Reference Architectures 2017

Deploying Oracle Database 12c Release 2 on Red Hat Enterprise Linux 7

Last Updated: 2018-04-05

Reference Architectures 2017 Deploying Oracle Database 12c Release 2 on Red Hat Enterprise Linux 7

Roger Lopez
refarch-feedback@redhat.com

Legal Notice

Abstract

This document provides the best practices to deploy Oracle Database 12c Release 2 on Red Hat Enterprise Linux 7.

Table of Contents

COMMENTS AND FEEDBACK

In the spirit of open source, we invite anyone to provide feedback and comments on any reference architecture. 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.

CHAPTER 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 Database 12c Release 2 with Oracle Automatic Storage Management (ASM). It is suited for system, storage, and database administrators deploying Oracle Database 12c Release 2 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 2
- Deploying Oracle Database Software 12c Release 2
- Deploying an Oracle Database 12c Release 2 with shared iSCSI disks
- Using Oracle ASM disks with udev rules
- Securing the Oracle Database 12c Release 2 environment with *SELinux*

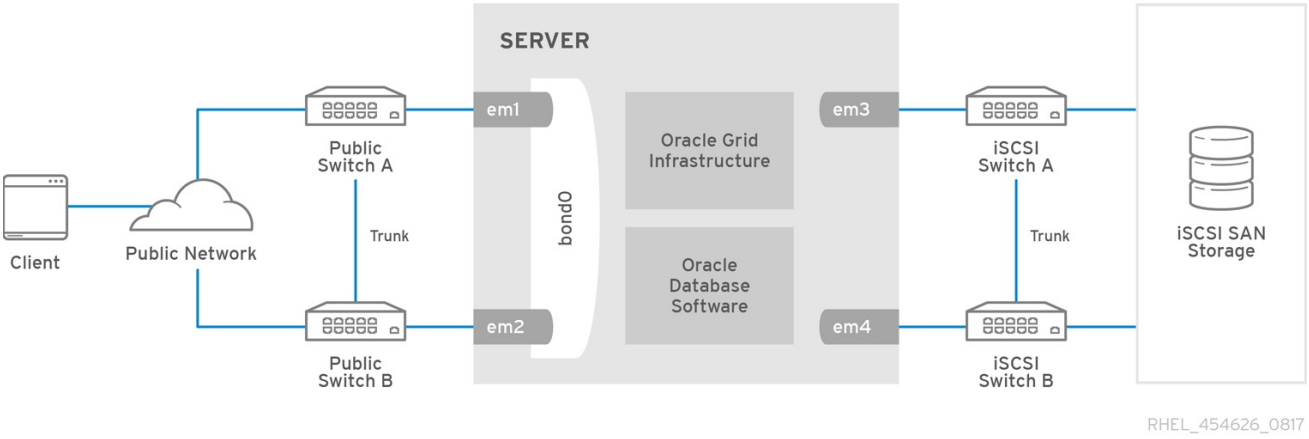
CHAPTER 2. REFERENCE ARCHITECTURE ENVIRONMENT

This section focuses on the components used during the deployment of Oracle Database 12c Release 2 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, “Reference Architecture Overview”

Figure 2.1. Reference Architecture Overview



2.2. NETWORK TOPOLOGY

The network topology in this reference environment consists of two public switches, and two iSCSI storage switches. *Public Switch A* and *Public Switch B*, with a link aggregation that connect them together creating a single logical switch. Ethernet device **em1** on the server connects to *Public Switch A*, while Ethernet device **em2** on the server connects to *Public Switch B*. Ethernet devices **em1** and **em2** are bonded together as a bond device, **bond0**, providing high availability for the public network traffic. Figure 2.2, “Network Bonding” shows the pictorial representation of the two public switches connecting to the server and the Ethernet bonding of device **em1** and **em2** as part of the **bond0** device. *iSCSI Switch A* and *iSCSI Switch B* also use a link aggregation that connects them together creating a single logical switch. Ethernet device **em3** on the server connects to *iSCSI Switch A* and **em4** on the server connects to *iSCSI Switch B*. It is recommended that **em3** and **em4** be 10GB Network cards for better performance when accessing the storage. Figure 2.3, “iSCSI Switch Connectivity” shows a pictorial representation of the connectivity of the Ethernet devices to the iSCSI switches.

Figure 2.2. Network Bonding

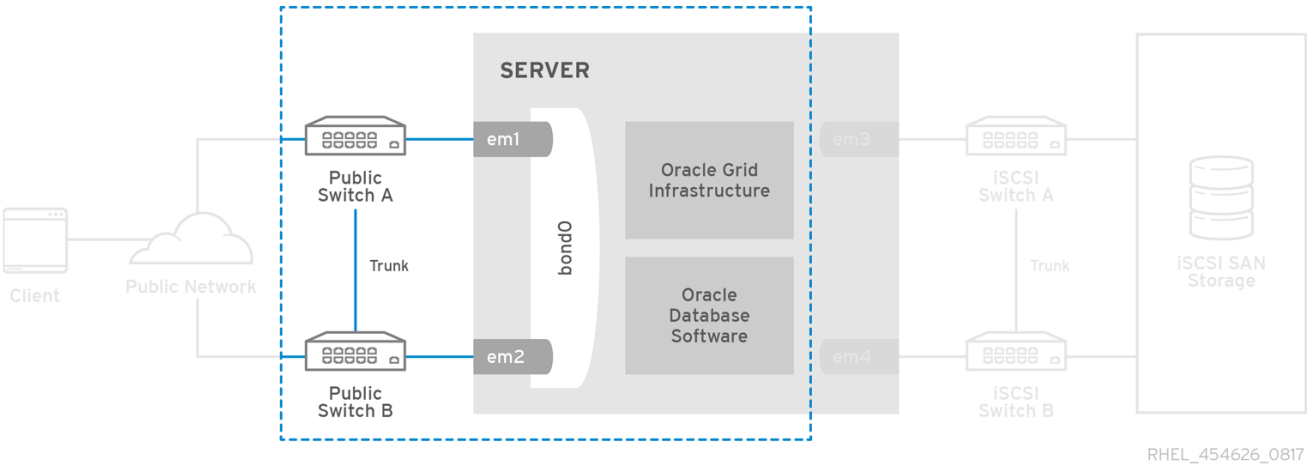
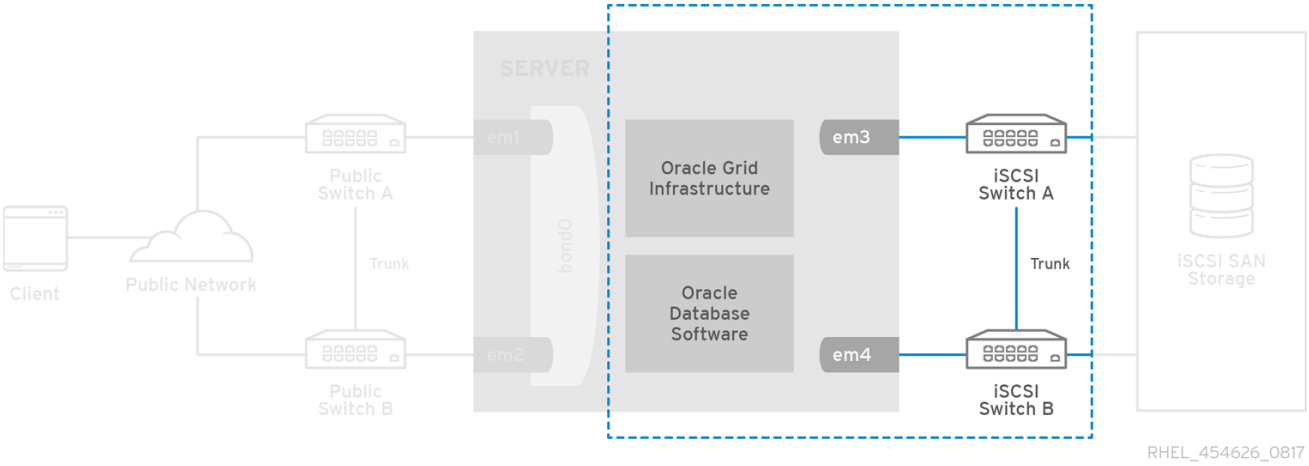


Figure 2.3. iSCSI Switch Connectivity



2.3. HARDWARE DETAILS

The following are the hardware requirements to properly install Oracle Database 12c Release 2 on a x86_64 system:

- Minimum of 8 GB of RAM for the installation of Oracle Grid Infrastructure
- Minimum of 1 GB of RAM for the installation of Oracle Database, however 2 GB of memory or more is recommended
- Minimum of 1 Network Interface Card (NIC), however 2 NICs are recommended for high availability (HA) as used in the reference environment
- Red Hat Enterprise Linux 7 with kernel 3.10.0-123.el7.x86_64 or higher
- Console access that supports 1024 x 768 for the Oracle Universal Installer (OUI)

Table 2.1, “Server Details” specifies the hardware for the server within this reference environment. This hardware meets the minimum requirements for properly installing Oracle Database 12c Release 2 on a x86_64 system.

Table 2.1. Server Details

Server Hardware	Specifications

Oracle Database 12c Release 2 Standalone Server (db-oracle-node1) [1 x PowerEdge M520]	Red Hat Enterprise Linux 7 3.10.0-514.el7.x86_64
	2 socket, 8 core, 16 threads, Intel® Xeon® CPU E5-2450 0 @ 2.10GHz
	96 GB of memory, DDR3 16384 MB @ 1600 Mhz DIMMs
	2x NetXtreme BCM5720 Gigabit Ethernet PCIe for public network traffic
	2x NetXtreme II BCM57810 10 Gigabit Ethernet for iSCSI network traffic

Table 2.2, “Switch Details” specifies the switches within this reference environment.

Table 2.2. Switch Details

Switch Hardware	
2 x Dell PowerConnect M6348	2 x Dell PowerConnect M8024-k

Table 2.3, “Storage Details” specifies the storage within this reference environment.

Table 2.3. Storage Details

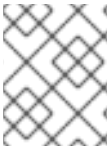
Storage Hardware
Dell Equallogic PS Array

2.4. FILE SYSTEM LAYOUT & DISK SPACE DETAILS

The following is the disk space requirements for properly installing Oracle Database 12c Release 2 software for this reference environment.

Table 2.4. Disk Space Requirements

Software	Disk Space
Oracle Grid Infrastructure Home (includes software files)	12 GB
Oracle Database Home Enterprise Edition (includes software files and data files)	12 GB
/tmp	1 GB



NOTE

The actual amount of disk space consumed for Oracle Grid Infrastructure Home and Oracle Database Home Enterprise Edition may vary.

Table 2.5, “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 Database 12c Release 2

Table 2.5. File System Layout

File System Layout	Disk Space Size
/	15 GB
/boot	250 MB
/home	8 GB
/tmp	4 GB
/u01	50 GB
/usr	5 GB
/var	8 GB

While the size of the Oracle data files varies for each solution, the following are the Oracle data file sizes for this reference environment.

Table 2.6. Oracle Data File Sizes for Reference Architecture

Volume	Volume Size	RAID Group Type	Redundancy
Database Volume 1 (db1)	100 GB	RAID 10	External
Database Volume 2 (db2)	100 GB	RAID 10	External
Fast Recovery Area (fra)	200 GB	RAID 5	External
Oracle Redo Log Volume (redo)	10 GB	RAID 1	External

2.5. 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 2.7. Recommended Swap Space

RAM	Swap Space
2 GB up to 16 GB	Equal to the size of RAM
Greater than 16 GB	16 GB of RAM

NOTE

When calculating swap space, ensure not to include RAM assigned for hugepages. More information on hugepages can be found in Section 4.5, “Enabling HugePages”

2.6. 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[1]. **`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 while the **`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[2]. 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 2.8, “Firewall Settings” lists the enabled ports in this reference environment.

1: Linux man pages - man (1) firewalld

2: 4.5.3 Comparison of firewalld to system-config-firewalld and iptables

Table 2.8. Firewall Settings

Port	Protocol	Description
22	TCP	Secure Shell (SSH)
443	TCP	Hypertext Transfer Protocol over SSL/TLS (HTTPS)
1521	TCP	Oracle Transparent Network Substrate (TNS) Listener default port
5500	TCP	EM Express 12c default port

2.7. 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.

CHAPTER 3. REFERENCE ARCHITECTURE CONFIGURATION DETAILS

This reference architecture focuses on the deployment of Oracle Database 12c Release 2 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 2
- Oracle Database 12c Release 2 Software Installation
- Deploying an Oracle Database 12c Release 2 with iSCSI disks
- Enabling Security-Enhanced Linux (*SELinux*)
- Configuring Device Mapper Multipathing
- Using *udev* rules instead of Oracle ASMLib or Oracle ASM Filter Driver

3.1. SETTING OS HOSTNAME

A unique host name is required for the installation of Oracle Database 12c Release 2. The host name within this reference environment is: **`oracle1.e2e.bos.redhat.com`**.

To set a hostname for a server use the **`hostnamectl`** command. An example of setting **`oracle1.e2e.bos.redhat.com`** hostname is shown below.

```
# hostnamectl set-hostname oracle1.e2e.bos.redhat.com
```

Verify the status:

```
# hostnamectl status
   Static hostname: oracle1.e2e.bos.redhat.com
         Icon name: computer-server
           Chassis: server
        Machine ID: f9650ab62cd449b8b2a02d39ac73881e
           Boot ID: 4b4edc0eb2d8418080d86e343433067f
  Operating System: Storage
       CPE OS Name: cpe:/o:redhat:enterprise_linux:7.3:GA:server
            Kernel: Linux 3.10.0-514.el7.x86_64
      Architecture: x86-64
```

3.2. NETWORK CONFIGURATION

The network configuration focuses on the proper creation of a bonded network interface. The bonded network interface provides an Oracle environment with high availability in case of a public network interface failure.

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[3]. 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 system **oracle1** 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 the reference environment is shown below.

```
# cat /etc/resolv.conf
# Generated by NetworkManager
search e2e.bos.redhat.com
nameserver 10.19.114.2
```

3: Linux man pages - man resolv.conf

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 **em1** and **em2** with a bond device labeled **bond0**.

NOTE

The usage of NetworkManager is optional.

