.
7. Repeat step 6 for any additional network devices to add to the bond device.
8. Change **Mode** to **Active-Backup** and select one of the Ethernet devices as the **Primary**, such as ens10f0.
9. Change **IPv4 Configuration** from <Automatic> to <Manual> and press **Show**.
10. Add **Addresses**, **Gateway**, **DNS servers**, **Search domains** as required.
11. Ensure **Automatically connect** and **Available to all users** is checked and press **OK**.
12. Exit the `nmtui` utility by pressing **Quit**.

**Figure 3.2.2.1: Bond Device using Network Manager** In the following page shows the setup of the bond device.
Restart *NetworkManager* service:

```
# systemctl restart NetworkManager.service
```

Figure 3.2.2.1: Bond Device using Network Manager
Once the bond0 device is configured on each node, use the **ping** command to verify connectivity as follows:

On node one labeled *db-oracle-node1*,

```
# ping db-oracle-node2
PING db-oracle-node2.cloud.lab.eng.bos.redhat.com (10.19.142.52) 56(84) bytes of data.
64 bytes from db-oracle-node2.cloud.lab.eng.bos.redhat.com (10.19.142.52):
  icmp_seq=1 ttl=64 time=0.150 ms
```

On node two labeled *db-oracle-node2*,

```
# ping db-oracle-node1
PING db-oracle-node1.cloud.lab.eng.bos.redhat.com (10.19.142.51) 56(84) bytes of data.
64 bytes from db-oracle-node1.cloud.lab.eng.bos.redhat.com (10.19.142.51):
  icmp_seq=1 ttl=64 time=0.151 ms
```

**NOTE:** Please ensure a DNS entry that resolves to the appropriate *hostname*. This reference architecture resolves the following IP address to the following host names:

<table>
<thead>
<tr>
<th>IP</th>
<th>Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.19.142.51</td>
<td>db-oracle-node1</td>
</tr>
<tr>
<td>10.19.142.52</td>
<td>db-oracle-node2</td>
</tr>
</tbody>
</table>

*Table 3.2.2.1: Public IP & Hostnames*

### 3.2.3 Configure SCAN via DNS

SCAN provides a single name in which a client server can use to connect to a particular Oracle database. The main benefit of SCAN is the ability to keep a client connection string the same even if changes within the Oracle RAC Database environment occur, such as adding or removing of nodes within the cluster. The reason this works is because every client connection sends a request to the SCAN Listener, which then reroutes the traffic to an available VIP Listener within the Oracle RAC cluster to establish a database connection. The setup of SCAN requires the creation of a single name, no longer than 15 characters in length not including the domain suffix, resolving to three IP addresses using a round-robin algorithm from the DNS server. SCAN must reside in the same subnet as the public network within the Oracle RAC Database cluster and be resolvable without the domain suffix. Within the reference environment, the domain is *cloud.lab.eng.bos.redhat.com* and SCAN name is *db-oracle-scan*.

An example DNS entry for the SCAN is as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>db-oracle-scan</td>
<td>A</td>
<td>10.19.142.53</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>10.19.142.54</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>10.19.142.55</td>
</tr>
</tbody>
</table>
An example of the DNS entry for the SCAN to enable reverse lookups is as follows:

```
53  IN  PTR  db-oracle-scan.cloud.lab.eng.bos.redhat.com
54  IN  PTR  db-oracle-scan.cloud.lab.eng.bos.redhat.com
55  IN  PTR  db-oracle-scan.cloud.lab.eng.bos.redhat.com
```

On each node within the Oracle RAC cluster, verify the SCAN configuration within the DNS server is setup properly using the `nslookup` and `host` command as follows:

```
# nslookup db-oracle-scan
Server:          10.19.143.247
Address: 10.19.143.247#53
Name: db-oracle-scan.cloud.lab.eng.bos.redhat.com
Address: 10.19.142.55
Name: db-oracle-scan.cloud.lab.eng.bos.redhat.com
Address: 10.19.142.53
Name: db-oracle-scan.cloud.lab.eng.bos.redhat.com
Address: 10.19.142.54
```

```
# host db-oracle-scan
db-oracle-scan.cloud.lab.eng.bos.redhat.com has address 10.19.142.53
db-oracle-scan.cloud.lab.eng.bos.redhat.com has address 10.19.142.54
db-oracle-scan.cloud.lab.eng.bos.redhat.com has address 10.19.142.55
```

On each node within the Oracle RAC cluster, verify the SCAN configuration reverse lookup is setup properly using the nslookup and host command as follows:

```
# nslookup 10.19.142.53
Server:          10.19.143.247
Address: 10.19.143.247#53
53.142.16.10.in-addr.arpa name = db-oracle-scan.cloud.lab.eng.bos.redhat.com.
```

Repeat the above step for the reverse lookup on the remaining IP addresses (10.19.142.54, 10.19.142.55) used for the SCAN.

**NOTE:** The reference environment resolves the following IP address to the following host name:

<table>
<thead>
<tr>
<th>IP</th>
<th>Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.19.142.53</td>
<td></td>
</tr>
<tr>
<td>10.19.142.54</td>
<td>db-oracle-scan</td>
</tr>
<tr>
<td>10.19.142.55</td>
<td></td>
</tr>
</tbody>
</table>

*Table 3.2.3.1: SCAN IP & Hostname*

For more information on SCAN, please refer to Oracle's documentation\(^{10}\).

\(^{10}\) Oracle Single Client Access Name (SCAN) -

3.2.4 Configure Virtual IP (VIP) via DNS

The virtual IP is an IP address assigned to each node within an Oracle RAC Database environment with the IP address residing in the public subnet. During the installation of the Oracle Grid Infrastructure, each VIP Listener registers with every SCAN Listener. The reason is because when a client sends a request, the SCAN Listener routes the incoming traffic to one of the VIP Listeners within the Oracle RAC Database cluster. If a client connection string uses the VIP to talk directly to the VIP Listener (as done in prior versions), every time changes to the Oracle RAC Database environment are made, such as adding or removing nodes within the cluster, the client connection string would require updating. Due to this, Oracle recommends always using the SCAN for the client connection string.

An example DNS entry for the VIPs is as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>db-oracle-node1-vip</td>
<td>A</td>
<td>10.19.142.56</td>
</tr>
<tr>
<td>db-oracle-node2-vip</td>
<td>A</td>
<td>10.19.142.57</td>
</tr>
</tbody>
</table>

On each node within the Oracle RAC cluster, verify the VIP address for `db-oracle-node1-vip` and `db-oracle-node2-vip` within the DNS server is setup properly using the `nslookup` and `host` command. An example of checking `db-oracle-node1-vip` can be seen below.

```
# nslookup db-oracle-node1-vip
Server: 10.19.143.247
Address: 10.19.143.247#53
Name:  db-oracle-node1-vip.cloud.lab.eng.bos.redhat.com
Address: 10.19.142.56
```

```
# host db-oracle-node1-vip
db-oracle-node1-vip.cloud.lab.eng.bos.redhat.com has address 10.19.142.56
```

An example of the DNS entry for the SCAN to enable reverse lookups is as follows:

<table>
<thead>
<tr>
<th>IN</th>
<th>PTR</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td></td>
<td>db-oracle-node1-vip.cloud.lab.eng.bos.redhat.com</td>
</tr>
<tr>
<td>57</td>
<td></td>
<td>db-oracle-node2-vip.cloud.lab.eng.bos.redhat.com</td>
</tr>
</tbody>
</table>
On each node within the Oracle RAC Database cluster, verify the VIP address reverse lookup for both VIP addresses (10.19.142.56 and 10.19.142.57) is setup properly using the `nslookup` and `host` command. An example is shown using VIP address 10.19.142.56 below.

```bash
# nslookup 10.19.142.56
Server: 10.19.143.247
Address: 10.19.143.247#53

56.142.16.10.in-addr.arpa name = db-oracle-node1-vip.cloud.lab.eng.bos.redhat.com.
```

```bash
# host 10.19.142.56
56.142.16.10.in-addr.arpa domain name pointer db-oracle-node1-vip.cloud.lab.eng.bos.redhat.com.
```

**NOTE:** The VIP address should provide a *Destination Host Unreachable* response if an attempt to ping the VIP or VIP host name is attempted.

This reference environment resolves the following Virtual IP addresses to the following host names:

<table>
<thead>
<tr>
<th>Hostname</th>
<th>10.19.142.56</th>
<th>10.19.142.57</th>
</tr>
</thead>
<tbody>
<tr>
<td>db-oracle-node1-vip</td>
<td>10.19.142.56</td>
<td>10.19.142.57</td>
</tr>
<tr>
<td>db-oracle-node2-vip</td>
<td>10.19.142.57</td>
<td>10.19.142.57</td>
</tr>
</tbody>
</table>

**Table 3.2.4.1: Virtual IP & Hostnames**

### 3.2.5 Private Network Configuration

The private network configuration consists of two network interfaces `ens10f2` and `ens10f3`. The private network is used to provide interconnect communication between all the nodes in the cluster. This is accomplished via Oracle’s Redundant Interconnect, also known as Highly Available Internet Protocol (HAIP), that allows the Oracle Grid Infrastructure to activate and load balance traffic on up to four Ethernet devices for private interconnect communication. The example below shows how to setup physical interfaces `ens10f2` and `ens10f3` to be used with HAIP.

On each node, as the root user, startup the `NetworkManager` text user interface (TUI) `nmtui` to setup the bonded devices:

```bash
# nmtui
1. Select *Edit a connection*, and press *OK*.
2. Highlight *ens10f2*, and select *Edit...*
3. Within the *Edit Connection* window, press *show* in *Ethernet* and set *MTU* to 9000 bytes, change *IPv4 Configuration* from `<Automatic>` to `<Manual>` and press *Show*.
4. Add *Addresses, Gateway, DNS servers, Search domains* as required.
```
5. Ensure *Automatically connect* and *Available to all users* is checked and press *OK*.

6. Exit the *nmtui* utility by pressing *Quit*.

7. Repeat the above steps for *ens10f3* and any other private network devices.

Figure 3.2.5.1: Setting Private Network using Network Manager

**NOTE:** The MTU size is set to 9000 for the enablement of jumbo frames. Ensure jumbo frames are enabled on the private Ethernet switches.
After the setup of Ethernet devices *ens10f2* and *ens10f3* on each node, restart the 
*NetworkManager* service:

```
# systemctl restart NetworkManager.service
```

Ensure all private Ethernet interfaces are set to different subnets on each node. If different 
subnets are not used and connectivity is lost, this can cause a node reboot within the cluster. 
For the reference environment, subnets 192.11.142.0/24 and 192.12.142.0/24 are used on 
each node within the Oracle RAC Database 12c cluster.

Verify connectivity on each node using the **ping** command.

On node one labeled *db-oracle-node1*,

```
# ping 192.11.142.52
# ping 192.12.142.52
```

On node two labeled *db-oracle-node2*,

```
# ping 192.11.142.51
# ping 192.12.142.51
```

<table>
<thead>
<tr>
<th>IP</th>
<th>Ethernet Interface</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.11.142.51</td>
<td>ens10f2</td>
<td>db-oracle-node1</td>
</tr>
<tr>
<td>192.12.142.51</td>
<td>ens10f3</td>
<td>db-oracle-node1</td>
</tr>
<tr>
<td>192.11.142.52</td>
<td>ens10f2</td>
<td>db-oracle-node2</td>
</tr>
<tr>
<td>192.12.142.52</td>
<td>ens10f3</td>
<td>db-oracle-node2</td>
</tr>
</tbody>
</table>

*Table 3.2.5.1: Private IP, Ethernet Interfaces, & Host*
3.3 OS Configuration

3.3.1 Using the Red Hat Subscription Manager

The `subscription-manager` command is used to register systems to the Red Hat Network (RHN) and to manage the subscription entitlements for the systems. The `--help` option can be specified on the command line to query the command for the available options. If the `--help` option is issued along with a command directive, then options available for the specific command directive are listed.

To use Red Hat Subscription Management for providing packages to a system, the system must first register with the service. To register a system, use the `subscription-manager` command and pass the register command directive. If the `--username` and `--password` options are specified, then the command will not prompt for the RHN Network authentication credentials.

An example of registering a system using `subscription-manager` is shown below.

```
# subscription-manager register --username [User] --password '[Password]'
The system has been registered with id: abcd1234-ab12-ab12-ab12-481ba8187f60
```

After a system is registered, it must be attached to an entitlement pool. For the purposes of this reference environment, the Red Hat Enterprise Linux Server is the pool chosen. To identify and subscribe to the Red Hat Enterprise Linux Server entitlement pool, the following command directives are required.

```
# subscription-manager list --available | grep -A8 "Red Hat Enterprise Linux Server"
Subscription Name: Red Hat Enterprise Linux Server, Standard (8 sockets) (Unlimited guests)
Provides:
     Red Hat Beta
     Oracle Java (for RHEL Server)
     Red Hat Enterprise Linux Server
     Red Hat Software Collections Beta (for RHEL Server)
SKU: RH0186633
Contract: 10541483
Pool ID: 8a85f98148a8c1a00148a961a0701b1b
Available: 47
Suggested: 1
Service Level: STANDARD
Service Type: L1-L3
```

```
# subscription-manager attach --pool 8a85f98148a8c1a00148a961a0701b1b
Successfully attached a subscription for: Red Hat Enterprise Linux Server, Standard (8 sockets) (Unlimited guests)
```
The Red Hat Enterprise Linux supplementary repository is part of subscribing to the Red Hat Enterprise Linux Server entitlement pool, however, it is disabled by default. To enable the supplementary repository, use the `subscription-manager` command.

**NOTE:** The following step is required in order to install the `compat-libstdc++-33` package that is required for a successful Oracle 12c Database install on Red Hat Enterprise Linux 7. The package is only available in the `rhel-7-server-optional-rpms` repository.

```
# subscription-manager repos --enable=rhel-7-server-optional-rpms
```

For more information on the use of Red Hat Subscription Manager, please visit the Red Hat Subscription management documentation.

---

3.3.2 NTP Configuration

The ntpd program is an operating system daemon which sets and maintains the system time, synchronizing with Internet standard time servers. The ntpd program operates by exchanging messages with one or more configured servers at designated poll intervals.

To configure the ntpd daemon, follow the instructions below.

On all nodes within the Oracle RAC Database cluster,
1. If not installed, install ntp via yum as follows:

```
# yum install ntp
```

2. Edit the /etc/ntp.conf file with a text editor such as vi.

```
# vi /etc/ntp.conf
```

3. Locate the following public server pool section, and modify to include the appropriate NTP servers. For the purposes of this reference environment, only one NTP server is used, but three is recommended. The iburst option is added to speed up the time in which it takes to properly sync with the NTP servers.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 10.5.26.10 iburst
```

4. Save all the changes within the /etc/ntp.conf file

5. The following -x option within the /etc/sysconfig/ntpd file needs to be added to prevent the time synched by the NTP daemon to be adjusted backward.

```
# Command line options for ntpd
OPTIONS="-x -g"
```

6. Start the ntpd daemon via the command:

```
# systemctl start ntpd.service
```

7. Ensure that the ntpd daemon is started when the system is booted.

```
# systemctl enable ntpd.service
ln -s '/usr/lib/systemd/system/ntpd.service' '/etc/systemd/system/multi-user.target.wants/ntpd.service'
```

---

12 ntpd – Network Time Protocol (NTP) daemon man page – man ntpd (8)
8. Verify the `ntpd` daemon status.

```
# systemctl status ntpd.service
ntpd.service - Network Time Service
Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled)
Active: active (running) since Tue 2015-01-27 10:51:34 CST; 2s ago
Process: 7743 ExecStart=/usr/sbin/ntpd -u ntp:ntp $OPTIONS (code=exited, status=0/SUCCESS)
Main PID: 7744 (ntpd)
   CGroup: /system.slice/ntpd.service
         /u2514\u25007744 /usr/sbin/ntpd -u ntp:ntp -x -g
```

### 3.3.3 Oracle Database 12c Release 1 (12.1) Package Requirements

A specific set of packages is required to properly deploy Oracle RAC Database 12c Release 1 (12.1) on Red Hat Enterprise Linux 7 (x86_64). The number of installed packages required varies depending on whether a default or minimal installation of Red Hat Enterprise Linux 7 (x86_64) is performed. For the purposes of this reference environment, a minimal Red Hat Enterprise Linux 7 installation is performed to reduce the number of installed packages. A sample kickstart file as been provided within [Appendix M Configuration Files](#). Red Hat Enterprise Linux 7 installation requires the following group packages:

<table>
<thead>
<tr>
<th>Required Group Packages</th>
</tr>
</thead>
<tbody>
<tr>
<td>@Base</td>
</tr>
<tr>
<td>@Core</td>
</tr>
</tbody>
</table>

*Table 3.3.3.1: Group Packages*
Oracle Grid Infrastructure 12.1 and Oracle Database 12.1 required x86_64 RPM packages:

<table>
<thead>
<tr>
<th>Required Packages</th>
</tr>
</thead>
<tbody>
<tr>
<td>binutils</td>
</tr>
<tr>
<td>compat-libcap1</td>
</tr>
<tr>
<td>compat-libstdc++-33</td>
</tr>
<tr>
<td>gcc</td>
</tr>
<tr>
<td>gcc-c++</td>
</tr>
<tr>
<td>glibc-devel</td>
</tr>
<tr>
<td>ksh</td>
</tr>
<tr>
<td>libgcc</td>
</tr>
<tr>
<td>libstdc++</td>
</tr>
<tr>
<td>libstdc++-devel</td>
</tr>
<tr>
<td>libX11</td>
</tr>
<tr>
<td>libXau</td>
</tr>
<tr>
<td>libaio</td>
</tr>
<tr>
<td>libaio-devel</td>
</tr>
<tr>
<td>libdmx</td>
</tr>
<tr>
<td>glibc</td>
</tr>
<tr>
<td>make</td>
</tr>
<tr>
<td>sysstat</td>
</tr>
<tr>
<td>xorg-x11-utils</td>
</tr>
<tr>
<td>xorg-x11-xauth</td>
</tr>
<tr>
<td>libXext</td>
</tr>
<tr>
<td>libXv</td>
</tr>
<tr>
<td>libXi</td>
</tr>
<tr>
<td>libXt</td>
</tr>
<tr>
<td>libXxf86misc</td>
</tr>
<tr>
<td>libXxf86dga</td>
</tr>
<tr>
<td>nfs-utils</td>
</tr>
<tr>
<td>LibXxf86vm</td>
</tr>
</tbody>
</table>

Table 3.3.3.2: Required Packages

After the installation of Red Hat Enterprise Linux 7 is completed, create a file, `req-rpm.txt`, that contains the name of each RPM package listed above on a separate line. For simplicity, this `req-rpm.txt` file is included in Appendix F Oracle Database Package Requirements Text File.

Use the `yum` package manager to install the packages and any of their dependencies with the following command:

```
# yum install `awk '{print $1}' ./req-rpm.txt`
```

A minimum installation of Red Hat Enterprise Linux 7 does not install the X Window System server package, but only the required X11 client libraries. In order to run the Oracle Universal Installer (OUI), a system with the X Window System server package installed is required. Using a system with X Window System installed, `SSH` into the Oracle Database server with the `-Y` option to ensure trusted X11 forwarding is set. The command is as follows:

```
# ssh -Y db-oracle-node1
```

Alternatively, if a system with the X Window System server package is unavailable, install the X Window System server package directly on the Oracle Database Server.

---

13 Packages for Red Hat Enterprise Linux 7 -
https://docs.oracle.com/database/121/LADBI/pre_install.htm#LADBI7534
3.3.4 Configuring Security-Enhanced Linux (SELinux)

*SELinux* is an implementation of a mandatory access control (MAC) mechanism developed by the National Security Agency (NSA). The purpose of *SELinux* is to apply rules on files and processes based on defined policies. When policies are appropriately defined, a system running *SELinux* enhances application security by determining if an action from a particular process should be granted thus protecting against vulnerabilities within a system. The implementation of Red Hat Enterprise Linux 7 enables *SELinux* by default and appropriately sets it to the default setting of *enforcing*. It is highly recommended that *SELinux* be kept in *enforcing* mode when running Oracle RAC Database 12c Release 1 (12.1).

Verify that *SELinux* is running and set to *enforcing*:

On each node within the Oracle RAC cluster, as the *root* user,

```
# sestatus
SELinux status:               enabled
SELinuxfs mount:             /sys/fs/selinux
SELinux root directory:      /etc/selinux
Loaded policy name:           targeted
Current mode:                 enforcing
Mode from config file:        enforcing
Policy MLS status:            enabled
Policy deny_unknown status:   allowed
Max kernel policy version:    28
```

If the system is running in *PERMISSIVE* or *DISABLED* mode, modify the */etc/selinux/config* file and set *SELinux* to enforcing as shown below.

```
SELINUX=enforcing
```

The modification of the */etc/selinux/config* file takes effect after a reboot. To change the setting of *SELinux* immediately without a reboot, run the following command:

```
# setenforce 1
```


3.3.5 Configuring Firewall Settings

Firewall access and restrictions play a critical role in securing your Oracle RAC Database 12c Release 1 (12.1) environment. It is not uncommon for corporations to be running hardware based firewalls to protect their corporate networks. Due to this, enabling firewall may not be required. However, this reference environment demonstrates how to successfully implement firewall settings for an Oracle RAC Database environment. The firewall rules described below only apply to the public network. Oracle recommends the private network should not have any firewall rules as this can cause issues with the installation of Oracle RAC Database, as well
as, disruption with the Oracle RAC Database private interconnect. It is highly recommended that the private network be isolated and communicate only between nodes locally. Red Hat Enterprise Linux 7 introduces the use of **firewalld**, a dynamic firewall daemon, instead of the traditional **iptables service**. **firewalld** works by assigning network zones to assign a level of trust to a network and its associated connections and interfaces\(^{14}\). The key difference and advantage of **firewalld** over **iptables service** is that it does not require flushing of old firewall rules to apply the new firewall rules. **firewalld** changes the settings during runtime without losing existing connections\(^{14}\). With the implementation of **firewalld**, the **iptables service** configuration file `/etc/sysconfig/iptables` does not exist. It is recommended that the firewall settings be configured to permit access to the Oracle Database network ports only from authorized database or database-management clients. For example, in order to allow access to a specific database client with an IP address of 10.19.142.54 and to make requests to the database server via SQL*Net using Oracle's TNS (Transparent Network Substrate) Listener (default port of 1521), the following permanent firewall rule within the public zone must be added to the **firewalld** configuration.

On all nodes within the Oracle RAC cluster unless otherwise specified,

```
# firewall-cmd --permanent --zone=public --add-rich-rule="rule family="ipv4"
  source address="10.19.142.54" port protocol="tcp" port="1521" accept"
success
```

Likewise, if a particular database client with an IP address of 10.19.142.54 required access to the web-based EM Express that uses the default port of 5500, the following firewall rich rule must be added using the `firewall-cmd` command.

```
# firewall-cmd --permanent --zone=public --add-rich-rule="rule family="ipv4"
  source address="10.19.142.54" port protocol="tcp" port="5500" accept"
success
```

Ensure the firewall allows all traffic from the private network by accepting all traffic (trusted) from the private Ethernet interfaces `ens10f2` and `ens10f3` from all nodes within the Oracle RAC cluster. It is highly recommended that the private network be isolated and communicate only between nodes locally.

```
# firewall-cmd --permanent --zone=trusted --change-interface=ens10f2
success

# firewall-cmd --permanent --zone=trusted --change-interface=ens10f3
success
```

The following rules are added to satisfy the Oracle Installer's prerequisites. Once Oracle installation is complete, the following rules can be removed. Steps for removal shown upon the completion of the Oracle RAC Database installation in Section 6.1.2 Removal of **firewalld** Trusted Source Address

---

On node one of the Oracle RAC cluster, add the public source IP of all remaining nodes. This reference environment only adds the public IP of node two of the Oracle RAC cluster as it is a two-node Oracle RAC Database environment.

```
# firewall-cmd --permanent --zone=trusted --add-source=10.19.142.52/21
```

On node two of the Oracle RAC cluster, add the public source IP of all remaining nodes. This reference environment only adds the public IP of node one of the Oracle RAC cluster as it is a two-node Oracle RAC Database environment.

```
# firewall-cmd --permanent --zone=trusted --add-source=10.19.142.51/21
```

Once the rules have been added, run the following command to activate:

```
# systemctl restart firewalld.service
```

To verify the port 1521 & 5500 has been added and database client with IP address of 10.19.142.54 has been properly added, run the following command:

```
# firewall-cmd --zone=public --list-all
```

```
public (default, active)
  interfaces: bond0 ens10f0 ens10f1
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
    rule family="ipv4" source address="10.19.142.54" port port="1521" protocol="tcp" accept
    rule family="ipv4" source address="10.19.142.54" port port="5500" protocol="tcp" accept
```

To verify the firewall rules being applied to the trusted zone for the private Ethernet interfaces, and temporarily for the source public IP run the following command:

Example of `db-oracle-node1`

```
# firewall-cmd --zone=trusted --list-all
```

```
trusted (active)
  interfaces: ens10f2 ens10f3
  sources: 10.19.142.52/21
  services: 
  ports: 
  masquerade: no 
  forward-ports: 
  icmp-blocks: 
  rich rules: 
```

**NOTE:** A full listing of all the firewall settings for this reference architecture can be found at Appendix D Firewalld Configuration File.
3.3.6 Setting Virtual Memory

Tuning virtual memory requires the modification of five kernel parameters that affect the rate at which virtual memory is used within Oracle RAC databases. It is important to note the recommended values listed are to be used as a starting point when setting virtual memory.

A brief description and recommended settings for the virtual memory parameters, as well as, the definition of dirty data are described below.

**SWAPPINESS**\textsuperscript{15} - Starting with Red Hat Enterprise Linux 6.4 and above, the definition of swappiness has changed. Swappiness is defined as a value from 0 to 100 which controls the degree to which the system favors anonymous memory or the page cache. A high value improves file-system performance, while aggressively swapping less active processes out of memory. A low value avoids swapping processes out of memory, which usually decreases latency, at the cost of I/O performance. The default value is 60. Warning: Since Red Hat Enterprise Linux 6.4, setting swappiness to 0 will even more aggressively avoid swapping out, which increases the risk of out-of-memory (OOM) killing under strong memory and I/O pressure. To achieve the same behavior of swappiness as previous versions of Red Hat Enterprise Linux 6.4 in which the recommendation was to set swappiness to 0, set swappiness to the value of 1. The recommendation of swappiness for Red Hat Enterprise Linux 6.4 or higher running Oracle databases is now the value of 1.

**DIRTY DATA** – Dirty data is data that has been modified and held in the page cache for performance benefits. Once the data is flushed to disk, the data is clean.

**DIRTY\_RATIO**\textsuperscript{15} – Contains, as a percentage of total system memory, the number of pages at which a process which is generating disk writes will itself start writing out dirty data. The default value is 20. The recommended value is 80. The reasoning behind increasing the value from the standard Oracle 15 recommendation to 80 is because dirty ratio defines the maximum percentage of total memory that be can be filled with dirty pages before user processes are forced to write dirty buffers themselves during their time slice instead of being allowed to do more writes. All processes are blocked for writes when this occurs due to synchronous I/O, not just the processes that filled the write buffers. This can cause what is perceived as unfair behavior where a single process can hog all the I/O on a system. As for all parameters in this reference architecture, there is no “one-size fits all” value and the recommendation should be only seen as a starting point.

**DIRTY\_BACKGROUND\_RATIO**\textsuperscript{15} – Contains, as a percentage of total system memory, the number of pages at which the background write back daemon will start writing out dirty data. The Oracle recommended value is 3.

**NOTE:** With the dirty\_background\_ratio set to 3 and dirty\_ratio set to 80, the background write back daemon will start writing out the dirty data when it hits the 3% threshold

---

\textsuperscript{15} RHEL7 Kernel Documentation (requires package kernel-doc to be installed) - /usr/share/doc/kernel-doc-3.10.0/Documentation/sysctl/vm.txt
asynchronously, however, none of that data is written synchronously until the dirty_ratio is 80% full which is what causes for all processes to be blocked for writes when this occurs.

$DIRTY\_EXPIRE\_CENTISECS^{15}\$ - Defines when dirty in-memory data is old enough to be eligible for writeout. The default value is 3000, expressed in hundredths of a second. The Oracle recommended value is 500.

$DIRTY\_WRITEBACK\_CENTISECS^{15}\$ - Defines the interval of when writes of dirty in-memory data are written out to disk. The default value is 500, expressed in hundredths of a second. The Oracle recommended value is 100.

Prior to making any changes to the `/etc/sysctl.conf`, ensure to create a backup as follows:

```
# cp /etc/sysctl.conf /etc/sysctl.conf.bkup
```

```
vmm.swappiness = 1
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
```

For the changes to take effect immediately, run the following command on each node of the Oracle RAC cluster:

```
# sysctl -p /etc/sysctl.conf
```

**NOTE:** A full listing of all the kernel parameters modified within the `/etc/sysctl.conf` file can be found at Appendix G Kernel Parameters.

### 3.3.7 Setting Shared Memory (SHMMAX, SHMALL, SHMMNI)

Shared memory allows processes to communicate with each other by placing regions of memory into memory segments. In the case of Oracle, shared memory segments are used by the System Global Area (SGA) to store incoming data and control information. The size of Oracle's SGA impacts the amount of shared memory pages and shared memory segments to be set within a system. By default, Red Hat Enterprise Linux 7 provides a large amount of shared memory pages and segments. However, the appropriate allocation for a system depends on the size of the SGA within an Oracle RAC database instance.

In order to allocate the appropriate amount of shared memory pages and shared memory segments for a system running an Oracle database, the kernel parameters $SHMMAX$, $SHMALL$, and $SHMMNI$ must be set.

$SHMMAX$ – is the maximum size in bytes of a single shared memory segment

$SHMALL$ – is the maximum total amount of shared memory pages

$SHMMNI$ – is the maximum total amount of shared memory segments

A default installation of Red Hat Enterprise Linux 7 x86_64 provides a maximum size of a single shared memory segment, $SHMMAX$, to 4294967295 bytes, equivalent to 4 GB -1 byte. This value is important since it regulates the largest possible size of one single Oracle SGA.
shared memory segment. If the Oracle SGA is larger than the value specified by `SHMMAX` (default 4 GB-1 byte), then Oracle will be required to create multiple smaller shared memory segments to completely fit Oracle’s SGA. This can cause a significant performance penalty, especially in NUMA environments. In an optimal NUMA configuration, a single shared memory segment for Oracle’s SGA is created on each NUMA node. If `SHMMAX` is not properly sized and creates multiple shared memory segments, `SHMMAX` limitations may keep the system from evenly distributing the shared memory segments across each NUMA node. To determine a proper `SHMMAX` value, the Oracle Database parameter `SGA_MAX_SIZE` needs to be analyzed. If intending to set `SGA_MAX_SIZE` to a value larger than 4 GB, set `SHMMAX` to a size in bytes larger than the `SGA_MAX_SIZE`. If in doubt on how to properly set `SHMMAX`, a value of 4398046511104 (4 TB) can be used. This value is purposely set higher than the architectural memory limits to ensure that any Oracle SGA value set within an Oracle RAC database instance may fit in one single shared memory segment.

After calculating `SHMMAX`, the next step is to determine the maximum amount of shared memory pages (`SHMALL`) in a system by capturing system’s page size in bytes. The following command can be used to obtain the system page size.

```
# getconf PAGE_SIZE
4096
```

A default installation of Red Hat Enterprise Linux 7 x86_64 provides a `SHMALL` value of 268435456, the equivalent of 1 TB in system pages. This is determined by the following formula:

```
SHMALL IN BYTES * PAGE_SIZE
```

From the example above, 268435456 bytes * 4096 bytes = 1099511627776 bytes = 1 TB in system pages

To ensure an adequate amount of memory pages are allocated to a single Oracle SGA, it is recommended that the value of `SHMALL` be set to the following formula:

```
SHMMAX IN BYTES / PAGE_SIZE
```

For example, on a system with the `SHMMAX` set to 4398046511104, the value of `SHMALL` is calculated as follows:

```
# echo “4398046511104 / 4096” | bc
1073741824
```

`SHMMNI` is the maximum total amount of shared memory segments. A default installation of Red Hat Enterprise Linux 7 x86_64 provides a `SHMMNI` default value of 4096. By optimizing the `SHMMAX` value with one shared memory segment per Oracle SGA, this parameter reflects the maximum number of Oracle and ASM instances that can be started on a system. Oracle recommends the value of `SHMMNI` to be left at the default value of 4096.

Snippet of the `/etc/sysctl.conf` file:

```
kernell.shmaxx = 4398046511104
kernell.shmall = 1073741824
kernell.shmnni = 4096
```
NOTE: If the current value found within `/etc/sysctl.conf` for any parameter is higher than the value calculated for `SHMMAX` and `SHMALL` on any nodes within the Oracle RAC cluster, do not change the value found within the `/etc/sysctl.conf` file.

In order for the changes take effect immediately, run the following command on each node of the Oracle RAC cluster:

```
# sysctl -p /etc/sysctl.conf
```

NOTE: A full listing of all the kernel parameters modified within the `/etc/sysctl.conf` file can be found at Appendix G Kernel Parameters.

### 3.3.8 Setting Semaphores (SEMMSL, SEMMNI, SEMMNS)

Red Hat Enterprise Linux 7 provides semaphores for synchronization of information between processes. The kernel parameter `sem` is composed of four parameters:

- `SEMMSL` – is defined as the maximum number of semaphores per semaphore set
- `SEMMNI` – is defined as the maximum number of semaphore sets for the entire system
- `SEMMNS` – is defined as the total number of semaphores for the entire system

NOTE: `SEMMNS` is calculated by `SEMMSL * SEMMNI`

- `SEMOPM` – is defined as the total number of semaphore operations performed per `semop` system call.

The following line is required within the `/etc/sysctl.conf` file to provide sufficient semaphores for Oracle:

```
kernell.sem = 250 32000 100 128
```

In order for the changes take effect immediately, run the following command on each node of the Oracle RAC cluster:

```
# sysctl -p /etc/sysctl.conf
```

### 3.3.9 Ephemeral Network Ports

Oracle recommends that the ephemeral default port range be set starting at 9000 to 65500. This ensures that all well known ports used by Oracle and other applications are avoided. To set the ephemeral port range, modify the `/etc/sysctl.conf` file on each node of the Oracle RAC Database cluster and add the following line:

```
net.ipv4.ip_local_port_range = 9000 65500
```

In order for the changes take effect immediately, run the following command on each node of the Oracle RAC cluster:

```
# sysctl -p /etc/sysctl.conf
```
3.3.10 Optimizing Network Settings

Optimizing the network settings for the default and maximum buffers for the application sockets in Oracle is done by setting static sizes to \texttt{RMEM} and \texttt{WMEM}. The \texttt{RMEM} parameter represents the receive buffer size, while the \texttt{WMEM} represents the send buffer size. The recommended values by Oracle are configured within the \texttt{/etc/sysctl.conf} file on each node of the Oracle RAC cluster.

\begin{verbatim}
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
\end{verbatim}

In order to make the changes take effect immediately, run the following command on each node of the Oracle RAC cluster:

\texttt{# sysctl -p /etc/sysctl.conf}

3.3.11 Setting \texttt{NOZEROCONF}

On each node within the Oracle RAC Database cluster, set the value of \texttt{NOZEROCONF} to \texttt{yes} within the \texttt{/etc/sysconfig/network} file. Setting \texttt{NOZEROCONF} ensures that the route 169.254.0.0/16 is not added to the routing table.

\texttt{NOZEROCONF=yes}

3.3.12 Disabling the \texttt{avahi-daemon} service

From the Red Hat Customer Portal article: \url{https://access.redhat.com/solutions/25463},

The Avahi website defines Avahi as: 'a system which facilitates service discovery on a local network. This helps to plug the laptop or computer into a network and instantly be able to view other people who you can chat with, find printers to print to, or find files being shared...' Avahi-daemon (on by default on Red Hat Enterprise Linux) can interfere with Oracle RAC's multicast heartbeat causing the application-layer interface to assume it has been disconnected on a node and reboot the node. It is not recommended to remove the package due to its many dependencies. The \texttt{avahi} libraries are being used by many packages on a system.

On each node within the Oracle RAC Database cluster, stop and disable the avahi services run the following commands:

\begin{verbatim}
# systemctl stop avahi-dnsconfd
# systemctl stop avahi-daemon
\end{verbatim}

Warning: Stopping avahi-daemon, but it can still be activated by:

\begin{verbatim}
    avahi-daemon.socket
\end{verbatim}

To keep the avahi services off persistently across reboots, on each node run the following:

\begin{verbatim}
# systemctl disable avahi-dnsconfd
# systemctl disable avahi-daemon
rm '/etc/systemd/system/dbus-org.freedesktop.Avahi.service'
rm '/etc/systemd/system/multi-user.target.wants/avahi-daemon.service'
rm '/etc/systemd/system/sockets.target.wants/avahi-daemon.socket'
\end{verbatim}
### 3.3.13 Increasing synchronous I/O Requests

The kernel parameter `FS.AIO-MAX-NR` sets the maximum number of current asynchronous I/O requests. Oracle recommends setting the value to 1048576. In order to add `FS.AIO-MAX-NR` to 1048576, modify the `/etc/sysctl.conf` file on each node of the Oracle RAC cluster as follows:

```
f.s.aio-max-nr = 1048576
```

In order for the changes take effect immediately, run the following command on each node of the Oracle RAC cluster:

```
# sysctl -p /etc/sysctl.conf
```

### 3.3.14 Increasing File Handles

The kernel parameter `FS.FILE-MAX` sets the maximum number of open file handles assigned to the Red Hat Enterprise Linux 7 operating system. Oracle recommends that for each Oracle RAC database instance found within a system, allocate 512*`PROCESSES` in addition to the open file handles already assigned to the Red Hat Enterprise Linux 7 operating system. `PROCESSES` within a database instance refers to the maximum number of processes that can be concurrently connected to the Oracle RAC database by the `oracle` user. The default value for `PROCESSES` is 300 for Oracle RAC Database 12c Release 1 (12.1). To properly calculate the `FS.FILE-MAX` for a system, first identify the current `FS.FILE-MAX` allocated to the system via the following command:

```
# cat /proc/sys/fs/file-max
4897258
```

Next, add all the `PROCESSES` together from each Oracle RAC database instance found within the system and multiply by 512 as seen in the following command.