Check the status of Network Manager:

```
# systemctl status NetworkManager.service
\u25cf NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service;
enabled; vendor preset: enabled)
   Active: active (running) since Tue 2017-07-18 02:29:13 UTC; 2 days ago
     Docs: man:NetworkManager(8)
 Main PID: 2038 (NetworkManager)
   CGroup: /system.slice/NetworkManager.service
           \u251c\u25002038 /usr/sbin/NetworkManager --no-daemon
           \u2514\u25002140 /sbin/dhclient -d -q -sf /usr/libexec/nm-dhcp-
helper -pf /var/run/dhclient-em1.pid -lf /var/lib/NetworkManager/dhclie...
```

Create a channel bonding interface:

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond0
DEVICE=bond0
NAME=bond0
BONDING_MASTER=yes
BOOTPROTO=static
IPADDR=10.19.114.44
PREFIX=23
GATEWAY=10.19.115.254
BONDING_OPTS="mode=1 miimon=100 primary=em1"
ONBOOT=yes
```

Create **em1** and **em2** as slave interfaces:

```
# cat /etc/sysconfig/network-scripts/ifcfg-em1
DEVICE=em1
HWADDR="44:a8:42:af:58:66"
ONBOOT=yes
IPV6INIT=no
PEERROUTES=yes
SLAVE=yes
BOOTPROTO="none"
MASTER=bond0
```

```
# cat /etc/sysconfig/network-scripts/ifcfg-em2
DEVICE=em2
HWADDR="44:a8:42:af:58:67"
ONBOOT=yes
IPV6INIT=no
PEERROUTES=yes
SLAVE=yes
BOOTPROTO="none"
MASTER=bond0
```

Restart the **network** service

```
# systemctl restart network.service
```

To ensure **NetworkManager** is aware of the changes issue the command:

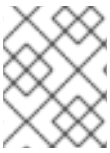
```
# nmcli con reload
```

NOTE

If for some reason, issues getting **bond0** properly to add the different interfaces, reboot the host.

Once the **bond0** device is configured on the host, ensure connectivity by pinging the gateway IP.

```
# ping 10.19.115.254
PING 10.19.115.254 (10.19.115.254) 56(84) bytes of data.
64 bytes from 10.19.115.254: icmp_seq=1 ttl=64 time=0.978 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:

Table 3.1. Public IP & Hostname

IP	Hostname

10.19.114.44	oracle1.e2e.bos.redhat.com

3.2.3. iSCSI Network Configuration

The following section only applies to environments taking advantage of iSCSI storage. If not using an iSCSI storage array, please skip to the following section Section 3.3, “OS Configuration”.

The iSCSI network configuration consists of two network interfaces **em3** and **em4**. Set **em3** and **em4** for iSCSI traffic. An example below:

```
# cat /etc/sysconfig/network-scripts/ifcfg-em3
DEVICE=em3
HWADDR="44:a8:42:af:52:61"
ONBOOT=yes
IPV6INIT=no
BOOTPROTO=static
IPADDR=172.17.114.250
PREFIX=24
MTU=9000
```

```
# cat /etc/sysconfig/network-scripts/ifcfg-em4
DEVICE=em4
HWADDR="44:a8:42:af:52:62"
ONBOOT=yes
IPV6INIT=no
BOOTPROTO=static
IPADDR=172.17.114.251
PREFIX=24
MTU=9000
```

NOTE

It is recommended to take advantage of Jumbo Frames for iSCSI storage. Ensure that the iSCSI switches have Jumbo Frames enabled.

Stop and start the network interface

```
ifdown em3; ifdown em4; ifup em3; ifup em4
```

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

```
# ping <Equallogic Group IP>
```

3.2.3.1. iSCSI Switch and Dell EqualLogic Recommendations

Regarding the Dell EqualLogic PS Array, the following are recommendations to achieve optimal performance.

- Create an isolated network for iSCSI traffic, i.e. VLANs

- A trunk between the switches that equals the ttoal amount of bandwidth available on the EqualLogic PS Array
- Enable Rapid Spanning Tree Protocol (RSTP) on the iSCSI switches
- Enable PortFast within the switch ports on the iSCSI switches
- Enable Flow Control within the switch ports on the iSCSI switches
- Disable unicast storm control within the switch ports on the iSCSI switches
- Enable Jumbo Frames on the iSCSI switches

3.3. OS CONFIGURATION

3.3.1. Red Hat Subscription Manager

The **subscription-manager** command registers a system to the Red Hat Network (RHN) and manages the subscription entitlements for a system. The **--help** option specifies 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. In order 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 does not prompt for the RHN Network authentication credentials.

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

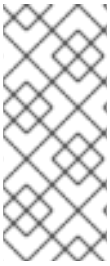
```
# subscription-manager register --username [User] --password '[Password]'
The system has been registered with id: abcd1234-ab12-ab12-ab12-
481ba8187f60
```

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. 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: <poolid>
Available: 47
Suggested: 1
Service Level: STANDARD
Service Type: L1-L3
```

```
# subscription-manager attach --pool <pool_id>
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. Enable the supplementary repository via 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 Database 12c Release 2 install on Red Hat Enterprise Linux 7 and to install the custom tuned profile labeled **tuned-profiles-oracle**. The packages are only available in the **rhel-7-server-optional-rpms** repository.

```
# subscription-manager repos --enable=rhel-7-server-optional-rpms
Repo 'rhel-7-server-optional-rpms' is enabled for this system.
```

For more information on the use of Red Hat Subscription Manager, please visit the Red Hat Subscription management documentation[4].

4: Red Hat Subscription Management

3.3.2. Chrony Configuration

The **chronyd** — is a daemon for synchronisation of the system clock. It can synchronise the clock with NTP servers, reference clocks (e.g. a GPS receiver), and manual input using wristwatch and keyboard via chronyd. It can also operate as an NTPv4 (RFC 5905) server and peer to provide a time service to other computers in the network[5].

5: chronyd - chronyd daemon man page – man chronyd (8)

In order to configure the **chronyd** daemon, follow the instructions below.

1. If not installed, install chrony via yum as follows:

   ```
   # yum install chrony
   ```

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

   ```
   # vi /etc/chrony.conf
   ```

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

5. Start the **chronyd** daemon via the command:

```
# systemctl start chronyd.service
```

6. Ensure that the **chronyd** daemon is started when the host is booted.

```
# systemctl enable chronyd.service
```

7. Verify the **chronyd** daemon status.

```
# systemctl status chronyd.service
● chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled;
vendor preset: enabled)
   Active: active (running) since Mon 2018-01-08 17:16:07 UTC; 2
months 26 days ago
     Docs: man:chronyd(8)
           man:chrony.conf(5)
  Process: 815 ExecStartPost=/usr/libexec/chrony-helper update-
daemon (code=exited, status=0/SUCCESS)
  Process: 732 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited,
status=0/SUCCESS)
 Main PID: 754 (chronyd)
   CGroup: /system.slice/chronyd.service
           └─754 /usr/sbin/chronyd
```

3.3.3. Oracle Database 12c Release 2 Package Requirements

A specific set of packages is required to properly deploy Oracle Database 12c Release 2 on Red Hat Enterprise Linux 7. 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 H, *Sample Kickstart File*. Red Hat Enterprise Linux 7 installation requires the following group packages:

Table 3.2. Group Packages

Required Group Packages	
@Base	@Core

Oracle Grid Infrastructure and Oracle Database 12c Release 2 required x86_64 RPM packages.

Table 3.3. Required Packages

Required Packages			
binutils	libX11	compat-libcap1	libXau

compat-libstdc++-33	libaio	gcc	libaio-devel
gcc-c++	libdmx	glibc-devel	glibc
ksh	make	libgcc	sysstat
libstdc++	xorg-x11-utils	libstdc++-devel	xorg-x11-xauth
libXext	libXv	libXtst	libXi
libxcb	libXt	libXmu	libXxf86misc
libXxf86dga	libXxf86vm	nfs-utils	smartmontools

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 D, *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 oracle1.e2e.bos.redhat.com
```

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.

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 Database 12c Release 2.

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

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
```

For more information on Security Enhanced Linux, please visit the Red Hat Enterprise Linux 7 Security - Enhanced Linux User Guide

3.3.5. Configuring Firewall Settings

Firewall access and restrictions play a critical role in securing your Oracle Database 12c Release 2 environment. The use of 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[6]. The key difference and advantage of **firewalld** over the **iptables** service is it does not require flushing of the old firewall rules to apply the new firewall rules. **firewalld** changes the settings during runtime without losing existing connections[6]. 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.

```
# 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="10.16.142.54" port protocol="tcp" port="5500" accept"
success
```

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

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

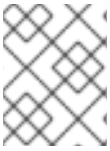
To verify the port 1521 has been added and database client with IP address of 10.19.142.54 has been properly added to access port 5500, run the following command:

```
# firewall-cmd --zone=public --list-all
public (default, active)
 interfaces: bond0 em1 em2
 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
```

6: Red Hat Enterprise Linux 7 – Using Firewalls

3.3.6. Modifying Kernel Parameters

The following sections regarding virtual memory, shared memory, semaphores, network ports, I/O synchronous requests, file handles, and kernel panic on OOPS parameters provide a detailed explanation of what these parameters are and their effect in an Oracle deployment. It is recommended to read carefully each parameter for a better understanding on how to tweak a specific environment for a particular workload.



NOTE

The recommended values listed are to be used as a starting point when setting virtual memory, there is no "one-size fits all" regarding performance tuning.

Each section provides the manual steps to tweaking the parameters. With that said, if looking to tweak parameters immediately, Section 3.4.6, “Optimizing Database Storage using Automatic System Tuning” covers setting the parameters using the **`oracle-tuned-profile`**.

3.3.7. Setting Virtual Memory

Tuning virtual memory requires the modification of five kernel parameters that affect the rate that virtual memory is used within Oracle databases.

A brief description[7] and recommended settings for the virtual memory parameters, as well as, the definition of dirty data are described below.

SWAPPINESS[7] - 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 that 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, that 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 between 1 and 20. The recommendation of swappiness for Red Hat Enterprise Linux 6.4 or higher running Oracle databases is now a value between 1-20.

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[7] – Contains, as a percentage of total system memory, the number of pages at which a process that is generating disk writes will itself start writing out dirty data. The default value is 20. The recommended value is between 40 and 80. The reasoning behind increasing the value from the standard Oracle 15 recommendation to a value between 40 and 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 the value of dirty_ratio is increased, it is less likely that all processes will be blocked due to synchronous I/O, however, this allows for more data to be sitting in memory that has yet to be written to disk.

DIRTY_BACKGROUND_RATIO[7] – Contains, as a percentage of total system memory, the number of pages that the background write back daemon will start writing out dirty data. The Oracle recommended value is 3.

NOTE

An example 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 asynchronously, however, non 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[7] - 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[7] - 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.

Create a file labeled *98-oracle-kernel.conf* within */etc/sysctl.d/*

-

```
# vi 98-oracle-kernel.conf
vm.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:

```
# sysctl -p /etc/sysctl.d/98-oracle-kernel.conf
```

NOTE

A full listing of all the kernel parameters modified within the */etc/sysctl.d/98-oracle-kernel.conf* file can be found at Appendix E, *Kernel Parameters (98-oracle-kernel.conf)*.

7: RHEL7 Kernel Documentation (requires package **`kernel-doc`** to be installed) - */usr/share/doc/kernel-doc-3.10.0/Documentation/sysctl/vm.txt*

3.3.8. Setting Shared Memory (SHMMAX, SHALL, 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 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 7.0 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 is 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.

Starting with Red Hat Enterprise Linux 7.1 and above, *SHMMAX* default value is set to 18446744073692774399 bytes, equivalent to roughly 18 petabytes. Due to this, there is no need to calculate *SHMMAX* because of the very large size already provided. It is recommended to use the value set in Red Hat Enterprise Linux 7.1 and above because the value is purposely set higher than the architectural memory limits to ensure that any Oracle SGA value set within an Oracle database instance may fit in one single shared memory segment.

The value of *SHMMAX* can be confirmed via the command:

```
# sysctl kernel.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 7.0 x86_64 provides a *SHMALL* value of 268435456 pages, the equivalent of 1 TB in system pages. This is determined by the following formula:

SHMALL IN BYTES * PAGE_SIZE

Starting with Red Hat Enterprise Linux 7.1 and above, *SHMALL* default value is 18446744073692774399 pages, the same value set to *SHMMAX*.

The value of *SHMALL* can be confirmed via the command:

```
# sysctl kernel.shmall
```

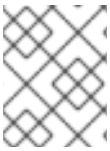
To ensure an adequate amount of memory pages are allocated to a single Oracle SGA, it is recommended that the value of *SHMALL* be set to the at least the value using the following formula:

SHMMAX IN BYTES / PAGE_SIZE

Since the default value of *SHMALL* in Red Hat Enterprise Linux 7.1 and above is 18446744073692774399 pages, and the minimum recommended value by Oracle for *SHMALL* is 1073741824, the larger default value is kept.

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 Red Hat Enterprise Linux 7 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.

Prior to Red Hat Enterprise Linux 7.1, changes to the kernel parameters were required. However, with the new *SHMMAX*, *SHMALL*, and *SHMMNI* defaults no changes are made.



NOTE

A full listing of all the kernel parameters modified within the */etc/sysctl.d/98-oracle-kernel.conf* file can be found at Appendix E, *Kernel Parameters (98-oracle-kernel.conf)*.

3.3.9. 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

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

NOTE

SEMMNS is calculated by SEMMSL * SEMMNI

The following line is required within the */etc/sysctl.d/98-oracle-kernel.conf* file to provide default values for semaphores for Oracle:

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

For the changes to take effect immediately, run the following command:

```
# sysctl -p /etc/sysctl.d/98-oracle-kernel.conf
```

The values above are sufficient for most environments and no tweaking should be necessary. However, the following describes how these values can be optimized and should be set when the defaults don't suffice.

Example errors:

```
ORA-27154: post/wait create failed
ORA-27300: OS system dependent operation:semget failed with status: 28
ORA-27301: OS failure message: No space left on device
ORA-27302: failure occurred at: sskgpcreates
```

NOTE

It is recommended to first use the default values and adjust only when deemed necessary.

Semaphores are used by Oracle for internal locking of SGA structures. Sizing of semaphores directly depends on ***only*** the *PROCESSES* parameter of the instance(s) running on the system. The number of semaphores to be defined in a set should be set to a value that minimizes the waste of semaphores.

For example, say our environment consists of two Oracle instances with the *PROCESSES* set to 300 for database one and 600 for database two. With *SEMMSL* set at 250 (default), the first database requires 2 sets. The first set is 250 semaphores but an additional 50 semaphores is required thus an additional *SEMMSL* set is required thus wasting 200 semaphores. Our 2nd instance requires 3 sets, set one 250 semaphores, set two 250 semaphores, giving us a total of 500, but an additional 100 semaphores is required thus adding an additional *SEMMSSL* set wasting 150 semaphores. A better value of *SEMMSL* in this particular case would be 150. With *SEMMSL* set at 150, the first database requires two sets (wasting zero semaphores), our second instance requires four sets (wasting zero semaphores). This is an ideal example, and most likely some semaphore wastage is expected and okay as semaphores in general consume small amounts of memory. As more databases are created in an environment, these calculations may get complicated. In the end, the goal is to limit semaphore waste.

Regarding *SEMMNI*, *SEMMNI* should be set high enough for proper amount of sets to be available on the system. Using the value of *SEMMSL*, one can determine max amount of *SEMMNI* required. Round up to the nearest power of 2.

SEMMNI = *SEMMNS*/*SEMMSL*

Oracle requires 2x value of *PROCESSES* in the **init.ora** parameter for semaphores (*SEMMNS* value) on startup of the database, then half of those semaphores are released. To properly size *SEMMNS*, one must know the sum of all *PROCESSES* set across all instances on the host. *SEMMNS* should best be set higher than *SEMMNI*SEMMSL* value (this is how we get 32000 for default value 250*128)

SEMOP is calculated using the total *SEMMNI* divided by *SEMMSL*. In the default scenario that is 3200/250 = 128

3.3.10. 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.d/98-oracle-kernel.conf* file and add the following line:

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

For the changes to take effect immediately, run the following command:

```
# sysctl -p /etc/sysctl.d/98-oracle-kernel.conf
```

3.3.11. 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 *RMEM* and *WMEM*. The *RMEM* parameter represents the receive buffer size, while the *WMEM* represents the send buffer size. The recommended values by Oracle are configured within the **/etc/sysctl.conf** file.

```
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
```

For the changes to take effect immediately, run the following command:

```
# sysctl -p /etc/sysctl.d/98-oracle-kernel.conf
```

3.3.12. 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.d/98-oracle-kernel.conf* file as follows:

```
fs.aio-max-nr = 1048576
```

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

```
# sysctl -p /etc/sysctl.d/98-oracle-kernel.conf
```

3.3.13. 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 database instance found within a system, allocate 512**PROCESSSES* 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 database by the **oracle** user. The default value for *PROCESSES* is 2560 for Oracle Database 12c Release 2. 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:

```
sysctl fs.file-max
```

Next, add all the *PROCESSES* together from each Oracle database instance found within the system and multiple by 512 using **bc** as seen in the following command.

```
# yum install bc
# echo “512 * 2560” | bc
```

NOTE

To determine the current *PROCESSES* value, log into each Oracle database instance and run the following command below. Since no Oracle database has yet been created within this reference environment, the default value of 2560 *PROCESSES* is used.

```
SQL> show parameter processes;
NAME                                 TYPE VALUE
------------------------------------ -----------
processes                            integer 2560
```

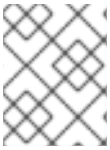
Finally, add the current *FS.FILE-MAX* value with the new value found from multiplying 512**PROCESSES* to attain the new *FS.FILE-MAX* value.

While the value of the *FS.FILE-MAX* parameter varies upon every environment, this reference environment uses the default value within Red Hat Enterprise Linux 7.4 (9784283). Oracle recommends a value no smaller than 6815744. In order to modify the value of *FS.FILE-MAX*, add to the_/etc/sysctl.d/98-oracle-kernel.conf_ file as follows:

```
fs.file-max = <value>
```

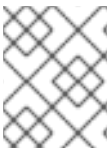
In order for the changes take effect immediately, run the following command:

```
# sysctl -p etc/sysctl.d/98-oracle-kernel.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.d/98-oracle-kernel.conf* file can be found at Appendix E, *Kernel Parameters (98-oracle-kernel.conf)*.

3.3.14. User Accounts & Groups

Prior to the installation of Oracle Database 12c Release 2, Oracle recommends the creation of a **grid** user for the Oracle Grid Infrastructure and an **oracle** user for the Oracle Database software installed on the system.

For the purposes of this reference environment, the Oracle Grid Infrastructure 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 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

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 (*SYSDG* privilege) to a user for administering and monitoring Oracle Data Guard

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

OSRACDBA group (*RACDBA* privilege) - grants the SYSRAC privileges to perform administrative tasks on an Oracle RAC cluster.

NOTE

RACDBA group is still used even within the Oracle Database Standalone server.

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

OSDBA group (*ASMDBA* privilege) – provides administrative access to Oracle ASM instances

OSASM group (*ASMADMIN* privilege) – provides administrative access for storage files via the SYSASM privilege

OSOPER group (*ASMOPER* privilege) – an optional group created to assign limited DBA privileges with regards to ASM to particular OS user accounts

OSRACDBA group (*RACDBA* privilege) - grants the SYSRAC privileges to perform administrative tasks on an Oracle RAC cluster.

NOTE

RACDBA group is still used even within the Oracle Database Standalone server.

As the **root** user, 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
# groupadd --gid 54330 racdba
# useradd --uid 54321 --gid oinstall --groups dba,oper,asmdba,racdba,\
> backupdba,dgdba,kmdba oracle
# passwd oracle
# useradd --uid 54322 --gid oinstall --groups
dba,asmadmin,asmdba,asmoper,\
> racdba grid
# passwd grid
```

Verify the **oracle** and **grid** user correctly display the appropriate primary and supplementary groups via the commands:

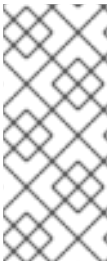
```
# id oracle
uid=54321(oracle) gid=54321(oinstall) groups=54321(oinstall),54322(dba),
54323(asmdba),54326(oper),54327(backupdba),54328(dgdba),54329(kmdba),
54330(racdba)

# id grid
uid=54322(grid) gid=54321(oinstall) groups=54321(oinstall),54322(dba),
54323(asmdba),54324(asmoper),54325(asmadmin),54330(racdba)
```

3.3.15. 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:

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



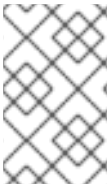
NOTE

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?[8]

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 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft stack 10240
oracle hard stack 32768
grid soft nproc 2047
grid hard nproc 16384
grid soft nofile 1024
grid hard nofile 65536
grid soft stack 10240
grid hard stack 32768
```

Due to Bug 1597142116, 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.

8: What order are limits files in the limits.d directory read in?

As the **root** user, 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
fi
if [ $USER = "grid" ]; then
 if [ $SHELL = "/bin/ksh" ]; then
 ulimit -u 16384
 ulimit -n 65536
 else
 ulimit -u 16384 -n 65536
 fi
fi
```

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 I, *Configuration Files*

As **oracle** and **grid** user, verify the *ULIMIT* values by running the following command:

```
# ulimit -a
core file size (blocks, -c) 0
data seg size (kbytes, -d) unlimited
scheduling priority (-e) 0
file size (blocks, -f) unlimited
pending signals (-i) 385878
max locked memory (kbytes, -l) 14854144
max memory size (kbytes, -m) unlimited
open files (-n) 65536
pipe size (512 bytes, -p) 8
POSIX message queues (bytes, -q) 819200
real-time priority (-r) 0
stack size (kbytes, -s) 10240
cpu time (seconds, -t) unlimited
max user processes (-u) 16384
virtual memory (kbytes, -v) unlimited
file locks (-x) unlimited
```

3.4. STORAGE CONFIGURATION

The following storage configuration section describes the best practices for setting up iSCSI CHAP Authentication, configuring host access to volumes, 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. iSCSI CHAP Authentication

This section applies to users taking advantage of iSCSI storage. If not using iSCSI storage, please skip to section Section 3.4.3, “Device Mapper Multipath”.

For security purposes, CHAP (Challenge-Handshake Authentication Protocol) is used to validate the identity of the node(s) connecting to it. The process includes creating a secret username and password to authenticate on each node(s). The details on enabling CHAP within the iSCSI storage itself may vary depending on the vendor. Within the Dell EqualLogic PS Array the steps are as follows:

- Within the left navigation var, select *Group Configuration*
- Within the right pane, select the *iSCSI* tab
- Within the *Local CHAP Accounts* section select *Add*
- Within the popup dialog box, enter the appropriate credentials and select OK.

Once the CHAP user is created within the iSCSI storage array, the following steps are required for each node(s).

1. Install **iscsi-initiator-utils** package
   -

```
# yum install -y iscsi-initiator-utils
```

2. Modify the */etc/iscsi/iscsid.conf* file with the CHAP credentials. An example below only shows the sections modified within CHAP Settings.

```
# *************
# CHAP Settings
# *************

# To enable CHAP authentication set node.session.auth.authmethod
# to CHAP. The default is None.
node.session.auth.authmethod = CHAP

# To set a CHAP username and password for initiator
# authentication by the target(s), uncomment the following lines:
node.session.auth.username = <username>
node.session.auth.password = <password>

# To enable CHAP authentication for a discovery session to the
target
# set discovery.sendtargets.auth.authmethod to CHAP. The default is
None.
discovery.sendtargets.auth.authmethod = CHAP

# To set a discovery session CHAP username and password for the
initiator
# authentication by the target(s), uncomment the following lines:
# authentication by the target(s), uncomment the following lines:
discovery.sendtargets.auth.username = <username>
discovery.sendtargets.auth.password = <password>
```

3. Start the iSCSI service and enable it persistently across reboots

```
# systemctl start iscsid.service
# systemctl enable iscsid.service
```

4. Verify the iSCSI service started

```
# systemctl status iscsid.service
```

3.4.2. Configuring Host Access to Volumes

The following section provides steps in connecting the Dell EqualLogic iSCSI volumes to be used for the Oracle installation.

As the **root** user,

1. Verify Ethernet devices **em3** and **em4** can ping the Dell EqualLogic group IP.

```
# ping -I em3 <EqualLogic_Group_IP>

# ping -I em4 <EqualLogic_Group_IP>
```

2. Create an iSCSI interface (**iface**) for each storage NIC. While the interface can have any

name, for easy identification purposes the **iface** are labeled **iem3 iem4**.

```
# iscsiadm –m iface –I iem3 --op=new
New interface iem3 added

# iscsiadm –m iface –I iem4 --op=new
New interface iem4 added
```

3. Associate the iSCSI interface to the corresponding Ethernet device

```
# isciadm -m iface -I iem3 --op=update -n iface.net_ifacename -v em3
iem3 updated.

# isciadm -m iface -I iem4 --op=update -n iface.net_ifacename -v em4
iem4 updated.
```

4. Verify the iSCSI interface configuration

```
# iscsiadm -m iface
iem3 tcp,<empty>,<empty>,em3,<empty>
iem4 tcp,<empty>,<empty>,em4,<empty>
```

5. Discover the iSCSI targets

```
# iscsiadm -m discovery -t st -p <EqualLogic_Group_IP> -I iem3 -I
iem4
```

6. Login the iSCSI targets

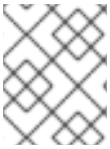
```
# iscsiadm -m node --login all
```

7. Verify the iSCSI sessions are logged in

```
# iscsiadm -m session
```

3.4.3. Device Mapper 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 database volumes are accessible via the operating system prior to continuing with the section below

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

```
# yum install device-mapper-multipath
```

2. Create the *multipath.conf* file in */etc/*

```
# mpathconf --enable
```

3. Capture the scsi id of the local disk(s) on the system. This example assumes the local disk is located within **/dev/sda**

```
# /usr/lib/udev/scsi_id --whitelisted --replace-whitespace \
--device=/dev/sda
3600508b1001030353434363646301200
```

4. Modify the *blacklist* section at the bottom of 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.

NOTE

Notice how the **wwid** matches the value found in the previous step.

```
blacklist {
 wwid 3600508b1001030353434363646301200
 devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
 devnode "^hd[a-z]"
}
```

5. Start the multipath daemon.

```
# systemctl start multipathd.service
```

6. Enable the multipath daemon to ensure it is started upon boot time.

```
# 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:

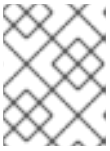
```
# multipath -ll
```

Figure 3.1. Multipath Device (mpathb)

Figure 3.1, “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.

8. The default values used by **`device-mapper-multipath`** can be seen using the command **`multipathd show config`**. Below is an example of the default output.

```
defaults {
        verbosity 2
        polling_interval 5
        max_polling_interval 20
        reassign_maps "yes"
        multipath_dir "/lib64/multipath"
        path_selector "service-time 0"
        path_grouping_policy "failover"
        uid_attribute "ID_SERIAL"
        prio "const"
        prio_args ""
        features "0"
        path_checker "directio"
        alias_prefix "mpath"
        failback "manual"
        rr_min_io 1000
        rr_min_io_rq 1
        max_fds 1048576
        rr_weight "uniform"
        queue_without_daemon "no"
        flush_on_last_del "no"
        user_friendly_names "yes"
        fast_io_fail_tmo 5
        bindings_file "/etc/multipath/bindings"
        wwids_file /etc/multipath/wwids
        log_checker_err always
        find_multipaths yes
        retain_attached_hw_handler no
        detect_prio no
        detect_path_checker no
        hw_str_match no
        force_sync no
        deferred_remove no
        ignore_new_boot_devs no
        skip_kpartx no
        config_dir "/etc/multipath/conf.d"
        delay_watch_checks no
        delay_wait_checks no
        retrigger_tries 3
        retrigger_delay 10
        missing_uev_wait_timeout 30
        new_bindings_in_boot no
        remove_retries 0
        disable_changed_wwids no
}
```



NOTE

The standard options can be customized to better fit the 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 2.6, "Oracle Data File Sizes for Reference Architecture".

```
multipaths {
    multipath {
    wwid 3600c0ff000d7e7a899d8515101000000
    alias db1
    }
    multipath {
    wwid 3600c0ff000dabfe5a7d8515101000000
    alias db2
    }
    multipath {
    wwid 3600c0ff000d7e7a8dbd8515101000000
    alias fra
    }
    multipath {
    wwid 3600c0ff000dabfe5f4d8515101000000
    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 EQL,100E-00
size=186G features='0' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
| |- 3:0:0:3 sdd 8:48 active ready running
| |- 3:0:1:3 sdh 8:112 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
 |- 3:0:2:3 sdl 8:176 active ready running
 |- 3:0:3:3 sdp 8:240 active ready running
```

3.4.4. Partitioning Device Mapper Shared Disks

Create a partition for each device mapper volume (*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
```

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

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*.

3.4.5. Configuring Oracle ASM Disks

The configuration of Oracle ASM requires the use of either *udev* rules, *Oracle ASMLib* or *Oracle ASM Filter Driver*.

The following table provides key considerations between *udev* rules, *Oracle ASMLib* and *Oracle ASM Filter Driver* (ASMFD).