```
# echo "512 * 300" | bc
153600
```

**NOTE:** To determine the current `PROCESSES` value, log into each Oracle RAC database instance and run the following command below. Since no Oracle RAC database has yet been created within this reference environment, the default value of 300 `PROCESSES` is used.

```
SQL> show parameter processes;
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>processes</td>
<td>integer</td>
<td>300</td>
</tr>
</tbody>
</table>

Finally, add the current `FS.FILE-MAX` value with the new value found from multiplying 512*`PROCESSES` to attain the new `FS.FILE-MAX` value.

```
# echo "4897258 + 153600" | bc
5050858
```
While the value of the `FS_FILE_MAX` parameter varies upon every environment, this reference environment sets the value at 6815744. Oracle recommends a value no smaller than 6815744. Due to the calculation in the above example equating to 5050858, the minimum Oracle recommended value is used. In order to add `FS_FILE_MAX` to 6815744, modify the `/etc/sysctl.conf` file on each node of the Oracle RAC cluster as follows:

```
fs.file-max = 6815744
```

In order for the changes take effect immediately, run the following command on each node of the Oracle RAC cluster:

```
# sysctl -p /etc/sysctl.conf
```

**NOTE:** It is recommended to revisit the `FS_FILE_MAX` value if the `PROCESSES` value is increased for any Oracle RAC databases created.

**NOTE:** A full listing of all the kernel parameters modified within the `/etc/sysctl.conf` file can be found at Appendix G Kernel Parameters.

### 3.3.15 Kernel Panic On OOPS Parameter

The kernel parameter `kernel.panic_on_oops` controls the kernel's behavior when an oops or bug is encountered. By default the value is set to 1, however, the oracle installer requires it to be set within the `/etc/sysctl.conf` file. Add the following parameter to the `/etc/sysctl.conf` file as follows:

```
kernelpanic_on_oops = 1
```

In order for the changes take effect immediately, run the following command:

```
# sysctl -p /etc/sysctl.conf
```

**NOTE:** A full listing of all the kernel parameters modified within the `/etc/sysctl.conf` file can be found at Appendix G Kernel Parameters.

### 3.3.16 Reverse Path Filtering

Red Hat Enterprise Linux 7 defaults to the use of Strict Reverse Path filtering. The reason strict mode is the default is to prevent IP spoofing from Distributed Denial-of-service (DDos) attacks. However, having strict mode enabled on the private interconnect of an Oracle RAC database cluster may cause disruption of interconnect communication. It is recommended to set the `RP_FILTER` from strict mode to loose mode. Loosening the security on the private Ethernet interfaces should not be of concern as best practices recommend for an isolated private network that can only communicate between nodes specifically for Oracle's private interconnect.

To satisfy the Oracle Installer prerequisite, add the following modifications to the `/etc/sysctl.conf` on each node of the Oracle RAC cluster as follows:

```
net.ipv4.conf.ens10f2.rp_filter = 2
net.ipv4.conf.ens10f3.rp_filter = 2
```
In addition to the above, please include the following modification within the 98-oracle.conf file located within the /etc/sysctl.d/ directory. The 98-oracle.conf file can be found within Appendix H 98-oracle.conf

In order for the changes to take effect immediately, run the following command on each node of the Oracle RAC cluster:

```
# sysctl -p /etc/sysctl.conf
[Output Appreciated ...]
net.ipv4.conf.ens10f2.rp_filter = 2
net.ipv4.conf.ens10f3.rp_filter = 2
# sysctl -p /etc/sysctl.d/98-oracle.conf
net.ipv4.conf.ens10f2.rp_filter = 2
net.ipv4.conf.ens10f3.rp_filter = 2
```

**NOTE:** Once the installation of Oracle is complete, the entries found within the /etc/sysctl.conf file are removed and the values are only stored in the 98-oracle.conf file within /etc/sysctl.d directory. See Section 6.1.1 Removal of Oracle Kernel Parameters from /etc/sysctl.conf for more details.

### 3.3.17 User Accounts & Groups

Prior to the installation of Oracle RAC Database 12c Release 1 (12.1), Oracle recommends the creation of a grid user for the Oracle Grid Infrastructure and an oracle user for the Oracle RAC Database software installed on the system. For the purposes of this reference environment, the Oracle Grid Infrastructure software owner is the user grid and the Oracle Database software owner is the user oracle. Each user is designated different groups to handle specific roles based on the software installed. However, the creation of separate users requires that both the oracle user and the grid user have a common primary group, the Oracle central inventory group (OINSTALL).

The following are the recommended system groups created for the installation of the Oracle RAC Database and part of the oracle user.

**OSDBA group (dba)** – determines OS user accounts with DBA privileges

**OSOPER group (oper)** – an optional group created to assign limited DBA privileges (SYSOPER privilege) to particular OS user accounts

With the release of Oracle RAC Database 12c (12.1), Oracle introduces three additional database groups that are optional, but recommended. The recommended following groups provide the ability to separate administrative privileges for common database operations and limit privileged access of the OSDBA group (dba).

**OSBACKUPDBA group (BACKUPDBA)** – an optional group created to assign limited administrative privileges (SYSBACKUP privilege) to a user for database backup and recovery

**OSDGDBA group (DGDBA)** – an optional group created to assign limited administrative
privileges (\textit{SYSDG} privilege) to a user for administering and monitoring Oracle Data Guard

OSKMDBA group (\textit{KMDBA}) – an optional group created to assign limited administrative privileges (\textit{SYSKM} privilege) to a user for encryption key management when using Oracle Wallet Manager

The following are the recommended system groups created for the installation of the Oracle Grid Infrastructure and part of the grid user

OSDBA group (\textit{ASMDBA}) – provides administrative access to Oracle ASM instances

OSASM group (\textit{ASMADMIN}) – provides administrative access for storage files via the \textit{SYSASM} privilege

OSOPER group (\textit{ASMOPER}) – an optional group created to assign limited DBA privileges with regards to ASM to particular OS user accounts

As the root user on each node, create the following user accounts, groups, and group assignments using a consistent UID and GID assignments across your organization:

```
# groupadd --gid 54321 oinstall
# groupadd --gid 54322 dba
# groupadd --gid 54323 asmdba
# groupadd --gid 54324 asmoper
# groupadd --gid 54325 asmadmin
# groupadd --gid 54326 oper
# groupadd --gid 54327 backupdba
# groupadd --gid 54328 dgdba
# groupadd --gid 54329 kmdba

# useradd --uid 54321 --gid oinstall --groups dba,oper,asmdba,asmoper,\> backupdba,dgdba,kmdba oracle
# passwd oracle

# useradd --uid 54322 --gid oinstall --groups dba,asmadmin,asmdba,asmoper grid
# passwd grid
```

Verify the oracle and grid user on each Oracle RAC database cluster node correctly displays the appropriate primary and supplementary groups via the commands:

```
# id oracle
uid=54321(oracle) gid=54321(oinstall)
groups=54321(oinstall),54322(dba),54323(asmdba),54324(asmoper),54326(oper),
54327(backupdba),54328(dgdba),54329(kmdba)
# id grid
uid=54322(grid) gid=54321(oinstall)
groups=54321(oinstall),54322(dba),54323(asmdba),54324(asmoper),54325(asmadmin)
```

### 3.3.18 Setting Shell Limits for the Grid and Oracle User

Oracle recommends the following settings for the soft and hard limits for the number of open file descriptors (nofile), number of processes (nproc), and size of the stack segment (stack) allowed by each user respectively. The purpose of setting these limits is to prevent a system wide crash that could be caused if an application, such as Oracle, were allowed to exhaust all of the OS resources under an extremely heavy workload.
Create a file labeled `99-grid-oracle-limits.conf` within `/etc/security/limits.d/` as follows:

```bash
# touch /etc/security/limits.d/99-grid-oracle-limits.conf
```

**NOTE:** The reason that the `/etc/security/limits.conf` file is not directly modified is due to the order in which limit files are read in the system. After reading the `/etc/security/limits.conf` file, files within the `/etc/security/limits.d/` directory are read. If two files contain the same entry, the entry read last takes precedence. For more information visit Red Hat Article: “What order are the limit files in the limits.d directory read in?"¹⁶

Within the `/etc/security/limits.d/99-grid-oracle-limits.conf` file, add the following soft and hard limits for the `oracle` and `grid` user:

```
oracle soft nproc 16384
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft stack 10240
oracle hard stack 32768
grid soft nproc 16384
grid hard nproc 16384
grid soft nofile 1024
grid hard nofile 65536
grid soft stack 10240
grid hard stack 32768
```

Due to Bug 15971421¹⁷, the soft limit of nproc is not adjusted at runtime by the Oracle database. Due to this, if the nproc limit is reached, the Oracle database may become unstable and not be able to fork additional processes. A high enough value for the maximum number of concurrent threads for the given workload must be set, or use the hard limit value of 16384 as done above if in doubt.

**NOTE:** Modifications made to the `99-grid-oracle-limits.conf` file take effect immediately. However, please ensure that any previously logged in `oracle` or `grid` user sessions (if any) are logged out and logged back in for the changes to take effect.

As the root user, on each node of the Oracle RAC Database cluster, create a shell script labeled `oracle-grid.sh` within `/etc/profile.d/` to create the `ulimits` for the `oracle` and `grid` user. The contents of the `oracle-grid.sh` script:

```
#Setting the appropriate ulimits for oracle and grid user
if [ $USER = "oracle" ]; then
  if [ "$SHELL = "/bin/ksh" ]; then
    ulimit -u 16384
    ulimit -n 65536
  else
    ulimit -u 16384 -n 65536
  fi
```

---

¹⁶ What order are the limits files in the limits.d directory read in? - [https://access.redhat.com/site/solutions/199993](https://access.redhat.com/site/solutions/199993)
¹⁷ Oracle Documentation 12.1 - [http://docs.oracle.com/cd/E16655_01/readmes.121/e17908/toc.htm#BABICCBB](http://docs.oracle.com/cd/E16655_01/readmes.121/e17908/toc.htm#BABICCBB)
NOTE: While the `ulimit` values can be set directly within the `/etc/profile` file, it is recommended to create a custom shell script within `/etc/profile.d` instead. The `oracle-grid.sh` script can be downloaded from the Appendix M Configuration Files.
As oracle and grid user, verify the \textit{ulimit} values by running the following command:

```
# ulimit -a

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>core file size</td>
<td>(blocks, -c) 0</td>
</tr>
<tr>
<td>data seg size</td>
<td>(kbytes, -d) unlimited</td>
</tr>
<tr>
<td>scheduling priority</td>
<td>(e) 0</td>
</tr>
<tr>
<td>file size</td>
<td>(blocks, -f) unlimited</td>
</tr>
<tr>
<td>pending signals</td>
<td>(-i) 385878</td>
</tr>
<tr>
<td>max locked memory</td>
<td>(kbytes, -l) 14854144</td>
</tr>
<tr>
<td>max memory size</td>
<td>(kbytes, -m) unlimited</td>
</tr>
<tr>
<td>open files</td>
<td>(-n) 65536</td>
</tr>
<tr>
<td>pipe size</td>
<td>(512 bytes, -p) 8</td>
</tr>
<tr>
<td>POSIX message queues</td>
<td>(bytes, -q) 819200</td>
</tr>
<tr>
<td>real-time priority</td>
<td>(-r) 0</td>
</tr>
<tr>
<td>stack size</td>
<td>(kbytes, -s) 10240</td>
</tr>
<tr>
<td>cpu time</td>
<td>(seconds, -t) unlimited</td>
</tr>
<tr>
<td>max user processes</td>
<td>(-u) 16384</td>
</tr>
<tr>
<td>virtual memory</td>
<td>(kbytes, -v) unlimited</td>
</tr>
<tr>
<td>file locks</td>
<td>(-x) unlimited</td>
</tr>
</tbody>
</table>
```
3.4 Storage Configuration

The following storage configuration section describes the best practices for setting up device mapper multipath, the use of udev rules for ASM disk management, and the use of the tuned package for optimal performance.

3.4.1 Setting up DM-Multipath

Device mapper multipath provides the ability to aggregate multiple I/O paths to a newly created device mapper mapping to achieve high availability, I/O load balancing, and persistent naming. The following procedures provide the best practices to installing and configuring device mapper multipath devices.

NOTE: Ensure Oracle RAC database volumes are accessible via the operating system on all nodes within the Oracle RAC Database cluster prior to continuing with the section below.

The following instructions are required on each node within the Oracle RAC Database 12c cluster.

1. As the root user, install the device-mapper-multipath package using the yum package manager.
   ```bash
   # yum install device-mapper-multipath
   ```

2. Copy the multipath.conf file found within /usr/share/doc/device-mapper-multipath-0.4.9/ to /etc/
   ```bash
   # cp /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf /etc/
   ```

3. Capture the scsi id of the local disk(s) on the system.
   ```bash
   # /usr/lib/udev/scsi_id --whitelisted --replace-whitespace --device=/dev/sda
   3600508b1001030353434363646301200
   ```

4. Uncomment and modify the blacklist section within the /etc/multipath.conf file to include the scsi id of the local disk on the system. Once complete, save the changes made to the multipath.conf file.
   ```bash
   blacklist {
       wwid 3600508b1001030353434363646301200
       devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
       devnode "^hd[a-z]"
   }
   ```

5. Start the multipath daemon.
   ```bash
   # systemctl start multipathd.service
   ```

6. Enable the multipath daemon to ensure it is started upon boot time.
   ```bash
   # systemctl enable multipathd.service
   ```
7. Identify the `dm`- device, size, and WWID of each device mapper volume for Oracle data disks and recovery disks. In this example, volume `mpathb` is identified via the following command:

```
# multipath -ll
```

![Figure 3.4.1.1: Multipath Device (mpathb)](image)

**Figure 3.4.1.1: Multipath Device (mpathb)**

**Figure 3.4.1.1: Multipath Device (mpathb)** properly identifies the current multipath alias name, size, WWID, and dm device. This information is required for the application of a custom alias to each volume as shown in step 9.

**NOTE:** Due to the size of some of the disks being the same, compare the WWID found on the system with the WWID located on the storage array to ensure the multipath alias name assigned corresponds with the name of the volume from the storage array.

8. Uncomment the defaults section found within the `/etc/multipath.conf` file.

```bash
defaults {
    udev_dir                /dev
    polling_interval        10
    path_selector           "round-robin 0"
    path_grouping_policy    multibus
    getuid_callout          "/lib/udev/scsi_id --whitelisted
    --device=/dev/%n"
    prio                    alua
    path_checker            readsector0
    rr_min_io               100
    max_fds                 8192
    rr_weight               priorities
    failback                immediate
    no_path_retry           fail
    user_friendly_names     yes
}
```

refarch-feedback@redhat.com 41 www.redhat.com
NOTE: The standard options can be customized to better fit your storage array capabilities. Check with your storage vendor for details.

9. Uncomment the multipath section found within the `/etc/multipath.conf` file and create an alias for each `device mapper` volume in order to enable persistent naming of those volumes. Once complete, save the changes made to the `multipath.conf` file. The output should resemble the example below. For reference, refer to the Oracle data volumes created for this reference environment displayed in Table 2.4.3: Oracle OCR, Voting Disk, & Data File Sizes.

```plaintext
multipaths {
    multipath {
        wwid                    3600c0ff000d7a899d8515101000000
        alias                   db1
    }
    multipath {
        wwid                    3600c0ff000dabfe5a7d8515101000000
        alias                   db2
    }
    multipath {
        wwid                    3600c0ff000d7e7a8dbd8515101000000
        alias                   fra
    }
    multipath {
        wwid                    3600c0ff000d7e7a8dbd8515101000000
        alias                   redo
    }
}
```

10. Restart the `device mapper multipath` daemon.

   ```
   # systemctl restart multipathd.service
   ```

11. Verify the `device mapper` paths and aliases are displayed properly. Below is an example of one `device mapper` device labeled `fra`.

   ```
   # multipath -ll
   fra (3600c0ff000d7e7a89e85ac5101000000) dm-10 HP,MSA2324fc
   size=186G features='1 queue_if_no_path' hwhandler='0' wp=rw
   | -+- policy='round-robin 0' prio=130 status=active
   |   |  3:0:0:3 sdd 8:48  active ready running
   |   |  3:0:1:3 sdh 8:112 active ready running
   |   -|  4:0:0:3 sdt 65:48 active ready running
   |   `-|  4:0:1:3 sdx 65:112 active ready running
   `-- policy='round-robin 0' prio=10 status=enabled
      -|  3:0:2:3 sdh 8:176 active ready running
      -|  3:0:3:3 sdp 8:240 active ready running
      -|  4:0:2:3 sdab 65:176 active ready running
      `- 4:0:3:3 sdaf 65:240 active ready running
   ```

12. Repeat all above steps for all nodes within the Oracle RAC Database cluster.
3.4.2 Partitioning Device Mapper Shared Disks

On the first node of the Oracle RAC Database cluster, create a partition for each device mapper volume \((ocrvote1, ocrvote2, ocrvote3, db1, db2, fra, redo)\) using `parted` as displayed below for device `db1`.

```
# parted /dev/mapper/db1 mklabel gpt mkpart primary "1 -1"
Information: You may need to update /etc/fstab.
```

Once the partition is created, a newly created device mapper device is created as `db1p1`.

```
# ls -l /dev/mapper/db1p1
lrwxrwxrwx. 1 root root 8 Apr 16 15:15 /dev/mapper/db1p1 -> ../dm-11
```

**NOTE:** A newly created partition alias name ending in a number i.e. `db1` requires the alias name followed by `p1` such as `db1p1` seen above. If `p1` is missing, please run the following `kpartx` command to add the partition mappings to the device mapper disks.

```
# kpartx -a /dev/mapper/db1
```

**NOTE:** If a newly created partition alias name ends in a letter i.e. `fra`, the alias name will be followed by just the partition number, i.e. `fra1`.