Table 3.4. Oracle ASM Key Considerations

Technology	Pros	Cons
udev rules	No proprietary user space utilities; native to OS; standard device manager on Linux distributions; same performance as *Oracle ASMLib* and ASMFD	Cannot stop an accidential I/O write done by a program or user error
Oracle ASMLib	No pros as it is slowly being deprecated in favor of ASMFD	Requires additional kernel module on OS; disks managed by Oracle instead of native OS; errors loading *Oracle ASMLib* may cause losing access of Oracle ASM disks until module can be reloaded; no performance benefit over native *udev* rules

Oracle ASM Filter Driver	Filters out all non-Oracle I/Os that may cause accidental overwrites to managed disks	Requires additional kernel module on OS; disks managed by Oracle instead of native OS; errors loading ASMFD may cause losing access of Oracle ASM disks until module can be reloaded; no performance benefit over native *udev* rules

This reference architecture takes advantage of Red Hat's native device manager *udev* rules as the method of choice for configuring Oracle ASM disks. For more information on Oracle ASM Filter Driver and installation method, visit: Administering Oracle ASM Filter Driver

3.4.5.1. Oracle ASMLib and Oracle ASM Filter Driver Alternative: Configuring *udev* Rules

This section focuses on the best practices of using Red Hat's native *udev* rules to setup the appropriate permissions for each device mapper disk.

1. 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 *db1p1*,*db2p1*,*fra1*,*redo1*.

```
# for i in db1p1 db2p1 fra1 redo1; do printf "%s %s\n" "$i"
"$(udevadm \
> info --query=all --name=/dev/mapper/$i | grep -i dm_uuid)"; done

db1p1 E: DM_UUID=part1-mpath-3600c0ff000d7e7a899d8515101000000
db2p1 E: DM_UUID=part1-mpath-3600c0ff000dabfe5a7d8515101000000
fra1 E: DM_UUID=part1-mpath-3600c0ff000d7e7a8dbd8515101000000
redo1 E: DM_UUID=part1-mpath-3600c0ff000dabfe5f4d8515101000000
```

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:

```
KERNEL=="dm-*",ENV{DM_UUID}=="part1-mpath-
3600c0ff000dabfe5f4d8515101000000",OWNER="grid",GROUP="asmadmin",MOD
E="06
60"
```

   To understand the rule above, it can be read as follows: If any **dm-** device (dm-*) matches the *DM_UUID* of **part1-mpath- 3600c0ff000dabfe5f4d8515101000000**, assign that **dm-** device to be owned by the **grid** user and part of the **asmadmin** group with the permission mode set to 0660.

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

5. 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:

```
# for i in db1p1 db2p1 fra1 redo1; 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
fra1 lrwxrwxrwx. 1 root root 8 May 20 20:39 /dev/mapper/fra1 ->
../dm-13
redo1 lrwxrwxrwx. 1 root root 8 May 20 20:39 /dev/mapper/redo1 ->
../dm-14
```

6. Apply and test the rules created within the *99-oracle-asmdevices.rules* by running a **udevadm test** on each device.

```
# 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
```

7. Confirm the device has the desired permissions

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

NOTE

For simplicity, this *99-oracle-asmdevices.rules* file is included in Appendix G, *99-oracle-asmdevices.rules*

3.4.6. 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.

In order to simplify the tuning process for Oracle databases, the creation of a custom oracle profile labeled **tuned-profiles-oracle** resides in the **rhel-7-server-optional-rpms** repository. The **tuned-profiles-oracle** profile uses the throughput performance profile as its foundation and additionally sets all the different parameters mentioned in previous sections of this reference architecture and disables Transparent HugePages (THP) for Oracle databases workload environments.

For more information on why THP is disabled, see Section 4.5, "Enabling HugePages". Table 3.5, "Profile Tuned Profile Comparsion" provides details between the balanced profile, throughput-performance profile, and the custom profile tuned-profiles-oracle.

Table 3.5. Profile Tuned Profile Comparsion

Tuned Parameters	balanced	throughput-performance	tuned-profiles-oracle
I/O Elevator	deadline	deadline	deadline
CPU governor	OnDemand	performance	performance
kernel.sched_min_granularity_ns	auto-scaling	10ms	10ms
kernel.sched_wake_up_granularity_ns	3ms	15ms	15ms
disk read-ahead	128 KB	4096 KB	4096 KB
vm.dirty_ratio	20%	40%	80%*
File-system barrier	on	on	on
Transparent HugePages	on	on	off
vm.dirty_background_ratio	10%	10%	3%*
vm.swappiness	60%	10%	1%*
energy_perf_bias	normal	performance	performance
min_perf_pct (intel_pstate_only)	auto-scaling	auto-scaling	auto-scaling
tcp_rmem_default	auto-scaling	auto-scaling	262144*
tcp_wmem_default	auto-scaling	auto-scaling	262144*
udp_mem(pages)	auto-scaling	auto-scaling	auto-scaling
vm.dirty_expire_centisecs	-	-	500*
vm.dirty_writeback_centisecs	-	-	100*
kernel.shmmax	-	-	439804651110417*
kernel.shmall	-	-	107374182417*

kernel.sem	-	-	250 32000 1000 128*
fs.file-max	-	-	681574417*
fs.aio-max-nr	-	-	104857617*
ip_local_port_range	-	-	9000 65500*
tcp_rmem_max	-	-	4194304*
tcp_wmem_max	-	-	104857617*
kernel.panic_on_oops	-	-	1*

- The values expressed within the **tuned-profiles-oracle** are subject to change. The values found within the **tuned-profiles-oracle** are meant to be used as starting points and may require changes for the specific environment being tuned for the optimal performance of the Oracle Database environment.

The following procedures provide the steps that are required to install, enable, and select the **tuned-profiles-oracle** profile.

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. Ensure that the **rhel-7-server-optional-rpms** repository is available, otherwise enable via:

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

5. Install the **tuned-profiles-oracle** package

```
# yum install tuned-profiles-oracle
```

6. Activate the **tuned-profiles-oracle** profile

```
# tuned-adm profile oracle
```

7. Verify that THP is now disable via:

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

8. Disable transparent huge pages persistently across reboots by adding **`transparent_hugepage=never`** 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"
```

9. For the grub changes to take effect, run the following:

```
# 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
```

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:

```
# 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.

3.4.6.1. Customizing the `tuned-profiles-oracle` profile

The purpose of the **`tuned-profiles-oracle`** profile is to provide a starting baseline for an Oracle Database deployment. When further customization is required, the following section describes how to modify the profiles settings to meet custom criteria.

In order to modify the existing **tuned-profiles-oracle** profile, changes to the **tuned.conf** file within */usr/lib/tuned/oracle* is required. Due to the changes since Red Hat Enterprise Linux 7.0, the following are recommendations for changes when running Red Hat Enterprise Linux 7.1 or higher.

The following parameters are commented out due to higher values being used by Red Hat Enterprise Linux 7.1 or higher with a default installation. The list includes:

```
#kernel.shmmax = 4398046511104
#kernel.shmall = 1073741824
#kernel.shmmni = 4096
#fs.file-max = 6815744
#kernel.panic_on_oops = 1
```

Example of **tuned.conf**

```
#
# tuned configuration
#

[main]
summary=Optimize for Oracle RDBMS
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.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

[vm]
transparent_hugepages=never
```

As mentioned earlier, all these values are starting points and may require additional adjustments to meet an environment's requirements.

Restart the **tuned** service for the changes to take effect.

```
# systemctl restart tuned.service
```

CHAPTER 4. ORACLE 12C RELEASE 2 CONFIGURATION

4.1. INSTALLING ORACLE GRID INFRASTRUCTURE (REQUIRED FOR ASM)

The installation of the Oracle Grid Infrastructure for Oracle 12c Release 2 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:

- Chapter 2, *Reference Architecture Environment*
- Chapter 3, *Reference Architecture Configuration Details*

NOTE

The reference environment uses the **/u01/app/12.2.0/grid** as the Grid home. The owner is set to **grid** and the group is set to **oinstall**.

The following commands create the Grid home directory and set the appropriate permissions:

As the **root** user,

```
# mkdir --parents /u01/app/12.2.0/grid
# chown --recursive grid.oinstall /u01
```

1. Download the Oracle Grid Infrastructure software files[9] from the Oracle Software Delivery Cloud
2. Change the ownership and permissions of the downloaded file, move the fileto the Grid home and install **unzip** package for unpackaging of the file.

```
# cd <grid_download_location>
# chown grid.oinstall V840012-01.zip
# mv V840012-01.zip /u01/app/12.2.0/grid
# yum install unzip
```

3. **ssh** as the **grid** user with the -Y option, change directory into the Grid home **/u01/app/12.2.0/grid** and **unzip** the download zip file.

```
ssh -Y grid@<hostname>
$ cd /u01/app/12.2.0/grid
$ unzip -q V840012-01.zip
```

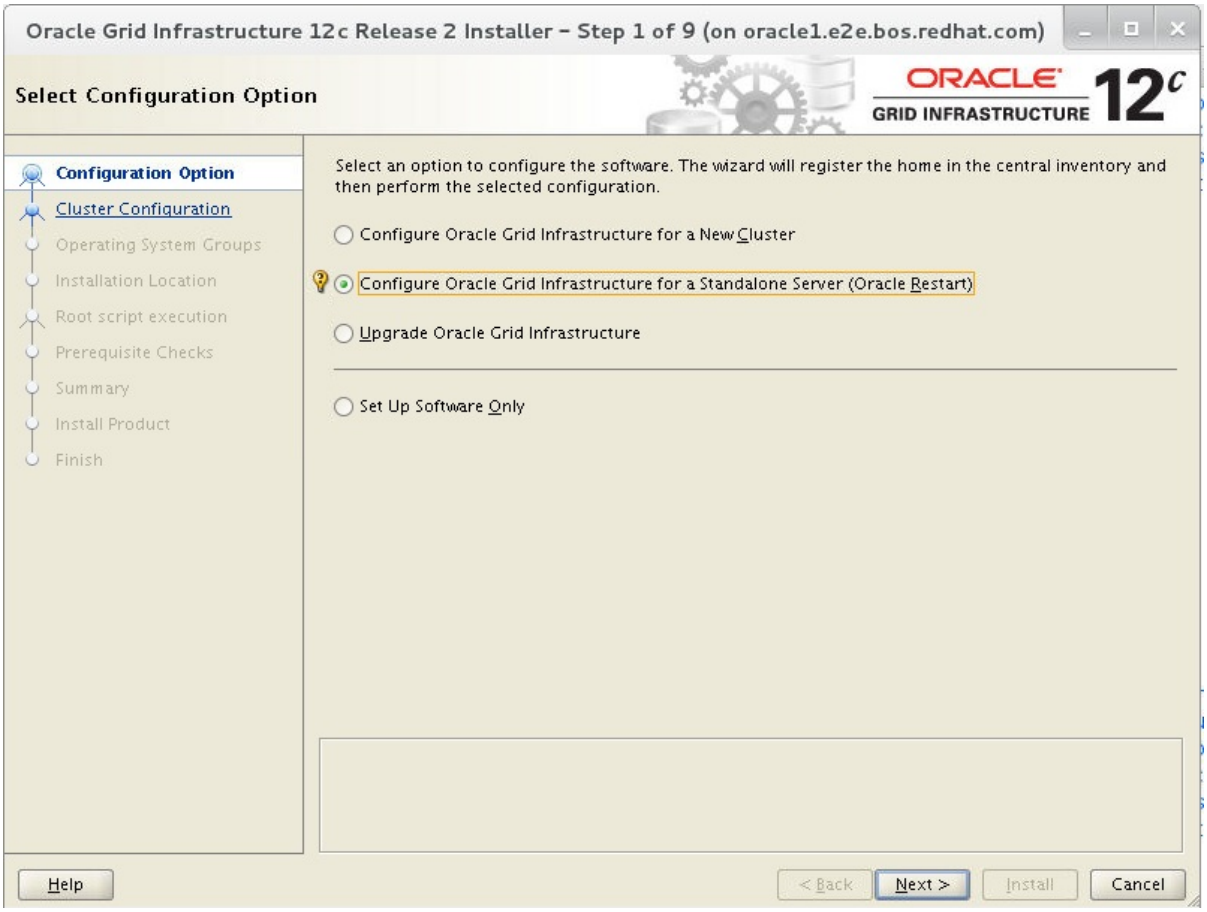
4. As the **grid** user, start the OUI via the command:

```
$ /u01/app/12.2.0/grid/gridSetup.sh
```

NOTE

Ensure to issue **ssh** with the -Y option as the **grid** user from the client server. Otherwise, a DISPLAY error may occur.

5. Within the ***Configuration Option*** window, select *Configure Oracle Grid Infrastructure for a Standalone Server (Oracle Restart)* and select *Next*.



6. Within the ***Create ASM Disk Group*** window, provide the following:

   - Disk group name, i.e. *DATA*
   - 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.
   - Disks to be assigned to the Disk group, i.e. */dev/mapper/db1p1*, */dev/mapper/db2p1*

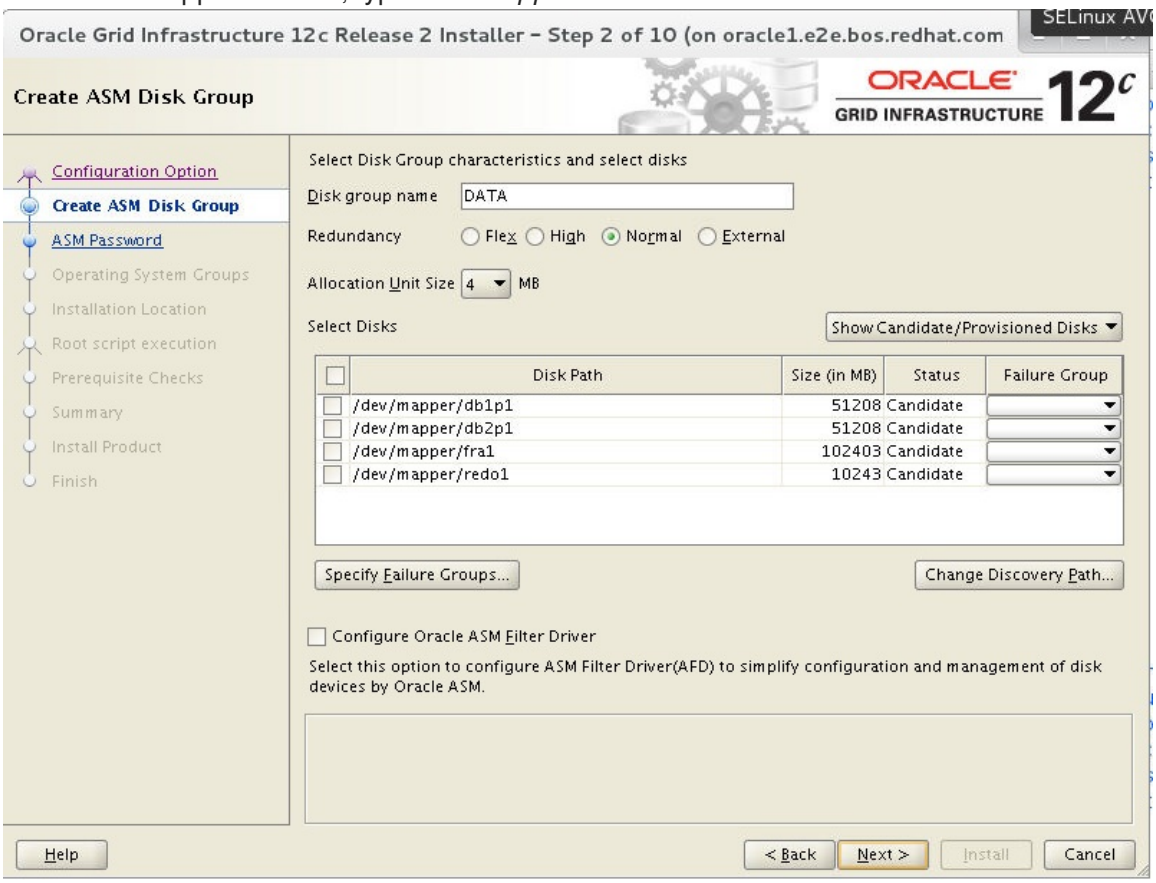
NOTE

This reference environment uses *Normal* redundancy

   - Allocation Unit (AU) Size set to 4MB
     - A 4MB AU size is used to crease the amount of extents Oracle needs to manage. With less extends 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 4 MB. Please visit Oracle's documentation[10] 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/**



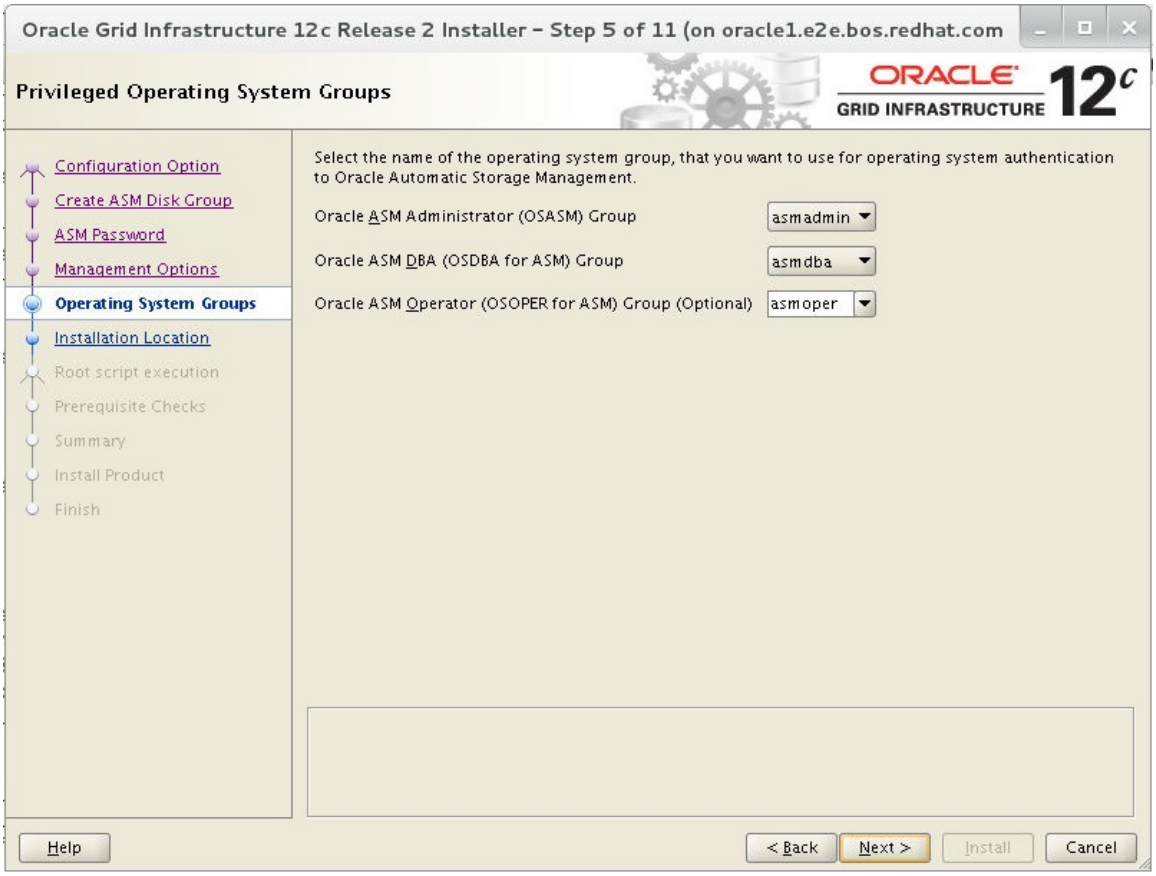
7. Click *Next* once complete within the ***Create ASM Disk Group*** window.

8. Within the ***ASM Password*** window, specify the password for the **SYS** and **ASMSNMP** user accounts, click *Next*.

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

10. Within the ***Operating System Groups*** window, select the appropriate OS groups and click *Next*. The values as created and assigned within this reference environment are as follows:

    - Oracle ASM Administrator Group – *ASMADMIN*
    - Oracle ASM DBA Group – *ASMDBA*
    - Oracle ASM Operator Group – *ASMOPER*



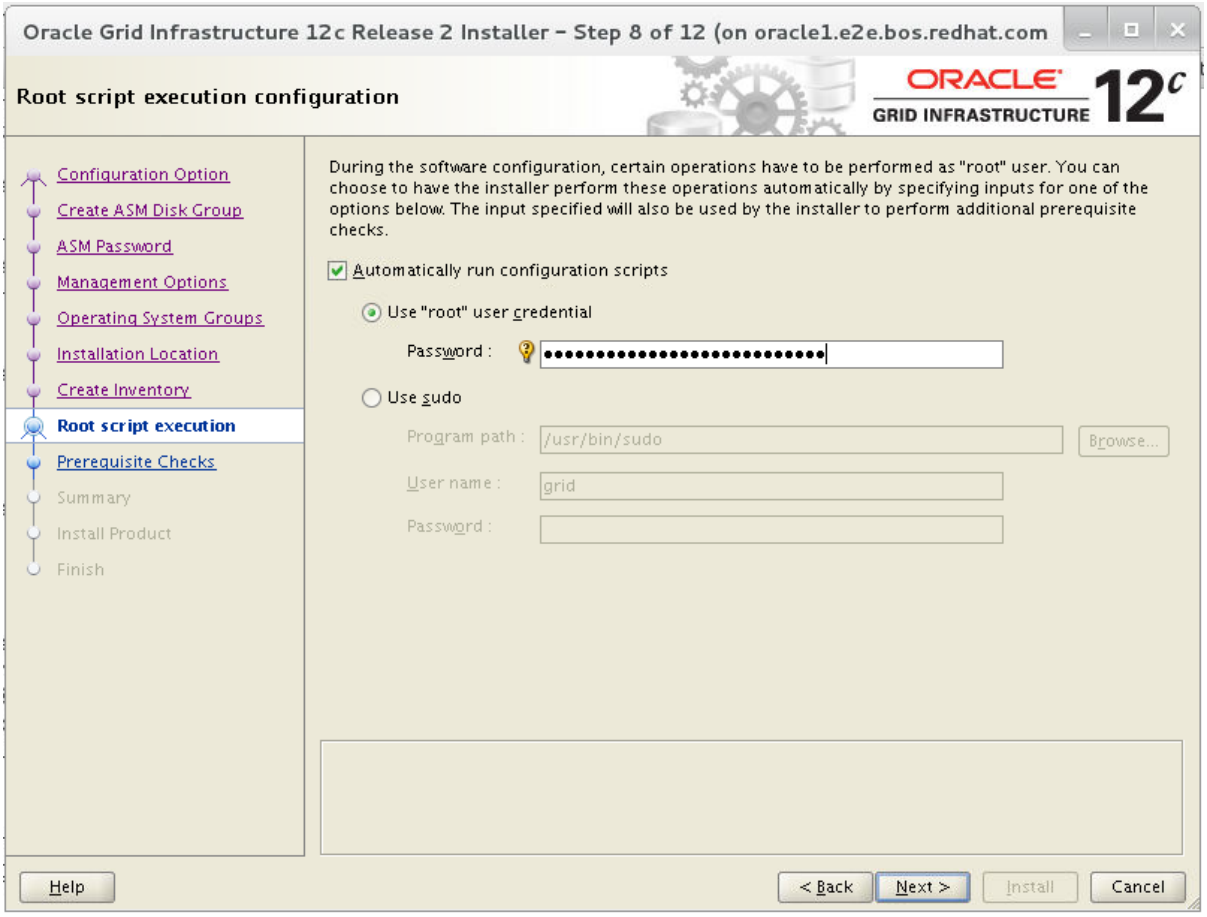
11. Within the ***Installation Location*** window, specify the appropriate Oracle base and software locations and click *Next*. The values set by this reference environment are as follows:

    - Oracle base: **`/u01/app/12.2.0`**
    - Software location: **`/u01/app/12.2.0/grid`**

12. Within the ***Create Inventory*** window, specify the inventory directory and click *Next*. 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. Click *Next*.



14. Within the ***Prerequiste 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 **`fixup.sh`** script automatically. Otherwise, as **`root`**, run the **`/tmp/gGridSetupActions_<timestamp>/CVU_<grid_version>_grid/runfixup.sh`** and click on the ***Check Again*** button once the **`runfixup.sh`** has finished.

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.

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

9: Oracle Database 12c Release 2 - V840012-01.zip from http://edelivery.oracle.com

10: Oracle ASM Extents - https://docs.oracle.com/database/121/OSTMG/GUID-1E5C4FAD-087F-4598-B959-E66670804C4F.htm

4.2. INSTALLING ORACLE 12C R1 DATABASE SOFTWARE

Prior to the installation of the Oracle 12c Release 2, ensure the following prerequisites from the following sections have been met:

- Chapter 2, *Reference Architecture Environment*

- Chapter 3, *Reference Architecture Configuration Details*
- Section 4.1, “Installing Oracle Grid Infrastructure (Required for ASM)”

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**.

The following commands create the Oracle base directory and set the appropriate permissions:

As the **root** user,

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

1. Download the Oracle Database software files[9] from the Oracle Software Delivery Cloud
2. Change the ownership and permissions of the downloaded file, move the file to the Oracle home and install **unzip** package for unpackaging of the file.

   ```
   # cd <oracle_download_location>
   # chown oracle.oinstall V839960-01.zip
   # mv V839960-01.zip /u01/app/oracle-software
   ```

3. **ssh** as the **oracle** user, change directory into the **/u01/app/oracle-software** and **unzip** the download zip file.

   ```
   ssh -Y oracle@<hostname>
   $ cd /u01/app/oracle-software
   $ unzip -q V839960-01.zip
   ```

4. As the **oracle** user, start the OUI via the command:

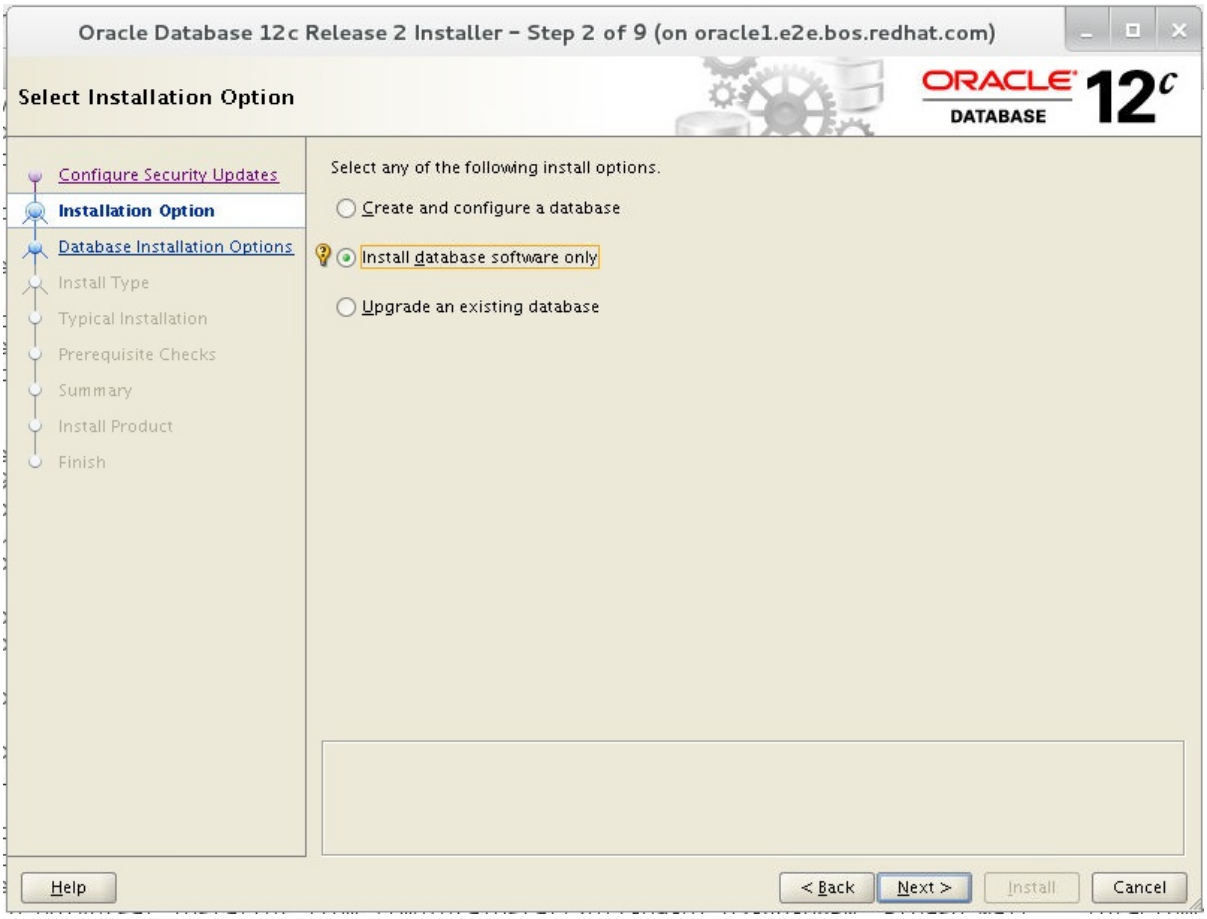
   ```
   $ /u01/app/oracle-software/database/runInstaller
   ```