Once the partitions are created, on all the nodes in the Oracle RAC Database cluster, run the following `kpartx` command on each device mapper volume in order to update the partition table mapping.

**NOTE:** If the following `kpartx` command does not add the `p1` suffix to the partitions ending in a number, reboot the system.

3.4.3 Configuring Oracle ASM Disks

The configuration of Oracle ASM requires the use of either `udev` rules or Oracle's `ASMLib`. Oracle's `ASMLib` is an optional utility used to manage and assist users with Oracle ASM devices and is not required for proper operation of Oracle ASM disks. Moreover, Oracle `ASMLib` does not have any impact on Oracle Database performance and requires a kernel module labeled `kmod-oracleasm` and proprietary user space utilities that need to be built for specific kernel versions to properly function. `udev` rules represent an alternative to Oracle's `ASMLib` and does not require additional kernel modules or a specific kernel version, thus keeping an overall smaller footprint on the Linux system and allowing more flexibility when upgrading kernel versions. This reference architecture only documents Red Hat's native `udev` rules as at the time of this writing Oracle does not provide an `oracleasmlib` package for Red Hat Enterprise Linux 7.

3.4.3.1 Oracle ASMLib Alternative: Configuring udev Rules

The configuration of Oracle ASM requires the use of either `udev` rules or Oracle's `ASMLib` (currently not available on Red Hat Enterprise Linux 7). This section, focuses on the best practices of using Red Hat's native `udev` rules to setup the appropriate permissions for each
On the first node of the Oracle RAC cluster as the root user, identify the Device Mapper Universally Unique IDentifier (DM_UUID) for each device mapper volume. The example below shows the DM_UUID for the partitions of the volumes labeled ocrvote1p1, ocrvote2p1, ocrvote3p1, db1p1, db2p1, fra1, and redo1.

```bash
# for i in ocrvote1p1 ocrvote2p1 ocrvote3p1 db1p1 db2p1 fra1 redo1; do printf "\%s %s\n" "$i" "$(udevadm info --query=all --name=/dev/mapper/$i | grep -i dm_uuid)"; done
ocrvote1p1 E: DM_UUID=part1-mpath-3600c0ff000dabfe596a0f6510100000
ocrvote2p1 E: DM_UUID=part1-mpath-3600c0ff000dabfe5a2a0f6510100000
ocrvote3p1 E: DM_UUID=part1-mpath-3600c0ff000dabfe5b4a0f6510100000
db1p1 E: DM_UUID=part1-mpath-3600c0ff000d7e7a899d851510100000
db2p1 E: DM_UUID=part1-mpath-3600c0ff000dabfe5a7d851510100000
fra1 E: DM_UUID=part1-mpath-3600c0ff000d7e7a8dbd851510100000
redo1 E: DM_UUID=part1-mpath-3600c0ff000dabfe5f4d851510100000
```

2. Create a file labeled 99-oracle-asmdevices.rules within /etc/udev/rules.d/

3. Within 99-oracle-asmdevices.rules file, create rules for each device similar to the example below:

```bash
KERNEL=="dm-*",ENV{DM_UUID}=="part1-mpath-3600c0ff000dabfe5f4d851510100000",OWNER="grid",GROUP="asmadmin",MODE="0660"
```

To understand the rule above, it can be read as follows:

If any dm- device (dm-*) matches the DM_UUID of part1-mpath-3600c0ff000dabfe5f4d851510100000, assign that dm- device to be owned by the grid user and part of the ASMADMIN group with the permission mode set to 0660. The 0660 value provides read and write permissions to the user grid and owner asmadmin.

4. Save the file labeled 99-oracle-asmdevices.rules

5. Copy the 99-oracle-asmdevices.rules file to each node within the Oracle RAC Database cluster using the scp command and enter the appropriate password credentials for the other nodes. The example below shows how to copy the file to node two of the Oracle RAC Database 12c cluster.

```bash
# scp /etc/udev/rules.d/99-oracle-asmdevices.rules db-oracle-node2:/etc/udev/rules.d/
root@db-oracle-node2's password: 99-oracle-asmdevices.rules                  100%  834     0.8KB/s   00:00
```

6. On each node within the Oracle RAC Database cluster, locate the dm- device for each Oracle related partition. An example of how to find the dm- device for each partition is to run the following command:

```bash
# for i in db1p1 db2p1 fra1 redo1 ocrvote1p1 ocrvote2p1 ocrvote3p1; do printf "\%s %s\n" "$i" "$(ls -ll /dev/mapper/$i)"; done
db1p1 lrwxrwxrwx. 1 root root 8 May 20 20:39 /dev/mapper/db1p1 -> ../dm-11
db2p1 lrwxrwxrwx. 1 root root 8 May 20 20:39 /dev/mapper/db2p1 -> ../dm-12
```
7. On each node within the Oracle RAC Database cluster, apply and test the rules created within the `99-oracle-asmdevices.rules` by running a `udevadm test` on each device. The example below demonstrates a `udevadm test` on `dm-11`.

```bash
# udevadm test /sys/block/dm-11
[ ... Output Abbreviated ... ]
udevadm_test: DM_NAME=db1p1
udevadm_test: DM_UUID=part1-mpath-3600c0ff000d7e7a86485ac5101000000
udevadm_test: DM_SUSPENDED=0
udevadm_test: DEVLINKS=/dev/mapper/db1p1 /dev/disk/by-id/dm-name-db1p1
/dev/disk/by-id/dm-uuid-part1-mpath-3600c0ff000d7e7a86485ac5101000000
/dev/block/253:11
udevadm_test: ID_FS_TYPE=oracleasm
```

8. Confirm each device has the desired permissions. Example of `db1p1 -> dm-11`, with owner set to grid and group set to asmadmin.

```bash
# ls -lh /dev/dm-11
brw-rw----. 1 grid asmadmin 253, 11 Jun 6 20:59 /dev/dm-11
```

**NOTE:** If the desired permissions are not visible, please reboot the particular node from the Oracle RAC Database cluster.

**NOTE:** For simplicity, this `99-oracle-asmdevices.rules` file is included in Appendix J `99-oracle-asmdevices.rules`.

### 3.4.4 Optimizing Database Storage using Automatic System Tuning

The `tuned` package in Red Hat Enterprise Linux 7 is recommended for automatically tuning the system for common workloads via the use of profiles. Each profile is tailored for different workload scenarios such as: throughput performance, balanced, & high network throughput.

It is recommended to create a custom profile (oracle) with similar settings as the throughput-performance profile but that disables Transparent HugePages (THP) and sets the kernel parameters for Oracle RAC databases workload environments. For more information on why THP is disabled, see Section 4.1.5 Enabling HugePages. The table Table 3.4.4.1: Profile Comparison provides details between the balanced (default) profile, throughput-performance profile, and the custom profile labeled oracle.
<table>
<thead>
<tr>
<th>Tuned Parameters</th>
<th>balanced</th>
<th>throughput-performance</th>
<th>oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Elevator</td>
<td>deadline</td>
<td>deadline</td>
<td>deadline</td>
</tr>
<tr>
<td>CPU governor</td>
<td>OnDemand</td>
<td>performance</td>
<td>performance</td>
</tr>
<tr>
<td>kernel.sched_min_granularity_ns</td>
<td>auto-scaling</td>
<td>10ms</td>
<td>10ms</td>
</tr>
<tr>
<td>kernel.sched_wake_up_granularity_ns</td>
<td>3ms</td>
<td>15ms</td>
<td>15ms</td>
</tr>
<tr>
<td>Disk read-ahead</td>
<td>128 KB</td>
<td>4096 KB</td>
<td>4096 KB</td>
</tr>
<tr>
<td>vm.dirty_ratio</td>
<td>20%</td>
<td>40%</td>
<td>80%</td>
</tr>
<tr>
<td>File-system barrier</td>
<td>on</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>Transparent HugePages</td>
<td>on</td>
<td>on</td>
<td>Off</td>
</tr>
<tr>
<td>vm.dirty_background_ratio</td>
<td>10%</td>
<td>10%</td>
<td>3%</td>
</tr>
<tr>
<td>vm.swappiness</td>
<td>60%</td>
<td>10%</td>
<td>1%</td>
</tr>
<tr>
<td>energy_perf_bias</td>
<td>Normal</td>
<td>Performance</td>
<td>Performance</td>
</tr>
<tr>
<td>min_perf_pct (intel_pstate_only)</td>
<td>Auto-scaling</td>
<td>Auto-scaling</td>
<td>Auto-scaling</td>
</tr>
<tr>
<td>tcp_rmem_default</td>
<td>Auto-scaling</td>
<td>Auto-scaling</td>
<td>262144</td>
</tr>
<tr>
<td>tcp_wmem_default</td>
<td>Auto-scaling</td>
<td>Auto-scaling</td>
<td>262144</td>
</tr>
<tr>
<td>udp_mem (pages)</td>
<td>Auto-scaling</td>
<td>Auto-scaling</td>
<td>Auto-scaling</td>
</tr>
<tr>
<td>vm.dirty_expre_centisecs</td>
<td>-</td>
<td>-</td>
<td>500</td>
</tr>
<tr>
<td>vm.dirty_writeback_centisecs</td>
<td>-</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>Kernel.shmmax</td>
<td>-</td>
<td>-</td>
<td>4398046511104</td>
</tr>
<tr>
<td>Kernel.shmall</td>
<td>-</td>
<td>-</td>
<td>1073741824</td>
</tr>
<tr>
<td>Kernel.sem</td>
<td>-</td>
<td>-</td>
<td>250 32000 100 128</td>
</tr>
<tr>
<td>fs.file-max</td>
<td>-</td>
<td>-</td>
<td>6815744</td>
</tr>
<tr>
<td>fs.aio-max-nr</td>
<td>-</td>
<td>-</td>
<td>1048576</td>
</tr>
<tr>
<td>ip_local_port_range</td>
<td>-</td>
<td>-</td>
<td>9000 65500</td>
</tr>
<tr>
<td>tcp_rmem_max</td>
<td>-</td>
<td>-</td>
<td>4194304</td>
</tr>
<tr>
<td>tcp_wmem_max</td>
<td>-</td>
<td>-</td>
<td>1048576</td>
</tr>
</tbody>
</table>

18 The value explicitly set within the /etc/sysctl.conf file has precedence over values set by tuned. Oracle values set in /etc/sysctl.conf are done to satisfy prerequisites set by Oracle’s installer and will be removed post install and reside only within the oracle tuned profile.
The following procedures provide the steps that are required to install, enable, and select the oracle profile.

On each node within the Oracle RAC Database cluster, as the root user,

1. Install the tuned package via the yum package manager.
   
   ```
   # yum install tuned
   ```

2. Enable tuned to ensure it is started upon boot time.
   
   ```
   # systemctl enable tuned.service
   ```

3. Start the tuned service
   
   ```
   # systemctl start tuned.service
   ```

4. Download the accompanying tar.gz file for this reference architecture and copy the oracle folder to the /usr/lib/tuned directory. For simplicity, Appendix L Custom Oracle Tuned Profile (script.sh & tuned.conf) provides the two files script.sh and tuned.conf that reside within the oracle profile directory.
   
   ```
   # cd /path/to/oracle/profile
   # cp -r oracle /usr/lib/tuned/
   ```

5. Activate the newly added oracle profile:
   
   ```
   # tuned-adm profile oracle
   ```

6. Verify that THP is now disabled via the following command:
   
   ```
   # cat /sys/kernel/mm/transparent_hugepage/enabled
   always madvise [never]
   ```

7. Disable transparent huge pages persistently across reboots by adding to the kernel boot command line within the /etc/default/grub and add within the GRUB_CMDLINE_LINUX the following:

   ```
   cat /etc/default/grub
   GRUB_TIMEOUT=5
   GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
   GRUB_DEFAULT=saved
   GRUB_DISABLE_SUBMENU=true
   GRUB_TERMINAL_OUTPUT="console"
   GRUB_CMDLINE_LINUX="rd.lvm.lv=myvg/swap rd.lvm.lv=myvg/usr
   vconsole.font=latarcyrheb-sun16 rd.lvm.lv=myvg/root crashkernel=auto
   vconsole.keymap=us rhgb quiet transparent_hugepage=never"
   GRUB_DISABLE_RECOVERY="true"
   ```
8. For the grub changes to take effect, run the following command:

```bash
# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-123.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-41c535c189b842eea5a8c20cbd9bff26
Found initrd image: /boot/initramfs-0-rescue-41c535c189b842eea5a8c20cbd9bff26.img
done
```

For simplicity, the `oracle` profile consisting of two files, `script.sh` & `tuned.conf` is included in Appendix M Configuration Files

**NOTE:** If at any point in time a revert to the original settings are required with persistence across reboots, the following commands can be run on each node within the Oracle RAC Database cluster:

```
# systemctl stop tuned.service
# systemctl disable tuned.service
```

**NOTE:** Even if reverting to the original settings, it is recommended to keep `transparent` huge pages disabled within the `/etc/default/grub` file.
4 Oracle 12c R1 Configuration

4.1.1 Installing Oracle Grid Infrastructure (Required for ASM)

The installation of the Oracle Grid Infrastructure for Oracle RAC Database 12c Release 1 (12.1) is required for the use of Oracle ASM. Prior to the installation of the Oracle Grid Infrastructure, ensure that the prerequisites from the following sections have been met:

- Section 2 Reference Architecture Environment
- Section 3 Reference Architecture Configuration Details

NOTE: The reference environment uses the /u01/app/grid as the grid base. The owner is set to grid and the group is set to OINSTALL. Run the following commands to create the grid base directory and set the appropriate permissions:

On each node within the Oracle RAC environment as the root user, create the following directory structure and set the proper permissions.

```
# mkdir --parents /u01/app/grid
# chown --recursive grid.oinstall /u01/
```

The following steps are intended only for node one of the Oracle RAC Database environment unless otherwise specified.

1. Download the two Oracle Grid Infrastructure software files\(^{19}\) from the Oracle Software Delivery Cloud site\(^{20}\).

2. As the grid user, create a temporary directory within /u01/app/grid/grid-software to store the Oracle Grid Software zip file, move the Oracle Grid Software zip file to the /u01/app/grid/grid-software location, ensure the Oracle Grid Software zip has the proper permissions and unpack its contents.

```
As the grid user,
  # mkdir /u01/app/grid/grid-software
  # mv V46096-01_1of2.zip /u01/app/grid/grid-software
  # mv V46096-01_2of2.zip /u01/app/grid/grid-software
```

As the root user,

```
  # chown grid.oinstall V46096-01_1of2.zip
  # chown grid.oinstall V46096-01_2of2.zip
```

As the grid user,

```
  # cd /u01/app/grid/grid-software
  # unzip V46096-01_1of2.zip
  # unzip V46096-01_2of2.zip
```

\(^{19}\) Oracle Database 12c Release 1 Grid Infrastructure (12.1.0.2.0) - V46096-01_1of2.zip and V46096-01_2of2.zip from http://edelivery.oracle.com
3. As the grid user, start the OUI via the command:

```
# /u01/app/grid/grid-software/grid/runInstaller
```

**NOTE:** Ensure to SSH with the -Y option as the grid user from the client server. Otherwise, the following error will occur.

```
# /u01/app/grid/grid-software/grid/runInstaller
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 415 MB. Actual 3689 MB Passed
Checking swap space: must be greater than 150 MB. Actual 16415 MB Passed
Checking monitor: must be configured to display at least 256 colors
>>> Could not execute auto check for display colors using command /usr/bin/xdpyinfo. Check if the DISPLAY variable is set. Failed <<<

Some requirement checks failed. You must fulfill these requirements before continuing with the installation,

Continue? (y/n) [n] n
```

1. Within the **Installation Option** window, select **Install and Configure Oracle Grid Infrastructure for a Cluster** and click **Next**.

2. Within the **Cluster Type** window, select between **Configure a Standard cluster** or **Configure a Flex cluster** and click **Next**. This reference architecture focuses on using a standard cluster.

3. Within the **Installation Type** window, select **Advanced Installation** and click **Next**.

4. Within the **Product Languages** window, select the appropriate language, and click **Next**.

5. Within the **Grid Plug and Play Information** window, provide the appropriate credentials for the SCAN and click **Next**. This reference architecture provides the following SCAN credentials:
   - Cluster Name: `db-ora-cluster`
   - SCAN Name: `db-oracle-scan.cloud.lab.eng.bos.redhat.com`
   - SCAN Port: 1521
   - Uncheck **Configure GNS**

   **NOTE:** The SCAN Name is the name registered with the DNS Server as seen in Section 3.2.3 Configure SCAN via DNS
**Figure 4.1.1.1: Oracle Plug and Play Window**

Single Client Access Name (SCAN) allows clients to use one name in connection strings to connect to the cluster as a whole. Client connect requests to the SCAN Name can be handled by any cluster node.

- **Cluster Name:** db-ers-cluster
- **SCAN Name:** e-scan.cloud.lab.ing.bos.redhat.com
- **SCAN Port:** 1521

- **Configure DNS**
  - [ ] Configure nodes Virtual IPs as assigned by the Dynamic Networks
  - [ ] Create a new DNS
    - DNS VIP Address: 
    - Domain Name: db-ers.cloud.lab.ing.bos.redhat.com
  - [ ] Use Shared DNS
    - DNS Client Data: 

- **Finish**
6. Within the same **Cluster Node Information** window, click the Add button to add each node within the Oracle RAC Database cluster and click OK. Each node within the Oracle RAC cluster requires the public hostname and VIP information as seen in Figure 4.1.1 Figure 4.1.1.2: Cluster Node Information Window, Add button

7. Within the same **Cluster Node Information** window, select the SSH Connectivity button to set the passwordless SSH connectivity by entering the OS Password: credentials for the grid user and clicking Setup. Once a dialog box returns with the 'Successfully established passwordless SSH connectivity between the selected nodes', click OK and click Next to continue to the next window.

8. Within the **Network Interface Usage** window, select the Interface Name, bond0, to be set as the Interface Type Public and the Interface Name, ens10f2 and ens10f3, to be set as the Interface Type Private. Any other Interfaces should be set to Do Not Use. Select Next to continue.

9. Within the **Storage Option** window, this reference architecture selects Use Standard ASM for storage and click Next.

10. Within the **Create ASM Disk Group** window, provide the following:
• a Disk Group Name, i.e. **OCRVOTE**

• Redundancy Level
  ◦ **EXTERNAL** – redundancy provided by the storage system RAID, and not by Oracle ASM.
  ◦ **NORMAL** – provides two-way mirroring by Oracle ASM, thus provided two copies of every data extent.
  ◦ **HIGH** – provides three-way mirroring by Oracle ASM thus enduring the loss of two ASM disks within different failure groups.

The following **Table 4.1.1 Table 4.1.1.1: ASM Disk Group Window** provides the details for the reference environment.

<table>
<thead>
<tr>
<th>ASM Disk Group Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Group Name</td>
</tr>
<tr>
<td>Redundancy Level</td>
</tr>
<tr>
<td>Device Mapper multipath and udev rules, Disks Assigned to ASM Disk Group</td>
</tr>
</tbody>
</table>

**Table 4.1.1.1: ASM Disk Group Window**
• Allocation Unit (AU) Size set to 4 MB
  • A 4 MB AU Size is used to decrease the amount of extents Oracle needs to manage. With less extents to manage, CPU utilization and memory consumption is reduced thus improving performance. The AU Size varies depending on the type of Oracle workload, I/O size per transaction, and overall diskgroup size. There is no “best size” for AU size, but a good starting point is a 4 MB. Please ensure to visit Oracle's documentation for more information.

To display the appropriate candidate disks, click on the Change Discovery Path button and enter as the Disk Discovery Path one of the following as appropriate:
  • For Device Mapper devices, type: /dev/mapper/*

![Create ASM Disk Group Window](image)

Figure 4.1.1.3: Create ASM Disk Group Window

---

20 Oracle ASM Extents - [http://docs.oracle.com/cd/E16655_01/server.121/e17612/asmcon.htm#OSTMG94063](http://docs.oracle.com/cd/E16655_01/server.121/e17612/asmcon.htm#OSTMG94063)
NOTE: There is a significant size increase for the OCR and Voting Disk Diskgroup. Starting with Oracle Database 12.1.0.2.0, Oracle has included the installation of the Grid Infrastructure Management Repository. For a two node Oracle RAC as shown within this reference architecture, a minimum of 11059 MB of space is required when using Normal Redundancy. If using External Redundancy, only 5528 MB required.

11. Click Next once complete within the Create ASM Disk Group window

12. Within the ASM Password window, specify the password for the SYS and ASMSNMP user accounts.

13. Within the Failure Isolation Support window, select whether to use or not use the Intelligent Platform Management Interface (IPMI). This reference architecture selects the Do not use Intelligent Platform Management Interface (IPMI) radio button and clicks Next.

14. Within the Management Options window, Register with Enterprise Manager (EM) Cloud Control is unchecked, click Next

15. Within the Operating System Groups window, select the appropriate OS groups. The values as created and assigned within this reference environment are as follows:
   - Oracle ASM Administrator Group – ASADMIN
   - Oracle ASM DBA Group – ASMDBA
   - Oracle ASM Operator Group – ASMOPER

11. Within the Installation Location window, specify the appropriate Oracle base and software locations. The values set by this reference environment are as follows:
   - ORACLE_BASE - /u01/app/grid
   - SOFTWARE LOCATION - /u01/app/grid/product/12.1.0/grid

12. Within the Create Inventory window, specify the inventory directory. The values set by this reference environment are as follows:
   - Inventory Directory - /u01/app/oraInventory

13. Within the Root script execution configuration window, select the check box labeled Automatically run configuration scripts and enter the root user credentials. The step specifying the root user credentials in order to run specific configuration scripts automatically at the end of the installation is optional. For the purposes of this reference environment, the root credentials are given in order to speed up the Oracle Grid Infrastructure installation process.

14. Within the Prerequisite Checks window, review the status and ensure there are no errors prior to continuing the installation. For failures with a status set to Fixable, select the Fix & Check Again button. The execution of the Fix & Check Again button provides a runfixup.sh script provided by the OUI. If selected Automatically run configuration scripts from the previous step, the Oracle OUI uses the root credentials and runs the
Run the `runfixup.sh` script automatically. Otherwise, as `root`, on each node within the Oracle RAC cluster, run the `/tmp/CVU_12.1.0.2.0_grid/runfixup.sh` and click on the **Check Again** button once the `runfixup.sh` has finished. For errors that can be ignored, select the **Ignore All** check box and click **Next**.

The following check errors are common and extra details are provided below.

- **Task resolv.conf Integrity** – This task checks consistency of the file `/etc/resolv.conf` across all nodes. Manually verify that all nodes across the Oracle RAC Database cluster provide the appropriate response with the use of `nslookup`. If the appropriate response is achieved, this error can be safely ignored.

- **Network Time Protocol (NTP)** – This task verifies cluster time synchronization on clusters. Manually verify that `ntp` is running on all nodes within the Oracle RAC cluster. Please review [Section 3.3.2 NTP Configuration](#) for more details on properly setting up NTP. If NTP is properly running and configured, this error can be safely ignored.

- **Device Checks for ASM** – This task verifies that specified devices meet the requirements for Oracle ASM. Once the prerequisite of `cvuqdisk` rpm is installed, this warning will disappear.

- **/dev/shm mounted as temporary file system** – Mounting of `/dev/shm` is required when using Automatic Memory Management (AMM). Due to this reference environment, taking advantage of HugePages, `/dev/shm` is not used. This warning can be safely ignored

15. Within the **Summary** window, review all the information provided, and select **Install** to start the installation.

16. During the installation process, within the **Oracle Grid Infrastructure** pop up window, select **yes** to allow the installer to run as the `root` user to execute the configuration scripts.

**NOTE:** Oracle Cluster Verification Utility will provide a **Failed** status due to NTP task error. Click **Next** within the **Install Product** window, and select **Yes** within the **Oracle Grid Infrastructure 12c Release 1 Installer** popup window.

17. Within the **Finish** window, verify the installation was successful and click **Close**.

18. Verify the installation was successful using the `crsctl` command as follows:

As the grid user:

```bash
# $GRID_HOME/bin/crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
```

**NOTE:** The Grid Home used in this reference environment is: `/u01/app/12.1.0/grid`
4.1.2 Installing Oracle 12c R1 Database Software

Prior to the installation of the Oracle RAC Database 12c Release 1 (12.1), ensure the following prerequisites from the following sections have been met:

- Section 2 Reference Architecture Environment
- Section 3 Reference Architecture Configuration Details

**NOTE:** The reference environment uses the `/u01/app/oracle` as the Oracle base. The owner is set to `oracle` and the group is set to `OINSTALL`. Run the following commands to create the oracle base directory set the appropriate permissions:

On each node within the Oracle RAC environment, as the root user, create the following directory structure and set the proper permissions.

```bash
# mkdir /u01/app/oracle
# chown --recursive oracle.oinstall /u01/app/oracle
```

The following steps are intended for only node one of the Oracle RAC Database 12c environment unless otherwise specified.

1. Download the two Oracle Database software files from the Oracle Software Delivery Cloud site.

2. As the oracle user, create a temporary directory within `/u01/app/oracle/oracle-software` to store the Oracle Database Software zip files, move the Oracle Database software zip files to the `/u01/app/oracle/oracle-software` location, ensure the Oracle Database software zip has the proper permissions and unpack its contents.

   As the oracle user,

   ```bash
   # mkdir /u01/app/oracle/oracle-software
   # mv V46095-01_1of2.zip V46095-01_2of2.zip /u01/app/oracle/oracle-software/
   # cd /u01/app/oracle/oracle-software
   # unzip V46095-01_1of2.zip
   # unzip V46095-01_2of2.zip
   ```

   As the root user,

   ```bash
   # chown oracle.oinstall /u01/app/oracle/oracle-software/V46095-01_1of2.zip
   # chown oracle.oinstall /u01/app/oracle/oracle-software/V46095-01_2of2.zip
   ```

3. As the oracle user, start the OUI via the command:

   ```bash
   # /u01/app/oracle/oracle-software/database/runInstaller
   ```

---

22 Oracle Database 12c Release 1 (12.1.0.2.0) - [V46095-01_1of2.zip](http://edelivery.oracle.com) and [V46095-01_2of2.zip](http://edelivery.oracle.com) from [http://edelivery.oracle.com](http://edelivery.oracle.com)
NOTE: Ensure to issue SSH with the -Y option as the oracle user from the client server. Otherwise, the following error will occur.

```
# /u01/app/oracle/oracle-software/database/runInstaller
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 500 MB. Actual 3461 MB Passed
Checking swap space: must be greater than 150 MB. Actual 20479 MB Passed
X11 connection rejected because of wrong authentication.
X11 connection rejected because of wrong authentication.
Checking monitor: must be configured to display at least 256 colors
>>> Could not execute auto check for display colors using command /usr/bin/xdpyinfo. Check if the DISPLAY variable is set. Failed <<<

Some requirement checks failed. You must fulfill these requirements before continuing with the installation,

Continue? (y/n) [n] n
```

4. Within the **Configure Security Updates** window, provide the My Oracle Support email address for the latest security issues information. Otherwise uncheck the **I wish to receive security updates via My Oracle Support** and click **Next**.
5. Within the **Installation Option** window, select **Install database software only** and click **Next**.

![Installation Option Window](image)

**Figure 4.1.2.1: Installation Option Window**
6. Within the Grid Installation Options window, select Oracle Real Application Clusters database installation as the type of database installation being performed.

Figure 4.1.2.2: Grid Installation Options Window
7. Within the **Select List of Nodes** window, ensure all nodes for the Oracle RAC database cluster are checked and click on the **SSH Connectivity** button. Within the **OS Password:** dialog box enter the user `oracle`'s password and click **Setup**. Once a dialog box returns with 'Successfully established passwordless SSH connectivity between the selected nodes', click **OK** and **Next** to continue.

![Select List of Nodes Window](image)

**Figure 4.1.2.3: Select List of Nodes Window**

8. Within the **Product Languages** window, select the appropriate language for the installation.

9. Within the **Database Edition** window, select the **Enterprise Edition** and click **Next**.

10. Within the **Installation Location** window, select the appropriate Oracle base and software location and click **Next**. For the purposes of this reference environment, the following values are set:

    - **Oracle Base** - `/u01/app/oracle`
    - **Software Location** - `/u01/app/oracle/product/12.1.0/dbhome_1`
11. Within the **Operating System Groups** window, select the appropriate OS groups and click **Next**. For the purposes of this reference environment, the following values are set as:

- Database Administrator Group — **DBA**
- Database Operator Group — **OPER**
- Database Backup and Recovery Group — **BACKUPDBA**
- Data Guard Administrative group — **DGDBA**
- Encryption Key Management Administrative group — **KMDBA**

12. Within the **Prerequisite Checks** window, review the status and ensure there are no errors prior to continuing the installation. For failures with a status set to Fixable, select the **Fix & Check Again** button. The execution of the **Fix & Check Again** button provides a `runfixup.sh` script provided by the OUI. As root, run the `runfixup.sh` and click on the **Check Again** button once the `runfixup.sh` has finished. For errors that can be ignored, select the Ignore All check box and click **Next**.

- The following check errors are common and extra details are provided below.
  - Task `resolv.conf` Integrity – This task checks consistency of the file `/etc/resolv.conf` across all nodes. Manually verify that all nodes across the Oracle RAC Database cluster provide the appropriate response with the use of `nslookup`. If the appropriate response is achieved, this error can be safely ignored.
  - Clock Synchronization – This task checks to see if NTP daemon or service is running. If the service is not running, it provides this error and displays a PRVF-5415 error. However, the OUI contains a bug that does not see the NTP service is actually running and provides a not running status. Manually verify that all nodes across the Oracle RAC Database cluster are running the `ntpd` service. Please review **Section 3.3.2 NTP Configuration** for more details on properly setting up NTP. If NTP is properly running and configured, this error can be safely ignored.
  - Maximum locked memory check – This task checks if `memlock` is set within the `/etc/security/limits.conf` file and is only a warning. Setting `memlock` allows the `oracle` user to lock a certain amount of memory from physical RAM that isn't swapped out. The value is expressed in kilobytes and is important from the Oracle perspective because it provides the `oracle` user permission to use huge pages. This warning can be safely ignored at the moment of installation as it is configured later during the setup of huge pages. More information on huge pages can be found in **Section 4.1.5 Enabling HugePages**
  - `/dev/shm` mounted as temporary file system – Mounting of `/dev/shm` is required when using Automatic Memory Management (AMM). Due to this reference environment, taking advantage of HugePages, `/dev/shm` is not used. This warning can be safely ignored.
13. Within the **Prerequisite Checks** window, select *yes* to the Oracle Database 12c Release 1 Installer popup window.

14. Within the **Summary** window, review all the information provided, and select **Install** to start the installation.

15. Once the installation completes, execute the scripts within the **Execute Configuration scripts** window. As the root user on each node, run the following:

```bash
# /u01/app/oracle/product/12.1.0/dbhome_1/root.sh
Performing root user operation.

The following environment variables are set as:
  ORACLE_OWNER= oracle
  ORACLE_HOME= /u01/app/oracle/product/12.1.0/dbhome_1

Enter the full path name of the local bin directory: [/usr/local/bin]:
The contents of "dbhome" have not changed. No need to overwrite.
The contents of "oraenv" have not changed. No need to overwrite.
The contents of "coraenv" have not changed. No need to overwrite.

Entries will be added to the /etc/oratab file as needed by Database Configuration Assistant when a database is created
Finished running generic part of root script.
Now product-specific root actions will be performed.
```

**NOTE:** In the example above, `/u01/app/oracle/product/12.1.0/dbhome_1` is the Oracle home directory.

16. Click **OK** within the **Execute Configuration scripts** window.

17. Within the **Finish** window, verify the installation was successful and click **Close**.
4.1.3 Creating ASM Diskgroups via the ASM Configuration Assistant (ASMCA)

Prior to the creation of an Oracle RAC database, create the Fast Recovery Area (FRA), Redo Logs, and Oracle ASM diskgroups via Oracle's ASM Configuration Assistant (ASMCA).

The following steps should be done on node one of the Oracle RAC Database cluster environment.

1. **ssh** with the `-Y` option as the grid user is required prior to running **asmca**.

2. As the grid user, start **asmca** via the following command:

```
# /u01/app/12.1.0/grid/bin/asmca
```

**NOTE:** In the example above, `/u01/app/12.1.0/grid` is the grid home directory.

3. Via the **asmca** application, select the **Disk Groups** tab and click **Create**.

---

**Figure 4.1.3.1: ASMCA Disk Groups Tab**
4. Within the **Create Disk Group** window, provide the following:

- A name for the disk group, i.e. **FRADG**
- Redundancy level for the disk group, i.e. **External Redundancy**
- Selection of the disks to be added to the disk group, i.e. `/dev/mapper/fra1`
- Click the **Show Advanced Options** radio button and provide the appropriate AU Size. This reference environment uses an AU Size of 4 MB for disk groups all disks groups: **DATA, FRADG, and REDODG**.

<table>
<thead>
<tr>
<th>ASMCA Disk Group</th>
<th>Diskgroup Name</th>
<th>Disk Path</th>
<th>Header Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATADG</td>
<td>If using device mapper multipath: /dev/mapper/db1p1 /dev/mapper/db2p1</td>
<td>If using device mapper multipath: Candidate</td>
</tr>
<tr>
<td></td>
<td>FRADG</td>
<td>If using device mapper multipath: /dev/mapper/fra1</td>
<td>If using device mapper multipath: Candidate</td>
</tr>
<tr>
<td></td>
<td>REDODG</td>
<td>If using device mapper multipath: /dev/mapper/redo1</td>
<td>If using device mapper multipath: Candidate</td>
</tr>
</tbody>
</table>

*Table 4.1.3.1: ASMCA Create Disk Group*
NOTE: To display the appropriate eligible disks, click on the Change Discovery Path button and enter as the 'Disk Discovery Path' one of the following as appropriate:

- For Device Mapper devices, type: /dev/mapper/*

NOTE: A 4 MB AU Size is used to decrease the amount of extents Oracle needs to manage. With less extents to manage, CPU utilization and memory consumption is reduced thus improving performance. The AU Size varies depending on the type of Oracle workload, I/O size per transaction, and overall diskgroup size. There is no “best size” for AU size, but a good starting point is a 4 MB. Please ensure to visit Oracle's documentation for more information.
Click the OK button once the steps above are complete.

5. Repeat steps 2, 3, and 4 to configure a disk group for the redo logs and Fast Recovery Area (FRA).
   **NOTE:** Separation of redo logs into a separate Oracle ASM disk group is optional, but recommended.

6. Once all disk groups are created, click the Exit button from the main ASM **Configuration Assistant** window. Click yes, when asked to confirm quitting the application.

---

### 4.1.4 Creating Pluggable Databases using Database Configuration Assistant (DBCA)

With the introduction to Oracle Database 12c, Oracle introduced the Multitenant architecture. The Multitenant architecture provides the ability to consolidate multiple databases known as pluggable databases (PDBs) into a single container database (CDB). It provides advantages which include easier management and monitoring of the physical database, fewer patches and upgrades, performance metrics consolidated into one CDB, and sizing one SGA instead of multiple SGAs. While using the Multitenant architecture is optional, this reference architecture focuses on describing the step-by-step procedure of taking advantage of it. When creating an Oracle RAC database, the recommended method is the usage of the dbca utility.

Prior to getting into the details of installing a container database (CDB) and deploying pluggable databases (PDB), an overview of the key concepts of the Multitenant Architecture is provided.

**Container** – is a collection of schemas, objects, and related structures in a multitenant container database (CDB) that appears logically to an application as a separate database. Within a CDB, each container has a unique ID and name.

A CDB consists of two types of containers: the root container and all the pluggable databases that attach to a CDB.

**Root container** – also called the root, is a collection of schemas, schema objects, and nonschema objects to which all PDBs belong. Every CDB has one and only one root container, which stores the system metadata required to manage PDBs (no user data is stored in the root container). All PDBs belong to the root. The name of the root container is `CDB$ROOT`.

**PDB** – is a user-created set of schemas, objects, and related structures that appears logically to an application as a separate database. Every PDB is owned by `SYS`, which is a common user in the CDB, regardless of which user created the CDB.

For more information on Oracle's Multitenant architecture, visit Oracle's documentation.

---

23 Purpose of a Multitenant Environment -

http://docs.oracle.com/cd/E16655_01/server.121/e17636/cdb_intro.htm#ADMIN13509

[refarch-feedback@redhat.com](mailto:refarch-feedback@redhat.com)  67  www.redhat.com
The following section describes the step-by-step procedure to create a container database (CDB) that holds two pluggable databases (PDB) thus taking advantage of Oracle's Multitenant architecture.

The following steps should be done on node one of the Oracle RAC Database cluster environment.

1. `ssh` with the `-Y` option as the `oracle` user prior to running `dbca`.
2. As the `oracle` user, run the `dbca` utility via the command:

   ```bash
   # /u01/app/oracle/product/12.1.0/dbhome_1/bin/dbca &
   ```

   **NOTE:** In the example above, `/u01/app/oracle/product/12.1.0/dbhome_1` is the Oracle home directory.

3. Within the **Database Operations** window, select **Create a Database** radio button and click **Next**.

4. Within the **Creation Mode** window, select **Advanced Mode** radio button and click **Next**.

5. Within the **Database Template** window, select ensure **Database Type** is set to **Oracle Real Application Clusters (RAC) database** and **Custom Database** radio button and click **Next**.