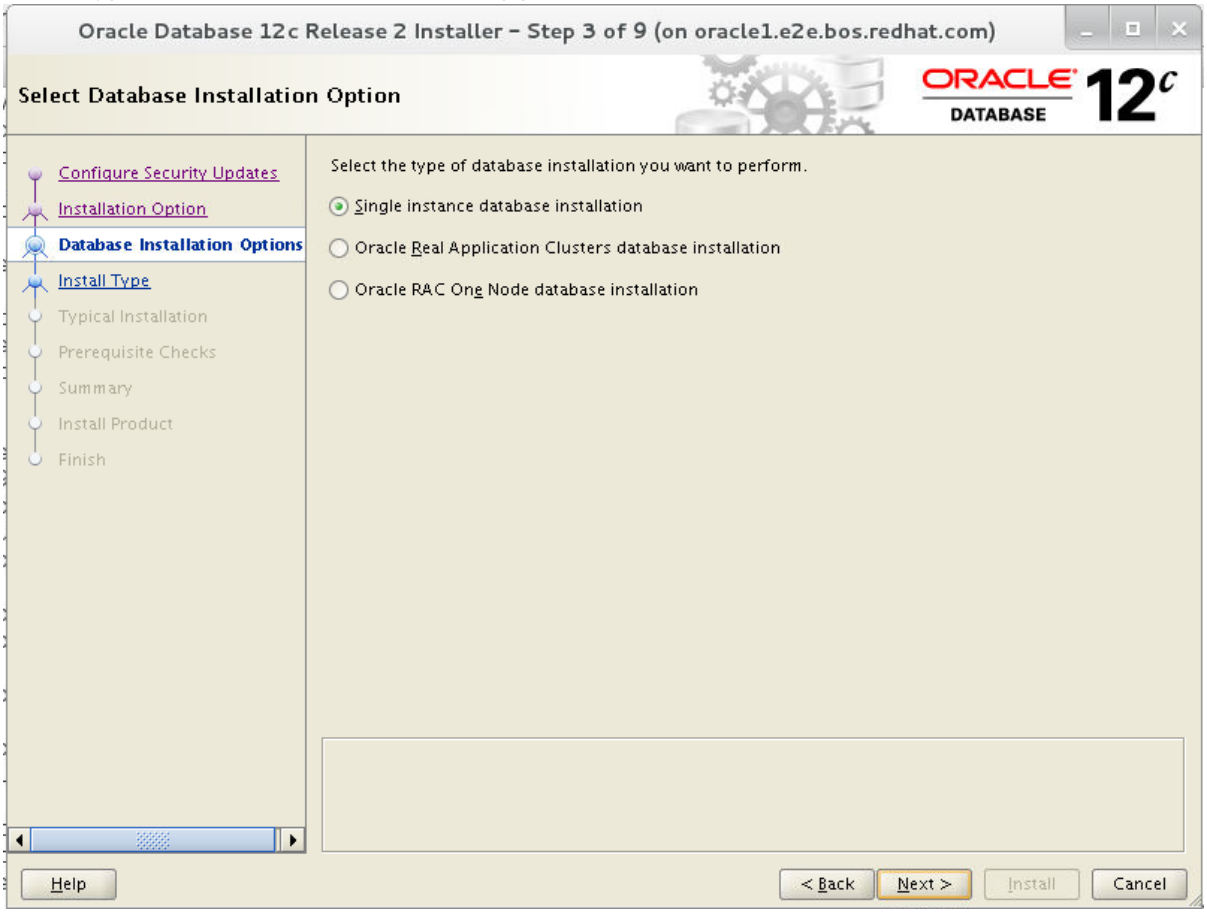
   NOTE

   Ensure to issue **ssh** with the -Y option as the **oracle** user from the client server. Otherwise, a DISPLAY error may occur.

5. 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*.
6. Within the ***Installation Option*** window, select *Install database software only* and click *Next*.



7. Within the ***Database Installation Options*** window, select *Single Instance database installation* as the type of database installation being performed and click *Next*.



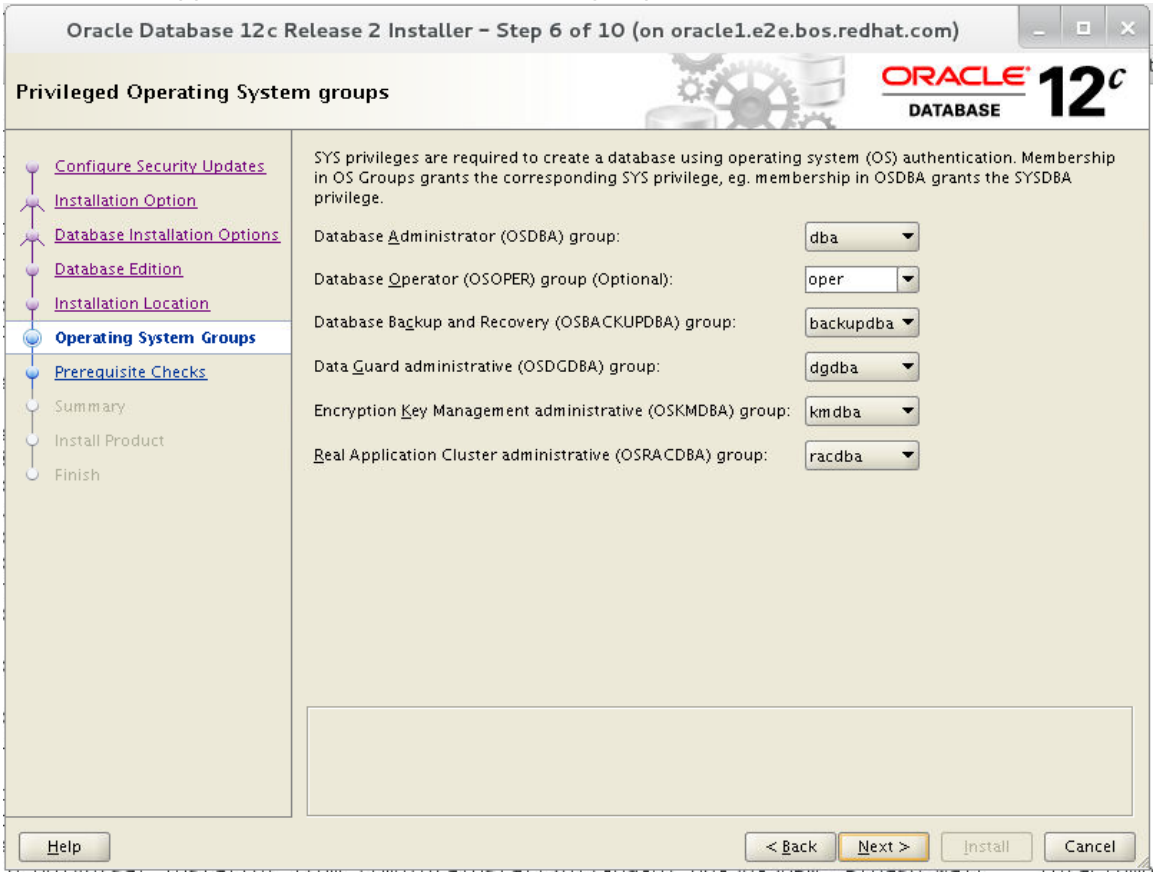
8. Within the ***Database Edition*** window, select the appropriate database edition and click *Next*. For the purposes of this reference environment, *Enterprise Edition* is the edition of choice.

9. 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/12.2.0/dbhome_1`**

10. 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:

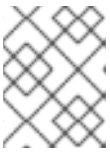
    - Database Administrator group – DBA
    - Database Operator group – OPER
    - Database Backup and Recovery group – BACKUPDBA
    - Data Guard Administrative group – DGDBA
    - Encryption Key Management Administrative group – KMDBA
    - Oracle Real Application Cluster Administration group - RACDBA



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

12. Once the installation completes, execute the scripts within the ***Execute Configuration scripts*** window. As the **`root`** user, run the following:

```
# /u01/app/oracle/product/12.2.0/dbhome_1/root.sh
```



NOTE

In the example above, */u01/app/oracle/product/12.2.0/dbhome_1* is the Oracle home directory.

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

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

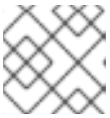
4.3. CREATING ASM DISKGROUPS VIA THE ASM CONFIGURATION ASSITANT (ASMCA)

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

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:

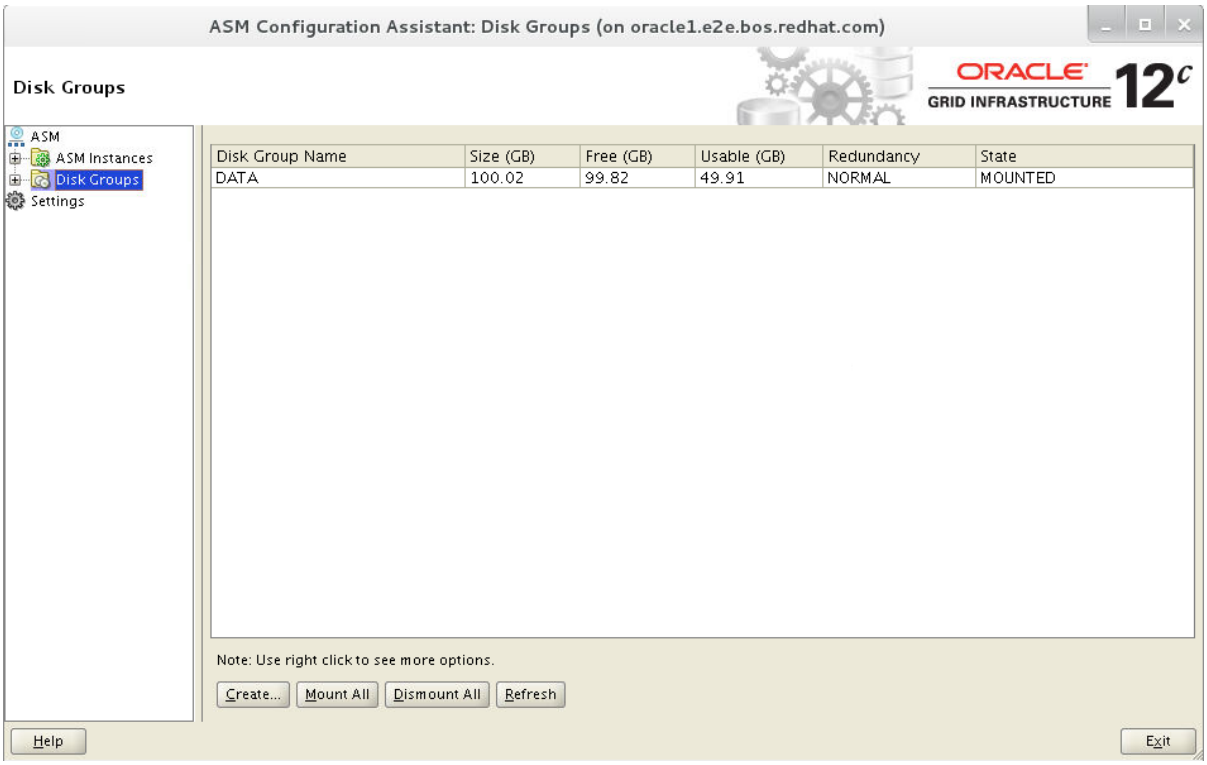
```
$ /u01/app/12.2.0/grid/bin/asmca
```



NOTE

`/u01/app/12.2.0/grid` is the Grid home directory.

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



4. Within the ***Create Disk Group*** window, provide the following:

   - A name for the disk group, i.e. *FRADG*

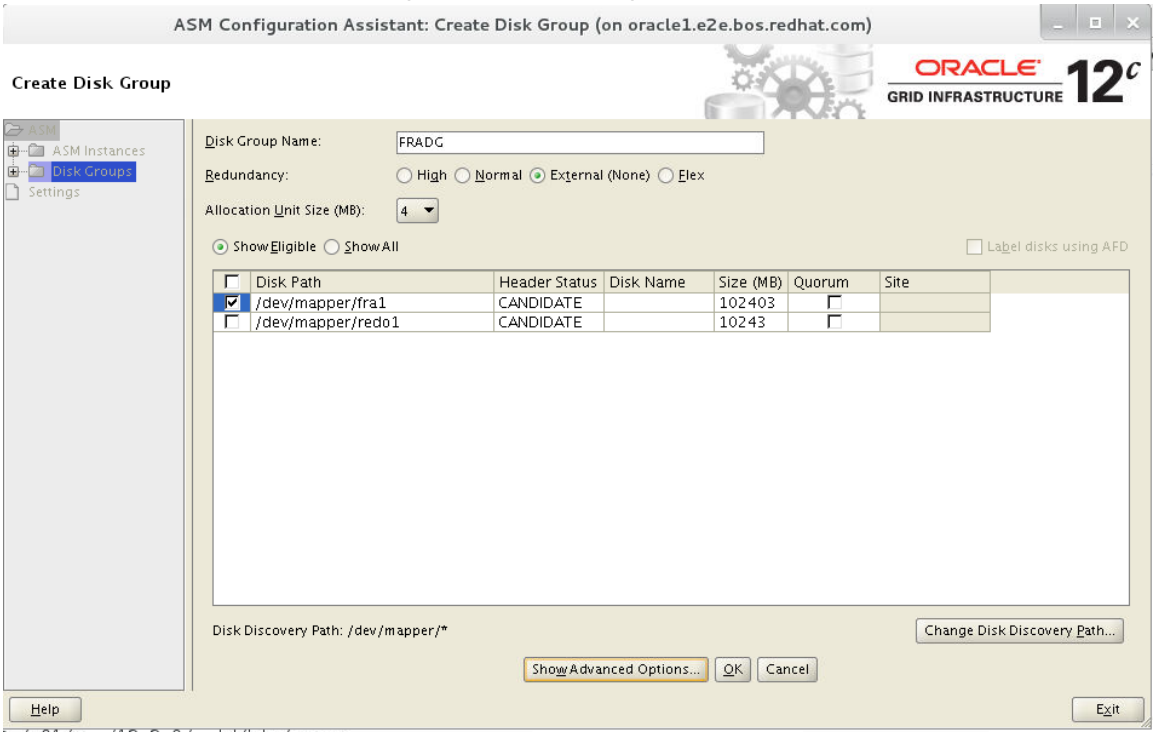
- Redundancy level for the disk group, i.e. *External Redundancy*
- Selection of the disks to be added to the disk group, i.e. */dev/mapper/fra1*
- Select an AU Size, i.e. 4 MB

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/**

1. Click the *OK* button once the steps above are complete.



2. Repeat the above steps to configure additional disk groups. It is recommended to create a diskgroup to separate the Redo logs, however, it is not required.

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

4.4. 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[11] that 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 database, the recommended method is the usage of the **dbca** utility. Prior to getting into to 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[11] – 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[11] – also known as 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, that 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[11]– 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*, that 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[11].