   **NOTE:** This reference environment sets **Configure Type** to **Admin-Managed**. If interested in Policy-Managed Configuration Type, more information can be found within Oracle’s documentation – Using Server Pools with Oracle RAC Databases.²⁴

---

²⁴ Oracle Documentation – Using Server Pools with Oracle RAC -
[https://docs.oracle.com/database/121/RILIN/srvpool.htm#RILIN1063](https://docs.oracle.com/database/121/RILIN/srvpool.htm#RILIN1063)
6. Within the **Database Identification** window, set a global database name and SID prefix, i.e. *cdb*. Check the check box that reads **Create as Container Database**. Select the number of PDBs to install and provide a PDB Name Prefix, i.e. *pdb* and click **Next**. This reference environment creates two PDBs.

![Database Identification Window](image)

**Figure 4.1.4.1: Database Identification Window**

7. Within the **Database Placement** window, move all available nodes within the Oracle RAC Database cluster to the **Selected** side via the arrow button and click **Next**.

8. Within the **Management Options** window, ensure the check box **Run Cluster Verification Utility (CV) Checks Periodically** and **Configure Enterprise Manager (EM) Database Express** is checked (default) and click **Next**.

9. Within the **Database Credentials** window, provide the administrative passwords for **SYS**, **SYSTEM**, and **PDBADMIN**, click **Next**.
10. Within the **Storage Locations** window, select the appropriate **Storage Type** and storage locations. For the purposes of this reference environment, the following selections were made:

- **Database Files Storage Type** – Automatic Storage Management (ASM)
  - Use Common Location for All Database Files selected
    - Database File Locations: +DATADG
    - Use Oracle-Managed Files (checked)

- **Recovery Related Files Storage Type** – Automatic Storage Management (ASM)
  - Storage Locations – Specify Fast Recovery Area
    - Fast Recovery Area: +FRADG
    - Fast Recovery Area Size: 190635 Megabytes (max size created of volume)
    - Enable Archiving (checked)

---

**Figure 4.1.4.2: Storage Locations Window**

---

11. Within the same Storage Locations window, select the Multiplex Redo Logs and Control Files button and within the Multiplex Redo Logs and Control Files window, add the Redo Logs diskgroup, i.e. +REDODG and click OK, then click Next in the Storage Locations window.

![Multiplex Redo Logs and Control Files](image)

<table>
<thead>
<tr>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>+REDODG</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

**Figure 4.1.4.3: Multiplex Redo Logs and Control Files Window**

12. Within the Database Options window, all the components should be greyed out. Click Next.

13. Within the Database Vault & Label Security window, this reference environment unchecks Configure Database Vault and Configure Label Security. These products provide a level of security with regards to access control decisions at the object level as done by Oracle Database Vault and access control decisions at the row level as done by Oracle Label Security. For more information on these two products, be sure to visit Oracle’s Documentation about Oracle Database Vault\(^\text{26}\) and Oracle Label Security\(^\text{27}\).

---

\(^\text{26}\) Oracle Documentation – Oracle Database Vault  
[http://docs.oracle.com/database/121/DVADM/dvintro.htm#DVADM001](http://docs.oracle.com/database/121/DVADM/dvintro.htm#DVADM001)

\(^\text{27}\) Oracle Documentation – Oracle Label Security  
[http://docs.oracle.com/database/121/OLSAG/intro.htm#OLSAG001](http://docs.oracle.com/database/121/OLSAG/intro.htm#OLSAG001)
14. Within the **Initialization Parameters** window under the **Memory** tab, select **Custom Settings** and enter the appropriate values for the **SGA** and **PGA** size. It is recommended that the Memory Management be set as **Automatic Shared Memory Management**. The values set for the reference environment with regards to SGA and PGA are the following, however, these values vary for every Oracle database environment.

- SGA – 14460 Megabytes
- PGA – 4820 Megabytes

![Figure 4.1.4.4: Initialization Parameters Window, Memory Tab](image)

**NOTE:** It is recommended to modify the values of the SGA and PGA based on the Oracle database workload requirements.

15. Within the **Sizing** tab of the **Initialization Parameters** window, appropriately set the block size and maximum number of user processes that can be simultaneously connected to the database. For the purpose of this reference environment, the defaults are used.
NOTE: The Character Sets tab and Connection Mode tab within the Initialization Parameters window are not pictorially represented in this reference architecture, however, the default values are set.

16. Within the Creation Options window, ensure the Create Database box is checked and click Next.

17. Within the Summary window, review the Create Database – Summary, and click Finish to start the database creation.

18. Within the Finish window, click Close.
4.1.5 Enabling HugePages

Transparent Huge Pages (THP) are implemented within Red Hat Enterprise Linux 7 to improve memory management by removing many of the difficulties of manually managing huge pages by dynamically allocating huge pages as needed. Red Hat Enterprise Linux 7, by default, uses transparent huge pages also known as anonymous huge pages. Unlike static huge pages, no additional configuration is needed to use them. Huge pages can boost application performance by increasing the chance a program will have quick access to a memory page. Unlike traditional huge pages, transparent huge pages can be swapped out (as smaller 4kB pages) when virtual memory clean up is required.

Unfortunately, Oracle Databases do not take advantage of transparent huge pages for interprocess communication. In fact, My Oracle Support [ID 1557478.1]28 states to disable THP due to unexpected performance issues or delays when THP is found to be enabled. To reap the benefit of huge pages for an Oracle database, it is required to allocate static huge pages and disable THP. Due to the complexity of properly configuring huge pages, it is recommended to copy the bash shell script found within Appendix E Huge Pages Script and run the script once the database is up and running. The reasoning behind allocating huge pages once the database is up and running is to provide a proper number of pages to handle the running shared memory segments. The steps are as follows:

On node one within the Oracle RAC Database environment,

1. Copy the bash script found within Appendix E Huge Pages Script and save it as huge_pages_settings.sh
2. As the root user, ensure the huge_pages_settings.sh is executable by running the following command:
   ```bash
   # chmod +x huge_pages_settings.sh
   ```
3. As the root user, execute the huge_pages_settings.sh script as follows:
   ```bash
   # ./huge_pages_settings.sh
   ```

   Recommended setting within the kernel boot command line: hugepages = <value>
   Recommended setting within /etc/security/limits.d/99-grid-oracle-limits.conf:
   oracle soft memlock <value>
   Recommended setting within /etc/security/limits.d/99-grid-oracle-limits.conf:
   oracle hard memlock <value>

28 ALERT: Disable Transparent HugePages on SLES11,RHEL6,OEL6 and UEK2 Kernels (DOC ID: 1557478.1)
On each node within the Oracle RAC Database cluster,

4. Add the number of hugepages provided by the `huge_pages_settings.sh` script that was run on Oracle RAC Database node one to the kernel boot command line within the `/etc/default/grub` on each Oracle RAC Database node as follows:

   ```
   # vi /etc/default/grub
   GRUB_TIMEOUT=5
   GRUB_DISTRIBUTOR="\$(sed 's, release .*$,,g' /etc/system-release)"
   GRUB_DEFAULT=saved
   GRUB_DISABLE_SUBMENU=true
   GRUB_TERMINAL_OUTPUT="console"
   GRUB_CMDLINE_LINUX="rd.lvm.lv=myvg/swap rd.lvm.lv=myvg/usr
                   vconsole.font=latarcyrheb-sun16 rd.lvm.lv=myvg/root crashkernel=auto
                   vconsole.keymap=us rhgb quiet transparent_hugepage=never
                   hugepages=<value-provided-by-script>"
   GRUB_DISABLE_RECOVERY="true"
   
   NOTE: Allocating the number of huge pages within the kernel boot command line is the most reliable method due to memory not yet becoming fragmented.
   
   5. For the grub changes to take effect, run the following command:

   ```
   # grub2-mkconfig -o /boot/grub2/grub.cfg
   Generating grub configuration file ... 
   Found linux image: /boot/vmlinux-3.10.0-123.el7.x86_64 
   Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img 
   Found linux image: /boot/vmlinux-0-rescue-41c535c189b842eea5a8c20cbd9bff26 
   Found initrd image: /boot/initramfs-0-rescue-41c535c189b842eea5a8c20cbd9bff26.img 
   done
   
   6. Oracle requires setting the soft and hard limits to `memlock`. Setting `memlock` allows the oracle user to lock a certain amount of memory from physical RAM that isn't swapped out. The value is expressed in kilobytes and is important from the Oracle perspective because it provides the oracle user permission to use huge pages. This value should be slightly larger than the largest SGA size of any of the Oracle Database instances installed in an Oracle environment. To set `memlock`, add within `/etc/security/limits.d/99-grid-oracle-limits.conf` the following:

   ```
   oracle soft memlock <value-provided-by-script>
   oracle hard memlock <value-provided-by-script>
   
   NOTE: This step satisfies the warning message provided during the Oracle Database Software Installation with regards to the Maximum locked memory check.
   
   7. Reboot the system to ensure the huge pages setting takes effect properly.
8. Verify the value provided by the `huge_pages_settings.sh` matches the total number of huge pages available on the system with the following command:

```
# cat /proc/meminfo | grep -i hugepages_total
HugePages_Total: <value-provided-by-script>
```

9. Verify the current status of the `transparent` huge pages is set to `NEVER` via the following command:

```
# cat /sys/kernel/mm/transparent_hugepage/enabled
always madvise [never]
```

**NOTE:** Starting with Oracle Database version 11.2.0.2, the initialization parameter “USE_LARGE_PAGES” was introduced to allocate huge pages on a per database use case. The default value for Oracle Database 12.1.0.2.0 is set to true. Section 5 Logging into the **Oracle Container Database 12c Release 1 (12.1)** shows how to set “USE_LARGE_PAGES” to the recommended value of **only** to ensure huge pages are always used upon Oracle database startup. For more information on the parameter and its value refer to My Oracle Support.

**NOTE:** Huge pages is not compatible with Automatic Memory Management (AMM).
5 Logging into the Oracle Container Database 12c Release 1 (12.1)

This section focuses on ensuring once the Oracle Database 12c Release 1 (12.1) deployment is complete, the oracle user can successfully log into the Oracle container database (CDB), and ensure the Oracle database is using the allocated huge pages. The following steps provide the details.

The following example steps are done on node one of the Oracle RAC environment unless otherwise specified.

As the oracle user,

1. Set the environment variable for ORACLE_HOME with the location of your Oracle Database 12c Release 1 (12.1) home. This reference environment sets ORACLE_HOME to /u01/app/oracle/product/12.1.0/dbhome_1

   ```bash
   # export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
   # echo $ORACLE_HOME
   /u01/app/oracle/product/12.1.0/dbhome_1
   ```

   **NOTE:** As a precaution, ensure not to include a trailing forward slash (/) when exporting the ORACLE_HOME.

2. Set the Oracle System ID (ORACLE_SID) used to identify the CDB database.

   ```bash
   # export ORACLE_SID=cdb1
   # echo $ORACLE_SID
cdb1
   ```

3. Invoke the sqlplus binary to log into the Oracle instance as a sysdba.

   ```bash
   # $ORACLE_HOME/bin/sqlplus / as sysdba;
   ```

   SQL*Plus: Release 12.1.0.2.0 Production on Tue Nov 18 10:43:55 2014
   Copyright (c) 1982, 2014, Oracle. All rights reserved.
   Connected to:
   Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
   With the Partitioning, Automatic Storage Management, OLAP, Advanced Analytics and Real Application Testing options

4. Verify the current value of the Oracle parameter `use_large_pages`

   ```sql
   SQL> show parameter use_large_pages;
   ```

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>use_large_pages</td>
<td>string</td>
<td>TRUE</td>
</tr>
</tbody>
</table>
**CAUTION:** Ensure there is enough physical RAM on the nodes within the Oracle RAC Database cluster to place the entire SGA in large pages, otherwise the Oracle RAC database instance(s) will not start. If there is not enough physical RAM on the Oracle RAC Database cluster to place the entire SGA into large pages, leave the default setting of true and ignore the remaining steps in this section.

5. Set the value of the Oracle parameter `use_large_pages` to the value of **only**.

```sql
SQL> alter system set use_large_pages=only scope=spfile sid='*';
System altered.
```

6. On all nodes in the Oracle RAC Database cluster, shutdown the Oracle database instance.

```sql
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
```

7. On all nodes in the Oracle RAC Database cluster, restart the Oracle database instance.

```sql
SQL> startup;
ORACLE instance started.
```

```
Total System Global Area 2.0176E+10 bytes
Fixed Size 3478432 bytes
Variable Size 2952793184 bytes
Database Buffers 1.7180E+10 bytes
Redo Buffers 40247296 bytes
Database mounted.
Database opened.
```

**NOTE:** Ensure to set the appropriate ORACLE_SID and ORACLE_HOME within each node.

8. Verify the current value of the Oracle parameter `use_large_pages` is now set to **only**.

```sql
SQL> show parameter use_large_pages;
```

```
NAME               TYPE     VALUE
------------------ --------- --------
use_large_pages    string    ONLY
```
9. Open the container database's alert log, named alert_<name-of-cdb>.log, located under the $ORACLE_BASE/diag/rdbms/<name-of-cdb>/<name-of-cdb-instance#>/trace/ using a text editor, such as vi, and search for the following snippet to ensure that the System Global Area (SGA) is 100% in large pages.

$ORACLE_BASE/diag/rdbms/<name-of-cdb>/<name-of-cdb-instance#>/trace/alert_<name-of-cdb>.log

**********************************************************************
Tue Nov 18 10:45:41 2014
Dump of system resources acquired for SHARED GLOBAL AREA (SGA)

Tue Nov 18 10:45:41 2014
Per process system memlock (soft) limit = 19G
Tue Nov 18 10:45:41 2014
Expected per process system memlock (soft) limit to lock
SHARED GLOBAL AREA (SGA) into memory: 19G
Tue Nov 18 10:45:41 2014
Available system pagesizes:
  4K, 2048K
Tue Nov 18 10:45:41 2014
Supported system pagesize(s):
Tue Nov 18 10:45:41 2014

<table>
<thead>
<tr>
<th>PAGESIZE</th>
<th>AVAILABLE_PAGES</th>
<th>EXPECTED_PAGES</th>
<th>ALLOCATED_PAGES</th>
<th>ERROR(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2048K</td>
<td>9668</td>
<td>9665</td>
<td>9665</td>
<td>NONE</td>
</tr>
</tbody>
</table>

Reason for not supporting certain system pagesizes:

Tue Nov 18 10:45:41 2014
4K - Large pagesizes only

**********************************************************************

NOTE: This reference environment’s SGA size is set to 19 GB, however, this value will vary depending on the value provided when creating an Oracle RAC database using dbca.
6 Post Installation Cleanup Tasks

This section describes the removal of certain tasks that were set in order to satisfy the prerequisites of the Oracle Universal Installer (OUI). While these prerequisites are required, the OUI does not handle them appropriately for a Red Hat Enterprise Linux 7 distribution.

6.1.1 Removal of Oracle Kernel Parameters from /etc/sysctl.conf

Changing kernel parameters within the /etc/sysctl.conf file directly can leave a system susceptible to having kernel parameter entries being overwritten, replaced, or removed all together if there were to be an update to the initscripts RPM. If not using the oracle tuned profile as described in this reference architecture, please ensure to have all the oracle related kernel parameters set within 98-oracle.conf file inside the /etc/sysctl.d directory. The reason they are set initially in the /etc/sysctl.conf file is due to Oracle’s Universal Installer checking the file /etc/sysctl.conf for kernel parameter settings instead of checking the kernel running values.

NOTE: Files in /etc/sysctl.d are read in order, i.e. 1-example.conf is read before 2-example.conf, hence it is possible for two files to have conflicting settings. The last file read, takes precedence.

All nodes within the Oracle RAC cluster, as root user,

1. Edit the /etc/sysctl.conf using an editor such as vi and remove all Oracle parameter related entries. The /etc/sysctl.conf file should be blank and all settings should be set in a .conf file within the /etc/sysctl.d or set by a tuned profile such as the oracle profile found in /usr/lib/tuned from Section 3.4.4 Optimizing Database Storage using Automatic System Tuning. This reference environment, uses the custom tuned oracle profile for kernel parameter settings.

2. Save the /etc/sysctl.conf file and run the following command to set the changes:

```bash
# sysctl -p /etc/sysctl.conf
```

3. Verify the Oracle tuned profile is still set by:

```bash
# tuned-adm active
Current active profile: oracle
```

NOTE: If not using the custom tuned profile oracle, please ensure to have all the oracle kernel parameters set within the /etc/sysctl.d/98-oracle.conf file

4. Verify the Oracle kernel parameters are still set by run the following command:

```bash
# sysctl -a | grep <oracle kernel parameter>
```

Example of checking kernel.shmmax:

```bash
# sysctl -a | grep kernel.shmmax
kernel.shmmax = 4398046511104
```
6.1.2 Removal of `firewalld` Trusted Source Address

During the installation of the Oracle Grid Infrastructure 12c Release 1, the installation will fail during the **Network Interface Usage** step if any type of firewall is running. Instead of completely disabling the firewall, the source address of our bonded interface, bonded0, is added to the trusted zone within `firewalld`. However, this should be immediately removed upon completion of the installation of Oracle as bonded0 is a public interface and should not reside in the trusted zone. For clarification, interfaces ens10f2 and ens10f3 are private interfaces that reside on a private network not accessible to the outside world. Due to this, it is acceptable for these interfaces to be part of the trusted zone.

On all nodes within the Oracle RAC cluster as the root user, remove the source public address of the other nodes in the cluster. For example, this reference environment consists of a two node Oracle RAC cluster, with node one's public IP set to 10.19.142.51 and node two's public IP set to 10.19.142.52.

Within node one of the Oracle RAC cluster, run the following command:

```bash
# firewall-cmd --permanent --zone=trusted --remove-source=10.19.142.52/21
success
# systemctl restart firewalld.service
```

Within node two of the Oracle RAC cluster, run the following command:

```bash
# firewall-cmd --permanent --zone=trusted --remove-source=10.19.142.51/21
success
# systemctl restart firewalld.service
```

With the removal of the public source addresses, Oracle RAC Database can now run properly with a firewall protecting the public interface.
Common Tasks when Managing Container Database (CDB) and Pluggable Databases (PDB)

This section describes tasks that are commonly used when dealing with a CDB and PDBs. The tasks covered within this section are as follows:

- Connect to a CDB
- Connect to a PDB
- Managing a CDB
- Managing a PDB
- Location of Data files in a CDB & PDB

7.1 Connect to a CDB

Example on how to connect to node one of an Oracle RAC Database cluster.

As the oracle user,

1. Set the environment variable for ORACLE_HOME with the location of your Oracle Database 12c Release 1 (12.1) home. This reference environment sets ORACLE_HOME to /u01/app/oracle/product/12.1.0/dbhome_1

```bash
# export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
# echo $ORACLE_HOME
/u01/app/oracle/product/12.1.0/dbhome_1
```

**NOTE:** As a precaution, ensure not to include a trailing forward slash (/) when exporting the ORACLE_HOME.

2. Set the Oracle System ID (ORACLE_SID) used to identify the CDB database.

```bash
# export ORACLE_SID=cdb1
# echo $ORACLE_SID
cdb1
```
3. Invoke the `sqlplus` binary to log into the Oracle instance as a `sysdba`.

```
# $ORACLE_HOME/bin/sqlplus / as sysdba;
```

SQL*Plus: Release 12.1.0.2.0 Production on Tue Nov 18 10:55:19 2014

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 – 64bit
Production
With the Partitioning, Automatic Storage Management, OLAP, Advanced
Analytics
and Real Application Testing options

4. Once connected, verify that the instance is connected to the root container, `CDB$ROOT` with a `CON_ID` is 1.

```
SQL> show con_name

CON_NAME
----------
CDB$ROOT

SQL> show con_id

CON_ID
-------
1
```

**NOTE:** The `CDB$ROOT` connection ID is always set to one.

5. To list all the available services and PDBs within the CDB:

```
SQL> select name, con_id from v$active_services;

NAME                | CON_ID |
---------------------|--------|
pdb2                | 4      |
pdb1                | 3      |
cdbXDB              | 1      |
cdb                 | 1      |
SYS$BACKGROUND      | 1      |
SYS$USERS           | 1      |

6 rows selected.
```
7.2 Connect to a PDB

The syntax to connect to a PDB varies depending on whether or not there is an entry within the tnsnames.ora file for the PDB.

Without an entry to the tnsnames.ora file, the syntax to connect to a PDB named `pdb` with an instance name of `pdb1` from node one of the Oracle RAC cluster is as follows:

```
$ORACLE_HOME/bin/sqlplus sys/<password>@db-oracle-node1:1521/pdb1 as sysdba;
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Tue Nov 18 10:57:15 2014
Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Advanced Analytics
and Real Application Testing options
```

**NOTE:** The value 1521, represents the Oracle Listener port.

With an entry to the tnsnames.ora file, the syntax to connect to a PDB instance labeled `pdb1` is as follows:

```
$ORACLE_HOME/bin/sqlplus sys/<password>@PDB1 as sysdba;
```

A snippet of the entry found within the tnsnames.ora file is displayed below:

```
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/tnsnames.ora

PDB1 =
    (DESCRIPTION =
       (ADDRESS = (PROTOCOL = TCP)(HOST = db-oracle-
        scan.cloud.lab.eng.bos.redhat.com)(PORT = 1521))
       (CONNECT_DATA =
          (SERVER = DEDICATED)
          (SERVICE_NAME = pdb1)
       )
    )
```

www.redhat.com  84  refarch-feedback@redhat.com
7.3 Managing a CDB

The process of starting and shutting down a CDB database instance is similar to the steps done in previous Oracle database versions for traditional databases. The key difference is to verify that the connection is to the root container prior to shutting down or starting up the Oracle database.

1. Connect to the CDB database as a SYSDBA using sqlplus. The steps are the same as shown in Section 7.1 Connect to a CDB steps one through three.

2. Once connected, verify the instance is the root container CDB$ROOT via the command:

   SQL> show con_name;

   CON_NAME
   ------------------------------
   CDB$ROOT

3. To shutdown the Oracle CDB database use the following command:

   SQL> shutdown immediate;
   Database closed.
   Database dismounted
   ORACLE instance shut down.

4. To start the Oracle CDB database use the following command:

   SQL> startup;

The startup command starts the instance, mounts the control files, and then opens the root container.
7.4 Managing a PDB

This section focuses on verifying the `OPEN_MODE` of a PDB, how to open and close a specific PDB, and how to open and close all PDBs within a CDB.

1. To verify the open_mode status of all the PDBs, while logged in as a `SYSDBA` in the CDB, use the following command:

   SQL> `select name, open_mode from v$pdbs;`

<table>
<thead>
<tr>
<th>NAME</th>
<th>OPEN_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDB$SEED</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>PDB1</td>
<td>MOUNTED</td>
</tr>
<tr>
<td>PDB2</td>
<td>MOUNTED</td>
</tr>
</tbody>
</table>

2. When a PDB is closed, the `OPEN_MODE` is set to `MOUNTED`. To open a PDB and verify the new `OPEN_MODE` of `READ WRITE`, run the following SQL syntax while logged in as a `SYSDBA` in the CDB:

   SQL> `alter pluggable database pdb1 open;`
   Pluggable database altered.

   SQL> `select name, open_mode from v$pdbs;`

<table>
<thead>
<tr>
<th>NAME</th>
<th>OPEN_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDB$SEED</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>PDB1</td>
<td>READ WRITE</td>
</tr>
<tr>
<td>PDB2</td>
<td>MOUNTED</td>
</tr>
</tbody>
</table>

3. To open all the PDBs connected to a CDB and verify the new `OPEN_MODE` of `READ WRITE`, run the following SQL syntax while logged in as a `SYSDBA` in the CDB:

   SQL> `alter pluggable database all open;`
   Pluggable database altered.

   SQL> `select name, open_mode from v$pdbs;`

<table>
<thead>
<tr>
<th>NAME</th>
<th>OPEN_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDB$SEED</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>PDB1</td>
<td>READ WRITE</td>
</tr>
<tr>
<td>PDB2</td>
<td>READ WRITE</td>
</tr>
</tbody>
</table>
4. To drop a particular PDB i.e. *pdb2*, and its data files, execute the following SQL syntax while logged in as a **SYSDBA** in the CDB:

```sql
SQL > alter pluggable database pdb2 close immediate;
Pluggable database altered.

SQL > drop pluggable database pdb2 including datafiles;
Pluggable database dropped.
```

5. To verify if the pluggable database with the name *pdb2* has been dropped:

```sql
SQL> select name, open_mode from v$pdbs;
NAME          OPEN_MODE
-------------- ----------
PDB$SEED       READ ONLY
PDB1           READ WRITE
```

### 7.5 Location of Data files of PDBs & CDB

The following section shows how to identify tablespace names, data files associated with the CDB and PDBs, including their temporary files.

1. Connect to the CDB database as a **SYSDBA** using **sqlplus**. The steps are the same as shown in **Section 7.1 Connect to a CDB** steps one through three.

2. To identify the tablespaces associated with the CDB or any of the PDBs installed, use the following syntax where the `con_id` varies upon the database chosen. The example below uses the `con_id` of 1 to show the CDB tablespaces.

```sql
SQL > select tablespace_name, con_id from cdb_tablespaces where con_id = 1;
TABLESPACE_NAME | CON_ID
-----------------|-------
SYSTEM           | 1
SYSAUX           | 1
UNDOTBS1         | 1
TEMP             | 1
USERS            | 1
```

3. To locate the data files from the CDB or PDBs installed, use the following syntax where the `con_id` varies upon the database chosen. The example below uses the `con_id` of 1 to show the CDB data file locations.

```sql
SQL> select file_name, con_id from cdb_data_files where con_id=1;
FILE_NAME                  CON_ID
---------------------------|------
+DATADG/CDB/DATAFILE/system.270.836232071 | 1
+DATADG/CDB/DATAFILE/sysaux.273.836232077  | 1
+DATADG/CDB/DATAFILE/undotbs1.262.836232081 | 1
```
4. To locate the temporary files from the CDB or PDBs installed, use the following syntax where the con_id varies upon the database chosen. The example below uses the con_id of 1 to show the CDB data file locations.

```sql
SQL> select file_name, con_id from cdb_temp_files where con_id =1 ;
```

<table>
<thead>
<tr>
<th>FILE_NAME</th>
<th>CON_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>+DATADG/CDB/TEMPFILE/temp.278.836232081</td>
<td>1</td>
</tr>
</tbody>
</table>

www.redhat.com  88  refarch-feedback@redhat.com
8 Conclusion

Red Hat Enterprise Linux 7 provides an excellent foundation for database deployments with demonstrated stability, scalability, and performance. With the support for Oracle RAC Database 12c Release 1 (12.1) on Red Hat Enterprise Linux 7, customers can increasingly look to deploy Oracle RAC Databases in advanced configurations.

The steps and procedures described in this reference architecture should provide system, database, and storage administrators the blueprint required to create a robust and performing solution based on Oracle RAC Databases. Administrators can reference this document to simplify and optimize the deployment process and employ the latest best practices for configuring Red Hat technologies while implementing the following tasks:

- Deploying Oracle Grid Infrastructure 12c Release 1 (12.1.0.2.0)
- Deploying Oracle Database Software 12c Release 1 (12.1.0.2.0)
- Deploying an Oracle RAC Database 12c Release 1 (12.1.0.2.0) with shared SAN disks
- Using Oracle ASM disks with udev rules
- Securing the Oracle Database 12c environment with SELinux
- Securing the public network using firewalld

For any questions or concerns, please email refarch-feedback@redhat.com and ensure to visit the Red Hat Reference Architecture page at http://red.ht/1Nd64ha to find out about all of our Red Hat solution offerings.
Appendix A: Revision History

Revision 1.1  Friday March 05, 2015  Roger Lopez
- Within the Conclusion, fixed hyperlink of Reference Architectures to point to http://red.ht/1Nd64ha
- Fixed some missing dependencies within the .tar.gz file that were already correctly reflexed in the scripts within the Appendix sections. Details within the CHANGELOG of the .tar.gz

Revision 1.0  Friday February 20, 2015  Roger Lopez
Initial Release
Appendix B: Contributors

1. Tom Tracy, content reviewer
2. David Wilson, content reviewer
3. John Boero, content reviewer
4. Yan Fisher, content reviewer
5. Scott McBrien, content reviewer
Appendix C: DM Multipath Configuration File

# This is a basic configuration file with some examples, for device mapper multipath.
# For a complete list of the default configuration values, see
# /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.defaults
# For a list of configuration options with descriptions, see
# /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.annotated
#
# REMEMBER: After updating multipath.conf, you must run
# service multipathd reload
#
# for the changes to take effect in multipathd

### By default, devices with vendor = "IBM" and product = "S/390.*" are
### blacklisted. To enable multilpathing on these devies, uncomment the
### following lines.
#blacklist_exceptions {
#       device {
#               vendor  "IBM"
#               product "S/390.*"
#       }
#}

### Here is an example of how to configure some standard options.
###
###
defaults {
    udev_dir                /dev
    polling_interval        10
    path_selector           "round-robin 0"
    path_grouping_policy    multibus
    getuid_callout          "/lib/udev/scsi_id --whitelisted
--device=/dev/%n"
    prio                    alua
    path_checker            readsector0
    rr_min_io               100
    max_fds                 8192
    rr_weight               priorities
    failback                immediate
    no_path_retry           fail
    user_friendly_names     yes
}

### The wwid line in the following blacklist section is shown as an example
### of how to blacklist devices by wwid. The 2 devnode lines are the
### compiled in default blacklist. If you want to blacklist entire types
### of devices, such as all scsi devices, you should use a devnode line.
### However, if you want to blacklist specific devices, you should use
### a wwid line. Since there is no guarantee that a specific device will
## not change names on reboot (from /dev/sda to /dev/sdb for example)
## devnode lines are not recommended for blacklisting specific devices.