11: https://docs.oracle.com/database/122/ADMIN/overview-of-managing-a-multitenant-environment.htm#ADMIN13507

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.

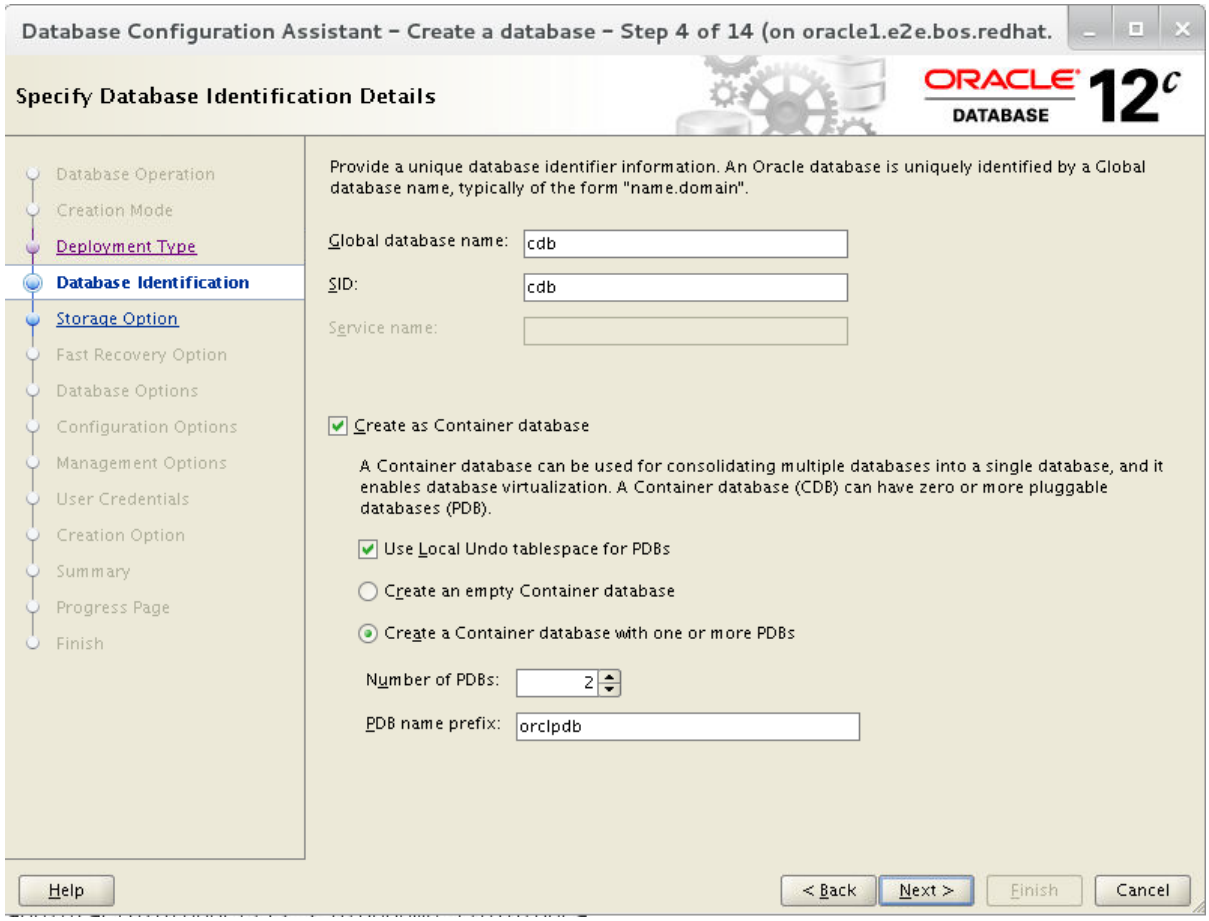
1. **ssh** with the -Y option as the **oracle** user prior to running **dbca**.
2. As the **oracle** user, run the **dbca** utility via the command:

```
$ /u01/app/oracle/product/12.2.0/dbhome_1/bin/dbca
```

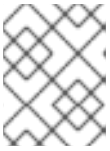
NOTE

In the example above, */u01/app/oracle/product/12.2.0/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 *Database Type* as *Oracle Single Instance database* and *Custom Database* radio button. Click *Next*.
6. Within the ***Database Identification*** window, set a global database name and Oracle System Identifier (SID), 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. *orclpdb* and click *Next*. This reference environment creates two PDBs.

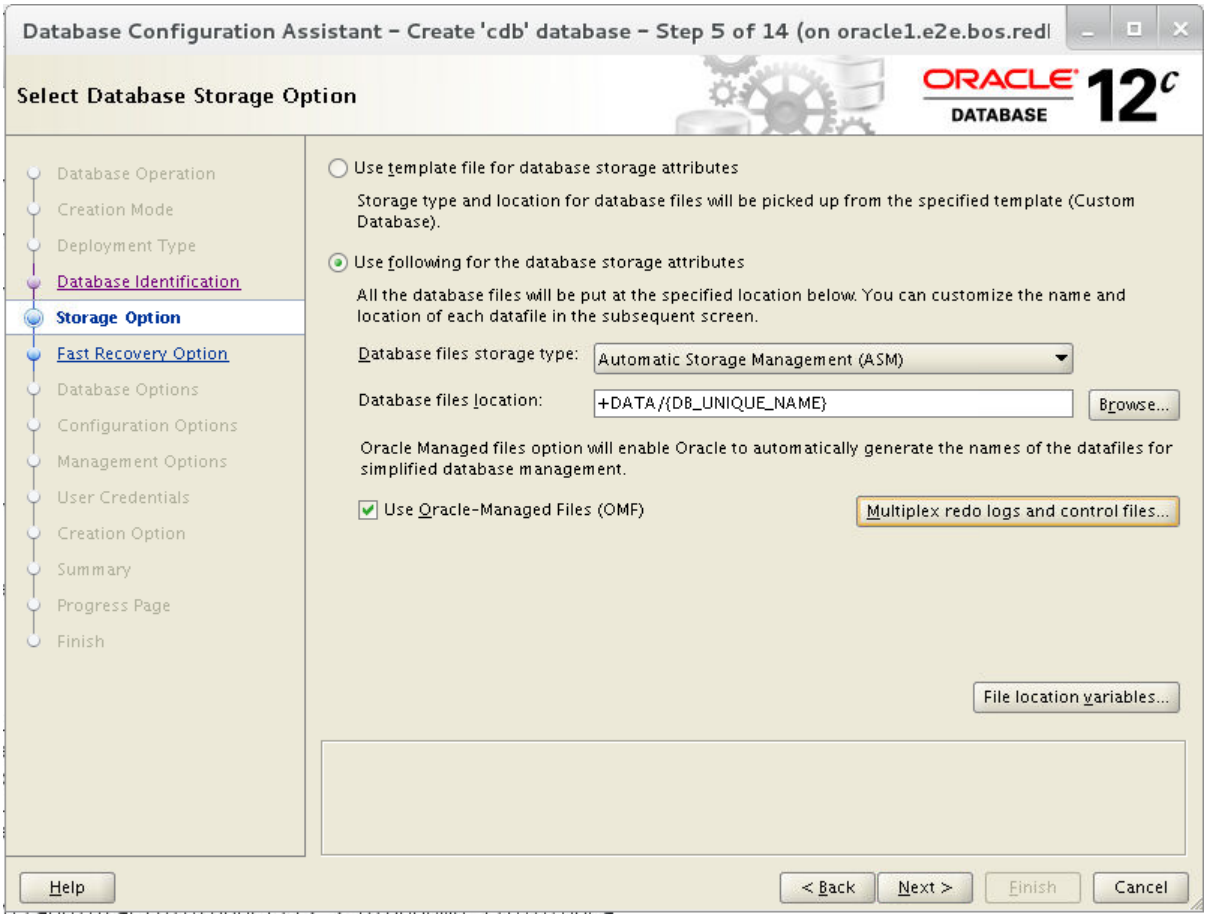


7. Within the ***Storage Option*** window, select *Use following for the database storage attributes* radio button. Change the *Database file storage type:* to *Automatic Storage Management (ASM). Within the _Database file location*: select the *Browse* button and pick the database disk group, i.e. **+DATA**. Select the *Mutliplex redo logs and control files* and enter the name of the redo log disk group (if created previously), i.e. **+REDODG**.

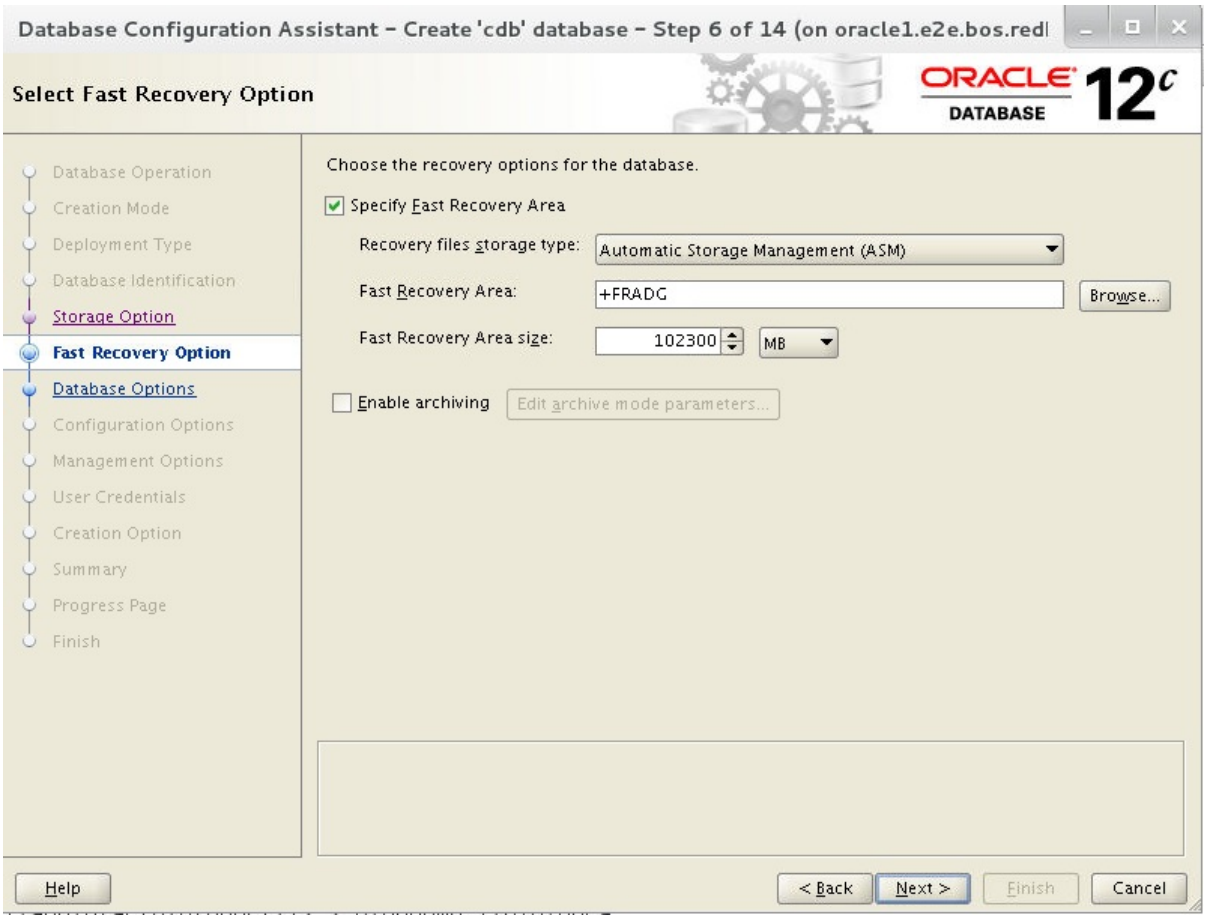


NOTE

The use of Oracle-Managed Files (OMF) is used within the reference environment, however, it is not required.



8. Within the ***Fast Recovery Option*** window, check the checkbox labeled *Specify Fast Recovery Area*, and select the *Browse* button to pick the diskgroup that is to be assigned for Fast Recovery Area, i.e. **+FRADG**. Enter an appropriate size based upon the size of the disk group.



9. Within the ***Network Configuration*** window, ensure the **`LISTENER`** is checked and click *Next*.

10. Within the ***Database Options*** window, select the database components to install. This reference environment kept the defaults. Once selected, click *Next*.

11. Within the ***Configuration Options*** window, ensure the *Use Automatic Shared Memory Segment* is selected, and use the scroll bar or enter the appropriate SGA and PGA values for the environment. The remaining tabs, *Sizing, Character sets, Connection mode*, the defaults are used.

12. Within the ***Management Options*** window, modify the Enterprise Manager database port or deselect *Configure Enterprise (EM) database express* if not being used. This reference architecture uses the defaults and selected *Next*.

13. Within the ***User Credentials*** window, enter the credentials for the different administrative users and click *Next*.

14. Within the ***Creation Option*** window, ensure the *Create database* checkbox is selected. This refernece architecture uses the defaults for all other options, but may be customizable to fit an environment's requirements.