```bash
blacklist {
    wwid 3600508b1001030353434363646301200
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z]"
}
multipaths {
multipath {
    wwid 3600c0ff000d7e7a899d8515101000000
    alias db1
}
multipath {
    wwid 3600c0ff000dabfe5a7d8515101000000
    alias db2
}
multipath {
    wwid 3600c0ff000d7e7a8dbd8515101000000
    alias fra
}
multipath {
    wwid 3600c0ff000dabfe5f4d8515101000000
    alias redo
}
}
devices {
#       device {
#               vendor                  "COMPAQ  
#               product                 "HSV110 (C)COMPAQ"
#               path_grouping_policy    multibus
#               getuid_callout          "/lib/udev/scsi_id --whitelisted
--device=/dev/%n"
#               path_checker            readsector0
#               path_selector          "round-robin 0"
#               hardware_handler        "0"
#               failback                15
#               rr_weight               priorities
#               no_path_retry           queue
#       }
#       device {
#               vendor                  "COMPAQ  
#               product                 "MSA1000         
#               path_grouping_policy    multibus
#       }
#}
```
Appendix D: Firewalld Configuration File

/etc/firewalld/zones/public.xml

<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Public</short>
  <description>For use in public areas. You do not trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.</description>
  <service name="dhcpv6-client"/>
  <service name="ssh"/>
  <rule family="ipv4">
    <source address="10.19.142.54"/>
    <port protocol="tcp" port="1521"/>
    <accept/>
  </rule>
  <rule family="ipv4">
    <source address="10.19.142.54"/>
    <port protocol="tcp" port="5500"/>
    <accept/>
  </rule>
</zone>

/etc/firewalld/zones/trusted.xml

<?xml version="1.0" encoding="utf-8"?>
<zone target="ACCEPT">
  <short>Trusted</short>
  <description>All network connections are accepted.</description>
  <interface name="<enter-private-interface-name2>"/>
  <interface name="<enter-private-interface-name2>"/>
</zone>
Appendix E: Huge Pages Script

The following huge pages script is from Tuning Red Hat Enterprise Linux For Oracle & Oracle RAC by Scott Crot, Sr. Consultant, Red Hat and modified to include the values Oracle’s soft memlock, hard memlock, and work with kernel 3.10

#!/bin/bash
KERN=`uname -r | awk -F. '{ printf("%d.%d\n",$1,$2); }'`
# Find out the HugePage size
HPG_SZ=`grep Hugepagesize /proc/meminfo | awk '{print $2}'`
# Start from 1 pages to be on the safe side and guarantee 1 free HugePage
NUM_PG=1
# Cumulative number of pages required to handle the running shared memory segments
for SEG_BYTES in `ipcs -m | awk '{print $5}' | grep "[0-9][0-9]*"`
do
    MIN_PG=`echo "$SEG_BYTES/($HPG_SZ*1024)" | bc -q`
    if [ $MIN_PG -gt 0 ]; then
        NUM_PG=`echo "$NUM_PG+$MIN_PG+1" | bc -q`
    fi
done
# Finish with results
case $KERN in
    '2.4') HUGETLB_POOL=`echo "$NUM_PG*$HPG_SZ/1024" | bc -q`;
    echo "Recommended setting: vm.hugetlb_pool = $HUGETLB_POOL"
    ;;
    '2.6') MEM_LOCK=`echo "$NUM_PG*$HPG_SZ" | bc -q`;
    echo "Recommended setting within the kernel boot command line: hugepages = $NUM_PG"
    echo "Recommended setting within /etc/security/limits.d/99-grid-oracle-limits.conf: oracle soft memlock $MEM_LOCK"
    echo "Recommended setting within /etc/security/limits.d/99-grid-oracle-limits.conf: oracle hard memlock $MEM_LOCK"
    ;;
    '3.10') MEM_LOCK=`echo "$NUM_PG*$HPG_SZ" | bc -q`;
    echo "Recommended setting within the kernel boot command line: hugepages = $NUM_PG"
    echo "Recommended setting within /etc/security/limits.d/99-grid-oracle-limits.conf: oracle soft memlock $MEM_LOCK"
    echo "Recommended setting within /etc/security/limits.d/99-grid-oracle-limits.conf: oracle hard memlock $MEM_LOCK"
    ;;
    *) echo "Unrecognized kernel version $KERN. Exiting." ;;
esac


refarch-feedback@redhat.com 95 www.redhat.com
Appendix F: Oracle Database Package Requirements Text File

binutils
compat-libcap1
compat-libstdc++-33
gcc
gcc-c++
glibc
glibc-devel
ksh
libgcc
libstdc++
libstdc++-devel
libaio
libaio-devel
libXext
test
libXtst
libX11
libXau
libxcb
libXi
make
sysstat
libXmu
libXt
libXv
libXxf86dgak
libXxf86misc
libXxf86vm
xorg-x11-utils
xorg-x11-xauth
nfs-utils
### Appendix G: Kernel Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>vm.swappiness</td>
<td>1</td>
</tr>
<tr>
<td>vm.dirty_background_ratio</td>
<td>3</td>
</tr>
<tr>
<td>vm.dirty_ratio</td>
<td>80</td>
</tr>
<tr>
<td>vm.dirty_expire_centisecs</td>
<td>500</td>
</tr>
<tr>
<td>vm.dirty_writeback_centisecs</td>
<td>100</td>
</tr>
<tr>
<td>kernel.shmmx</td>
<td>4390046511104</td>
</tr>
<tr>
<td>kernel.shmall</td>
<td>1073741824</td>
</tr>
<tr>
<td>kernel.shmmni</td>
<td>4096</td>
</tr>
<tr>
<td>kernel.sem</td>
<td>250 32000 100 128</td>
</tr>
<tr>
<td>fs.file-max</td>
<td>6815744</td>
</tr>
<tr>
<td>fs.aio-max-nr</td>
<td>1048576</td>
</tr>
<tr>
<td>net.ipv4.ip_local_port_range</td>
<td>9000 65500</td>
</tr>
<tr>
<td>net.core.rmem_default</td>
<td>262144</td>
</tr>
<tr>
<td>net.core.rmem_max</td>
<td>4194304</td>
</tr>
<tr>
<td>net.core.wmem_default</td>
<td>262144</td>
</tr>
<tr>
<td>net.core.wmem_max</td>
<td>1048576</td>
</tr>
<tr>
<td>net.ipv4.conf.&lt;priv-interface-name1&gt;.rp_filter</td>
<td>2</td>
</tr>
<tr>
<td>net.ipv4.conf.&lt;priv-interface-name2&gt;.rp_filter</td>
<td>2</td>
</tr>
<tr>
<td>kernel.panic_on_oops</td>
<td>1</td>
</tr>
</tbody>
</table>
Appendix H: 98-oracle.conf

# 98-oracle.conf resides in the /etc/sysctl.d directory
# Replace <priv-interface-name{1-2}> with the appropriate Ethernet device name
net.ipv4.conf.<priv-interface-name1>.rp_filter = 2
net.ipv4.conf.<priv-interface-name2>.rp_filter = 2

# If not using the custom oracle tuned profile
# please uncomment the following kernel parameters.
# vm.swappiness = 1
# vm.dirty_background_ratio = 3
# vm.dirty_ratio = 80
# vm.dirty_expire_centisecs = 500
# vm.dirty_writeback_centisecs = 100
# kernel.shmmax = 4398046511104
# kernel.shmall = 1073741824
# kernel.shmmni = 4096
# kernel.sem = 250 32000 100 128
# fs.file-max = 6815744
# fs.aio-max-nr = 1048576
# net.ipv4.ip_local_port_range = 9000 65500
# net.core.rmem_default = 262144
# net.core.rmem_max = 4194304
# net.core.wmem_default = 262144
# net.core.wmem_max = 1048576
# kernel.panic_on_oops = 1
Appendix I: Limits Configuration File (99-grid-oracle-limits.conf)

oracle soft nproc 16384 #Ora bug 15971421
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft stack 10240
oracle hard stack 32768
oracle soft memlock <value-provided-by-script>
oracle hard memlock <value-provided-by-script>

grid soft nproc 16384 #Ora bug 15971421
grid hard nproc 16384
grid soft nofile 1024
grid hard nofile 65536
grid soft stack 10240
grid hard stack 32768
Appendix J: 99-oracle-asmdevices.rules

KERNEL="dm-*",ENV{DM_UUID}="<enter-value-according-to-your-environment>",OWNER="grid",GROUP="asmadmin",MODE="0660"
KERNEL="dm-*",ENV{DM_UUID}="<enter-value-according-to-your-environment>",OWNER="grid",GROUP="asmadmin",MODE="0660"
KERNEL="dm-*",ENV{DM_UUID}="<enter-value-according-to-your-environment>",OWNER="grid",GROUP="asmadmin",MODE="0660"
KERNEL="dm-*",ENV{DM_UUID}="<enter-value-according-to-your-environment>",OWNER="grid",GROUP="asmadmin",MODE="0660"
Appendix K: Sample Kickstart File

# Red Hat | Oracle Solutions Kickstart Script

install
url --url=<place-distro-url-here>
lang en_US.UTF-8
keyboard us
network --onboot yes --device em1 --mtu=1500 --bootproto dhcp
rootpw <password-for-system>
# Reboot after installation
reboot
authconfig --enablemd5 --enableshadow
selinux --enforcing
timezone America/New_York
bootloader --location=mbr --driveorder=sda --append="crashkernel=auto rhgb quiet"

# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work

# Clear all parts

clearpart --all

volgroup myvg --pesize=32768 pv.008002
logvol /home --fstype=ext4 --name=home --vgname=myvg --size=8192
logvol / --fstype=ext4 --name=root --vgname=myvg --size=15360
logvol swap --name=swap --vgname=myvg --size=16400
logvol /tmp --fstype=ext4 --name=tmp --vgname=myvg --size=4096
logvol /u01 --fstype=ext4 --name=u01 --vgname=myvg --size=51200
logvol /usr --fstype=ext4 --name=usr --vgname=myvg --size=5120
logvol /var --fstype=ext4 --name=var --vgname=myvg --size=8192

part /boot --fstype=ext4 --size=256
part pv.008002 --grow --size=1000

%packages
@Base
@Core
Appendix L: Custom Oracle Tuned Profile
(script.sh & tuned.conf)

!/usr/lib/tuned/oracle/script.sh

#!/bin/bash

# Oracle ktune script

. /usr/lib/tuned/functions

start() {
    CURR_SHMALL=`sysctl -n kernel.shmall`
    CURR_SHMMAX=`sysctl -n kernel.shmmax`
    CURR_SHMMNI=`sysctl -n kernel.shmmni`
    CURR_SEM=`sysctl -n kernel.sem`
    CURR_SWAPPINESS=`sysctl -n vm.swappiness`
    CURR_DIRTY_BCKGRD_RATIO=`sysctl -n vm.dirty_background_ratio`
    CURR_DIRTY_RATIO=`sysctl -n vm.dirty_ratio`
    CURR_DIRTY_EXPIRE_CENTISECS=`sysctl -n vm.dirty_expire_centisecs`
    CURR_FILE_MAX=`sysctl -n fs.file-max`
    CURR_AIO_MAX_NR=`sysctl -n fs.aio-max-nr`
    CURR_LCL_PRT_RNG=`sysctl -n net.ipv4.ip_local_port_range`
    CURR_RMEM=`sysctl -n net.core.rmem_default`
    CURR_RMEM_MAX=`sysctl -n net.core.rmem_max`
    CURR_WMEM=`sysctl -n net.core.wmem_default`
    CURR_WMEM_MAX=`sysctl -n net.core.wmem_max`
    CURR_PANIC_ON_OOPS=`sysctl -n kernel.panic_on_oops`

    save_value kernel.shmall "$CURR_SHMALL"
    save_value kernel.shmmax "$CURR_SHMMAX"
    save_value kernel.shmmni "$CURR_SHMMNI"
    save_value vm.swappiness "$CURR_SWAPPINESS"
    save_value vm.dirty_background_ratio "$CURR_DIRTY_BCKGRD_RATIO"
    save_value vm.dirty_ratio "$CURR_DIRTY_RATIO"
    save_value vm.dirty_expire_centisecs "$CURR_DIRTY_EXPIRE_CENTISECS"
    save_value fs.file-max "$CURR_FILE_MAX"
    save_value fs.aio-max-nr "$CURR_AIO_MAX_NR"
    save_value net.ipv4.ip_local_port_range "$CURR_LCL_PRT_RNG"
    save_value net.core.rmem_default "$CURR_RMEM"
    save_value net.core.rmem_max "$CURR_RMEM_MAX"
    save_value net.core.wmem_default "$CURR_WMEM"
    save_value net.core.wmem_max "$CURR_WMEM_MAX"
    save_value kernel.panic_on_oops "$CURR_PANIC_ON_OOPS"

    return 0
}

stop() {
    SHMALL=`restore_value kernel.shmall`
}
SHMMAX=`restore_value kernel.shmmax`
SHMMNI=`restore_value kernel.shmmni`
SWAPPINESS=`restore_value vm.swappiness`
DIRTY_BCKGRD_RATIO=`restore_value vm.dirty_background_ratio`
DIRTY_EXPIRE_CENTISECS=`restore_value vm.dirty_expire_centisecs`
FILE_MAX=`restore_value fs.file-max`
LCL_PRT_RNG=`restore_value net.ipv4.ip_local_port_range`
RMEM_DEFAULT=`restore_value net.core.rmem_default`
RMEM_MAX=`restore_value net.core.rmem_max`
WMEM_DEFAULT=`restore_value net.core.wmem_default`
WMEM_MAX=`restore_value net.core.wmem_max`
PANIC_ON_OOPS=`restore_value kernel.panic_on_oops`

[ "$SHMALL" ] && sysctl -w kernel.shmall="$SHMALL"
[ "$SHMMAX" ] && sysctl -w kernel.shmmax="$SHMMAX"
[ "$SHMMNI" ] && sysctl -w kernel.shmmni="$SHMMNI"
[ "$SWAPPINESS" ] && sysctl -w vm.swappiness="$SWAPPINESS"
[ "$DIRTY_BCKGRD_RATIO" ] && sysctl -w vm.dirty_background_ratio="$DIRTY_BCKGRD_RATIO"
[ "$DIRTY_EXPIRE_CENTISECS" ] && sysctl -w vm.dirty_expire_centisecs="$DIRTY_EXPIRE_CENTISECS"
[ "$FILE_MAX" ] && sysctl -w fs.file-max="$FILE_MAX"
[ "$LCL_PRT_RNG" ] && sysctl -w net.ipv4.ip_local_port_range="$LCL_PRT_RNG"
[ "$RMEM_DEFAULT" ] && sysctl -w net.core.rmem_default="$RMEM_DEFAULT"
[ "$RMEM_MAX" ] && sysctl -w net.core.rmem_max="$RMEM_MAX"
[ "$WMEM_DEFAULT" ] && sysctl -w net.core.wmem_default="$WMEM_DEFAULT"
[ "$WMEM_MAX" ] && sysctl -w net.core.wmem_max="$WMEM_MAX"
[ "$PANIC_ON_OOPS" ] && sysctl -w kernel.panic_on_oops="$PANIC_ON_OOPS"

return 0
}

process $@
/usr/lib/tuned/oracle/tuned.conf
#
# tuned configuration
#
[main]
include=throughput-performance

[sysctl]
vm.swappiness = 1
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
kernel.shmmx = 439046511104
kernel.shmall = 1073741824
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 6815744
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
kernel.panic_on_oops = 1

[vm]
transparent_hugepages=never

[script]
script=script.sh
Appendix M: Configuration Files

All configuration files can be downloaded from the Red Hat customer portal\(^{32} \). A listing of all the files and a brief description can be seen on the Table 8.1: Configuration Files. Some of the configuration files require input with the proper information pertaining to your environment.

<table>
<thead>
<tr>
<th>Files</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>req-rpm.txt</td>
<td>The required RPMs to deploy Oracle.</td>
</tr>
<tr>
<td>huge_page_settings.sh</td>
<td>Script that provides the proper hugepage values to set.</td>
</tr>
<tr>
<td>multipath.conf</td>
<td>Device Mapper Multipath configuration file.</td>
</tr>
<tr>
<td>98-oracle.conf</td>
<td>Configuration file for the kernel parameters related to rp_filter</td>
</tr>
<tr>
<td>99-grid-oracle-limits.conf</td>
<td>Configuration file to set limits for a user.</td>
</tr>
<tr>
<td>99-oracle-asmdevices.rules</td>
<td>udev configuration file for Oracle ASM disks</td>
</tr>
<tr>
<td>public.xml</td>
<td>firewalld XML configuration</td>
</tr>
<tr>
<td>trusted.xml</td>
<td>firewalld XML configuration</td>
</tr>
<tr>
<td>oracle-grid.sh</td>
<td>Shell script used to set user limits</td>
</tr>
<tr>
<td>sample-ks.cfg</td>
<td>Sample Kickstart File</td>
</tr>
<tr>
<td>oracle</td>
<td>Directory of the custom tuned profile labeled oracle</td>
</tr>
<tr>
<td>CHAGELOG</td>
<td>Listing of the latest changes made to the .tar.gz file</td>
</tr>
<tr>
<td>script.sh</td>
<td>Oracle tuned profile script</td>
</tr>
<tr>
<td>tuned.conf</td>
<td>Configuration file for oracle tuned profile</td>
</tr>
</tbody>
</table>

\( \text{Table } 8.1: \text{Configuration Files} \)

\(^{32} \)https://access.redhat.com/node/1357883/40/1
Appendix N: Troubleshooting ORA-* Errors

This section focuses on using the command line tool, Automatic Diagnostic Repository Command Interpreter (ADRCI), to troubleshoot Oracle database related errors. ADRCI was introduced in Oracle Database 11g in order to help users diagnose errors within their Oracle database environments and provide health reports if an issue should arise. The following example shows how one could troubleshoot an Oracle database instance error using the ADRCI tool.

NOTE: The following steps are intended to produce an ORA-07445 error that can be troubleshooted using the ADRCI tool. Do not attempt on a Oracle Database Production environment. The following is for demonstration purposes only and intended only to show how to troubleshoot ORA-* related errors using the ADRCI tool.

Example of running ADRCI tool on node one of an Oracle RAC cluster.

1. In order to create an ORA-07445 error, an essential Oracle process will be killed via the following commands:

   ```
   # ps -A --format pid,args | grep ora_dbrm | grep -v grep
   7811 ora_dbrm_cdb1
   # kill -SEGV 7811
   ```

2. Export the ORACLE_HOME via the command:

   ```
   # export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
   ```

3. Start the ADRCI command tool via the command:

   ```
   # $ORACLE_HOME/bin/adrci
   ```

   ADRCI: Release 12.1.0.2.0 - Production on Wed Nov 12 10:44:50 2014
   Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.
   ADR base = "/u01/app/oracle"
   adrci>

4. At the ADRCI prompt, show Oracle Home's available via the command:

   ```
   adrci> show home
   ADR Homes:
   diag/rdbms/cdb/cdb1
   diag/tnslsnr/db-oracle-node1/listener
   ```

   **NOTE:** If more than one Oracle Home is available, one must specify a particular Oracle Database Home. An example on how to set to a particular Oracle Database Home is as follows:

   ```
   adrci> set home diag/rdbms/cdb/cdb1
   ```

5. At the ADRCI prompt, run the following command to see the last 50 entries in the alert log:

   ```
   adrci> show alert -tail -f
   ```
Exception [type: SIGSEGV, unknown code] [ADDR:0xD431000076FD] [PC:0x3199AEB22A, semtimedop()+10] [exception issued by pid: 30461, uid: 54321] [flags: 0x0, count: 1]
Errors in file /u01/app/oracle/diag/rdbms/cdb/cdb(trace/cdb_dbrm_7811.trc (incident=9673):
ORA-07445: exception encountered: core dump [semtimedop()+10] [SIGSEGV] [ADDR:0xD431000076FD] [PC:0x3199AEB22A] [unknown code] {}
Incident details in:
/u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_9673/cdb_dbrm_7811_19673.trc
Use ADRCI or Support Workbench to package the incident.
See Note 411.1 at My Oracle Support for error and packaging details.

NOTE: In this particular case, we are looking for an ORA-07445 within the alert log as seen above. However, this step is just to confirm what is seen in the future ADRCI steps. To exit the alert log, execute CTRL+C.

6. Within the ADRCI, there are two key terms to be aware of, problem and incident. An incident is a particular time when a problem occurred. For example, it is possible for an Oracle process to crash at different times with the same ORA-07445. The multiple occurrences of the crash are incidents, while the problem is still the ORA-07445 error. In order to view the problem, the following ADRCI command needs to be run.

   adrci> show problem
ADR Home = /u01/app/oracle/diag/rdbms/cdb/cdb1:
***************************************************************************************
PROBLEM_ID PROBLEM_KEY LAST_INCIDENT LASTINC_TIME
-------------------------------------------------------------------------
1 ORA 7445 [semtimedop] 9673 2014-11-12 10:20:50.273000-05:00
1 row fetched

7. In order to view how many incidents, the following ADRCI command must be run. In this example, I only have one incident in which the ORA-07445 problem occurred.

   adrci> show incident
ADR Home = /u01/app/oracle/diag/rdbms/cdb/cdb:
***************************************************************************************
INCIDENT_ID PROBLEM_KEY CREATE_TIME
-------------------------------------------------------------------------
9673 ORA 7445 [semtimedop] 2014-11-12 10:20:50.273000 -05:00
1 rows fetched

8. In order to view the incident in more detail, run the following command:

   adrci> show incident -mode detail -p "incident_id=9673"
ADR Home = /u01/app/oracle/diag/rdbms/cdb/cdb:
***************************************************************************************
INCIDENT INFO RECORD 1
***************************************************************************************
INCIDENT_ID 9673
NOTE: The two parameters of importance here are the PROBLEM_ID and INCIDENT_FILE.

9. When looking at the incident in further detail, the following incident file can be examined further via the following command:

   ```bash
   adrci> show trace
   /u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_9673/cdb_dbrm_7811_i9673.trc
   Output the results to file: /tmp/utsout_24439_14048_2.ado
   /bin/bash: adrci: command not found
   ```

10. Open the `/tmp/utsout_24439_14048_2.ado` file with an editor such as `vi`.

   ```bash
   /u01/app/oracle/diag/rdbms/cdb/cdb1/incident/incdir_9673/cdb_dbrm_7811_i9673.trc
   ```
Unix process pid: 7811, image: oracle@db-oracle-node1 (DBRM)

*** 2014-11-12 10:20:50.293
*** SESSION ID:(271.1) 2014-11-12 10:20:50.293
*** CLIENT ID:(271.1) 2014-11-12 10:20:50.293
*** SERVICE NAME:(SYS$BACKGROUND) 2014-11-12 10:20:50.293
*** MODULE NAME:() 2014-11-12 10:20:50.293
*** ACTION NAME:() 2014-11-12 10:20:50.293
*** CONTAINER ID:(1) 2014-11-12 10:20:50.293

Dump continued from file:
/u01/app/oracle/diag/rdbms/cdb/cdb1/trace/cdb_dbrm_7811.trc

1> ***** Error Stack *****
ORA-07445: exception encountered: core dump [semtimedop()+10] [SIGSEGV]
[ADDR:0xD431000076FD] [PC:0x3199AEB22A] [unknown code] []

1< ***** Error Stack *****
1> ***** Dump for incident 9673 (ORA 7445 [semtimedop]) *****
2> ***** Beginning of Customized Incident Dump(s) *****

Dumping swap information
Memory (Avail / Total) = 42169.27M / 64420.08M
Swap (Avail / Total) = 16415.99M / 16415.99M

Exception [type: SIGSEGV, unknown code] [ADDR:0xD431000076FD]
[PC:0x3199AEB22A, semtimedop()+10] [exception issued by pid: 30461, uid:
54321]
[flags: 0x0, count: 1]
Registers:
%rax: 0xfffffffffffffffc %rbx: 0x000000000000b %rcx:
0xffffffffffffff
[ ... Output Abbreviated ... ]

11. While this concludes how to examine trace files that pertain to a particular ORA error using ADRCI; if the issue cannot be solved by the end user, the ADRCI provides the Incident Packaging Service (IPS) tool to ZIP the necessary trace files based on the problem. It can then be sent to support for further debugging. To create the appropriate ZIP file, use the following commands:

adrci> ips create package problem 1 correlate all
Created package 1 based on problem id 1, correlation level all

NOTE: Problem 1 is the Problem_ID found in step 6.

adrci> ips generate package 1 in "/home/oracle"
Generated package 1 in file
/home/oracle/ORA7445se_20140120153727_COM_1.zip, mode complete

NOTE: Package 1 is the package ID captured from the ips create output command.

For more information about ADRCI, please visit http://docs.oracle.com/cd/E16655_01/server.121/e17639/adrci.htm#SUTIL700
Appendix O: References

TECH: Unix Semaphores and Shared Memory Explained [ID 15566.1]
https://support.oracle.com/epmos/faces/DocumentDisplay?id=15566.1

Oracle Grid Infrastructure, Oracle Documentation
http://docs.oracle.com/cd/E16655_01/install.121/e17888/concepts.htm#CWLIN506

Tuning Red Hat Enterprise Linux For Oracle & Oracle RAC by Scott Crot, Sr. Consultant, Red Hat

USE_LARGE_PAGES To Enable HugePages [ID 1392497.1]
https://support.oracle.com/epmos/faces/Dashboard?id=1392497.1

Large Pages Information in the Alert Log [ID 1392543.1]
https://support.oracle.com/epmos/faces/ui/km/SearchDocDisplay.jspx?id=1392543.1

Tuning Virtual Memory

Maximum SHMMAX values for Linux x86 and x86-64 [ID 567506.1]
https://support.oracle.com/epmos/faces/ui/km/SearchDocDisplay.jspx?id=567506.1

About the Oracle Database Fault Diagnosability Infrastructure
http://docs.oracle.com/database/121/ADMIN/diag.htm#ADMIN11257

Oracle 12c Database Installation Guide
https://docs.oracle.com/database/121/LADBI/toc.htm

Pro Oracle Database 11g RAC on Linux – Installation, Administration, Performance by Steve Shaw and Martin Bach
http://www.amazon.com/Pro-Oracle-Database-11g-Linux-ebook/dp/B004VJ472I/ref=sr_1_1?ie=UTF8&qid=1389977723&sr=8-1&keywords=pro+oracle+11g+rac

33 My Oracle Support Credentials Required to access article