15. Within the ***Summary*** window, review the Summary, and click *Finish*. to start the database creation.

4.5. 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 may 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 [12] 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 C, *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:

1. Copy the bash script found within Appendix C, *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:

```
# chmod +x huge_pages_settings.sh
```

3. As the **`root`** user, ensure the **`bc`** package is installed

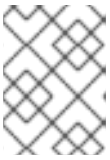
```
# yum install bc
```

4. As the **`root`** user, execute the *huge_pages_settings.sh* script as follows:

```
# /path/to/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>
```

5. Add the number of hugepages provided by the script to the kernel boot command line within the */etc/default/grub* as follows:

```
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="nofb splash=quiet crashkernel=auto
rd.lvm.lv=myvg/root rd.lvm.lv=myvg/swap rd.lvm.lv=myvg/usr 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.[13]

6. For the grub changes to take effect, run the command:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-693.1.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-693.1.1.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-514.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-514.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-
f9650ab62cd449b8b2a02d39ac73881e
Found initrd image: /boot/initramfs-0-rescue-
f9650ab62cd449b8b2a02d39ac73881e.img
done
```

7. 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>
```

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 node(s) 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 command:

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

12: ALERT: Disable Transparent HugePages on SLES11,RHEL6,OEL6 and UEK2 Kernels (DOC ID: 1557478.1)

13: https://www.kernel.org/doc/Documentation/vm/hugetlbpage.txt

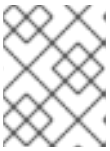
CHAPTER 5. LOGGING INTO THE ORACLE CONTAINER DATABASE 12C RELEASE 2

This section focuses on ensuring once the Oracle 12c Release 2 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.

As the **oracle** user,

1. Set the environment variable for *ORACLE_HOME* with the location of the Oracle home. This reference environment sets *ORACLE_HOME* to */u01/app/oracle/product/12.2.0/dbhome_1*

```
$ export ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1
$ echo $ORACLE_HOME
/u01/app/oracle/product/12.2.0/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.

```
$ export ORACLE_SID=cdb
$ echo ORACLE_SID
cdb
```

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

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

SQL*Plus: Release 12.2.0.1.0 Production on Tue Sep 5 18:56:49 2017

Copyright (c) 1982, 2016, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit
Production
```

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

```
SQL> show parameter use_large_pages;

NAME                                 TYPE    VALUE
------------------------------------ ----------- ---------------
--------------
use_large_pages              string  TRUE
```

WARNING

The following step requires that there is enough physical RAM on the system to place the entire SGA in large pages. If there is not enough RAM, the Oracle database instance won't start. If there is not enough RAM on the system to place the entire SGA into large pages, leave the default setting and ignore the remaining steps within this section.

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

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

6. Shutdown the Oracle database instance and restart the Oracle database instance.

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


SQL> startup;
ORACLE instance started.

Total System Global Area 3.1944E+10 bytes
Fixed Size         30045256 bytes
Variable Size        5637147576 bytes
Database Buffers     2.6240E+10 bytes
Redo Buffers           37060608 bytes
Database mounted.
Database opened.
```

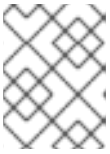
7. Verify the current value of the Oracle paramter **use_large_pages** is now set to **only**.

```
SQL> show parameter use_large_pages;

NAME                                 TYPE    VALUE
------------------------------------ ----------- ---------------
--------------
use_large_pages                      string  ONLY
```

8. 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>/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.

```
************************************************************************
**
2017-09-05T19:09:49.134349+00:00
Dump of system resources acquired for SHARED GLOBAL AREA (SGA)
```

```
2017-09-05T19:09:49.134648+00:00
 Per process system memlock (soft) limit = 30G
2017-09-05T19:09:49.134810+00:00
 Expected per process system memlock (soft) limit to lock
 SHARED GLOBAL AREA (SGA) into memory: 30G
2017-09-05T19:09:49.135130+00:00
 Available system pagesizes:
  4K, 2048K
2017-09-05T19:09:49.135431+00:00
 Supported system pagesize(s):
2017-09-05T19:09:49.135585+00:00
  PAGESIZE  AVAILABLE_PAGES  EXPECTED_PAGES  ALLOCATED_PAGES
ERROR(s)
2017-09-05T19:09:49.135905+00:00
     2048K            15235           15234           15234
NONE
2017-09-05T19:09:49.136094+00:00
 Reason for not supporting certain system pagesizes:
2017-09-05T19:09:49.136253+00:00
  4K - Large pagesizes only
2017-09-05T19:09:49.136405+00:00
**********************************************************************
**
```



NOTE

This reference environment's SGA size is set to 30 GB, however, this value varies depending on the value provided when creating an Oracle database using **dbca**.

CHAPTER 6. 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

6.1. CONNECT TO A CDB

As the **`oracle`** user:

1. Set the environment variable for *ORACLE_HOME* with the location of the Oracle home. This reference environment sets *ORACLE_HOME* to */u01/app/oracle/product/12.2.0/dbhome_1*

```
$ export ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1
$ echo $ORACLE_HOME
/u01/app/oracle/product/12.2.0/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.

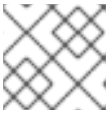
```
$ export ORACLE_SID=cdb
$ echo ORACLE_SID
cdb
```

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

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

SQL*Plus: Release 12.2.0.1.0 Production on Tue Sep 5 18:56:49 2017

Copyright (c) 1982, 2016, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit
Production
```

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. List all the available services and PDBs within the CDB:

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

NAME                     CON_ID
-----------------------------------------------
orclpdb2                 4
SYS$BACKGROUND           1
SYS$USERS                1
cdbXDB                   1
orclpdb1                 3
cdb                      1

6 rows selected.
```

6.2. 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.

As the **`oracle`** user:

Without an entry to the tnsnames.ora file, the syntax to connect to a PDB named *orclpdb1* is as follows:

```
$ $ORACLE_HOME/bin/sqlplus sys/<password>@localhost:1521/orclpdb1 as
sysdba;
SQL*Plus: Release 12.2.0.1.0 Production on Tue Sep 5 20:20:29 2017

Copyright (c) 1982, 2016, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit
Production
```

NOTE

The value 1521, represents the Oracle Listener port.

With an entry to the *tnsnames.ora* file, the syntax to connect to a PDB named *orclpdb1* 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:

$ORACLE_BASE/product/12.2.0/dbhome_1/network/admin/tnsnames.ora

```
PDB1 =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = TCP)(HOST = <FQDN_hostname>)(PORT = 1521))
 (CONNECT_DATA =
 (SERVER = DEDICATED)
 (SERVICE_NAME = orclpdb1)
 )
 )
```

6.3. MANAGING A CDB

The process of starting and shutting down a CDB database 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.

As the **`oracle`** user:

1. Connect to the CDB database as a *SYSDBA* using **`sqlplus`**. The steps are the same as shown in <connect_cdb>> steps one through three.
2. Once connected, verify the instance is the root container *CDB$ROOT*:

```
SQL> show con_name;

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

3. Shutdown the Oracle CDB database:

```
SQL> shutdown immediate;
```

4. Start the Oracle CDB database:

```
SQL> startup;
```

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

6.4. 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.

As the **oracle** user:

1. Verify the open_mode status of all the PDBs, while logged in as a *SYSDBA* in the CDB, use the following command

```
NAME                           OPEN_MODE
-------------------------------------
PDB$SEED                       READ ONLY
ORCLPDB1                       MOUNTED
ORCLPDB2                       MOUNTED
```

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 orclpdb1 open;

Pluggable database altered.

SQL> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
-------------------------------------
PDB$SEED                       READ ONLY
ORCLPDB1                       READ WRITE
ORCLPDB2                       MOUNTED
```

3. 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;

NAME                           OPEN_MODE
-------------------------------------
PDB$SEED                       READ ONLY
ORCLPDB1                       READ WRITE
ORCLPDB2                       READ WRITE
```

4. To drop a particular PDB i.e. *orclpdb2*, and its data files, execute the following SQL syntax while logged in as a *SYSDBA* in the CDB:

```
SQL> alter pluggable database orclpdb2 close immediate;

Pluggable database altered.

SQL> drop pluggable database orclpdb2 including datafiles;

Pluggable database dropped.
```

5. To verify if the pluggable database with the name *orclpdb2* has been dropped:

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

NAME                           OPEN_MODE
-------------------------------------
PDB$SEED                       READ ONLY
ORCLPDB1                       READ WRITE
```

6.5. LOCATION OF DATA FILES OF PDB & 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 6.1, “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> 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> select file_name, con_id from cdb_data_files where con_id=1;

FILE_NAME                                                  CON_ID
---------------------------------------------------------- ----------
-
+DATA/CDB/DATAFILE/system.270.836232071                      1
+DATA/CDB/DATAFILE/sysaux.273.836232077                      1
+DATA/CDB/DATAFILE/undotbs1.262.836232081                    1
+DATA/CDB/DATAFILE/users.275.836232097                       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> select file_name, con_id from cdb_temp_files where con_id =1 ;
FILE_NAME                                                  CON_ID
---------------------------------------------------------- ----------
```

```
-
+DATA/CDB/TEMPFILE/temp.278.836232081                    1
```

CHAPTER 7. CONCLUSION

Red Hat Enterprise Linux 7 provides an excellent foundation for database deployments with demonstrated stability, scalability, and performance. With the support for Oracle 12c Release 2 on Red Hat Enterprise Linux 7, customers can increasingly look to deploy Oracle 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 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 2
- Deploying Oracle Database Software 12c Release 2
- Deploying an Oracle Database 12c Release 2 using iSCSI disks
- Using Oracle ASM with *udev* rules
- Securing the Oracle Database 12c Release 2 environment with *SELinux*

For any questions or concerns, please email refarch-feedback@redhat.com and ensure to visit the Red Hat Reference Architecture page to find out about all of our Red Hat solution offerings.

APPENDIX A. CONTRIBUTORS

1. Ryan Cook (content reviewer)

APPENDIX B. 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, run either
# multipath -t
# or
# multipathd show config
#
# For a list of configuration options with descriptions, see the
multipath.conf
# man page

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

## Use user friendly names, instead of using WWIDs as names.
defaults {
    user_friendly_names yes
    find_multipaths yes
}
##
## Here is an example of how to configure some standard options.
##
#
#defaults {
#   polling_interval    10
#   path_selector       "round-robin 0"
#   path_grouping_policy    multibus
#   uid_attribute       ID_SERIAL
#   #prio           alua
#   path_checker        tur
#   rr_min_io       10
#   max_fds         8192
#   rr_weight       priorities
#   failback        immediate
#   no_path_retry       fail
#   user_friendly_names yes
#   find_multipaths     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.
##
blacklist {
       wwid 3644a84200205ec001d8d87ad30d9610b
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z]"
}
multipaths {
    multipath {
        wwid            3648171b7c4abb742729a5562a2f568ad
        alias           redo
    }
    multipath {
        wwid            3648171b7c4ab4741729a2562a2f52844
        alias           fra
    }
    multipath {
        wwid            3648171b7c4ab473d729ac561a2f558dc
        alias           db2
    }
    multipath {
        wwid            3648171b7c4ab973f729af561a2f5087c
        alias           db1
    }

#   multipath {
#       wwid            3600508b4000156d700012000000b0000
#       alias           yellow
#       path_grouping_policy    multibus
#       path_selector       "round-robin 0"
#       failback        manual
#       rr_weight       priorities
#       no_path_retry       5
#   }
#   multipath {
#       wwid            1DEC_____321816758474
#       alias           red
#   }
#}
#   device {
#       vendor          "COMPAQ  "
#       product         "HSV110 (C)COMPAQ"
#       path_grouping_policy    multibus
#       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 C. 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
```

APPENDIX D. 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
libXtst
libX11
libXau
libxcb
libXi
make
sysstat
libXmu
libXt
libXv
libXxf86dga
libdmx
libXxf86misc
libXxf86vm
xorg-x11-utils
xorg-x11-xauth
nfs-utils
smartmontools
```

APPENDIX E. KERNEL PARAMETERS (98-ORACLE-KERNEL.CONF)

```
vm.swappiness = 1
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
kernel.sem = 250 32000 100 128
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

#NOTE: These are just starting values and are not a "one-size fits all"
solution.
```

NOTE

The following parameters have been removed as the default value is equal or higher than the suggested recommendation by Oracle. Included in the list are:

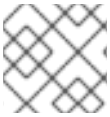
- **kernel.shmmax**
- **kernel.shmall**
- **kernel.shmmni**
- **fs.file-max**
- **kernel.panic_on_oops**

APPENDIX F. LIMITS CONFIGURATION FILE (99-GRID-ORACLE-LIMITS.CONF)

```
oracle soft nproc 2047
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 2047
grid hard nproc 16384
grid soft nofile 1024
grid hard nofile 65536
grid soft stack 10240
grid hard stack 32768
```

APPENDIX G. 99-ORACLE-ASMDEVICES.RULES

```
KERNEL=="dm-*",ENV{DM_UUID}=="
<UUID>",OWNER="grid",GROUP="asmadmin",MODE="0660"
```



NOTE

Ensure to include the following line for reach **dm-** device.

APPENDIX H. 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
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 I. CONFIGURATION FILES

All configuration files can be downloaded from GitHub. The GitHub URL is:
https://github.com/RHsyseng/oracle/tree/oracle-12.2-single-instance

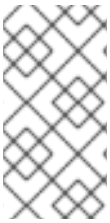
In order to access the GitHub files directly on the environment, the following steps are required:

As **root** user,

```
# cd /
# yum install -y git
# git clone https://github.com/RHsyseng/oracle.git
# cd /root/oracle
# git checkout oracle-12.2-single-instance
Branch oracle-12.2-single-instance set up to track remote branch oracle-
12.2-single-instance from origin.
Switched to a new branch 'oracle-12.2-single-instance'

# ls
98-oracle-kernel.conf       99-oracle-asmdevices.rules  multipath.conf
README.md    sample-ks.cfg
99-grid-oracle-limits.conf  huge_pages_settings.sh      oracle-grid.sh
req-rpm.txt
```

APPENDIX J. 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 occur. 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.

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

```
$ ps -A --format pid,args | grep ora_dbrm | grep -v grep
8480 ora_dbrm_cdb

$ kill -SEGV 8480
```

2. Export the ORACLE_HOME

```
$ export ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1
```

3. Start the *ADRCI* command tool via the command:

```
$ $ORACLE_HOME/bin/adrci

ADRCI: Release 12.2.0.1.0 - Production on Wed Sep 6 16:01:02 2017

Copyright (c) 1982, 2017, 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/cdb
```

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:

```
adcri> set home diag/rdbms/cdb/cdb
```

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

```
adrci> show alert -tail 50
```

NOTE

The above step is to view the alert log and check for errors. However, the following commands simplify the process of viewing problems with the Oracle deployment.

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 occurences 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/cdb:
*************************************************************************
*****
PROBLEM_ID           PROBLEM_KEY
LAST_INCIDENT        LASTINC_TIME
-------------------- -----------------------------------------
--------------- -------------------- -------------------------
-------------
1                    ORA 7445 [semtimedop]
61553                2017-09-06 15:59:10.480000 +00:00
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
-------------------- -----------------------------------------
--------------- ---------------------------------------
61553                ORA 7445 [semtimedop]
2017-09-06 15:59:10.480000 +00:00
1 row fetched
```

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

```
adrci> show incident -mode detail -p "incident_id=61553"

ADR Home = /u01/app/oracle/diag/rdbms/cdb/cdb:
*************************************************************************
*****
```

```
**********************************************************
INCIDENT INFO RECORD 1
**********************************************************
   INCIDENT_ID                   61553
   STATUS                        ready
   CREATE_TIME                   2017-09-06 15:59:10.480000 +00:00
   PROBLEM_ID                    1
   CLOSE_TIME                    <NULL>
   FLOOD_CONTROLLED              none
   ERROR_FACILITY                ORA
   ERROR_NUMBER                  7445
   ERROR_ARG1                    semtimedop
   ERROR_ARG2                    SIGSEGV
   ERROR_ARG3                    ADDR:0xD43100001DC3
   ERROR_ARG4                    PC:0x7FEA674FABDA
   ERROR_ARG5                    unknown code
   ERROR_ARG6                    <NULL>
   ERROR_ARG7                    <NULL>
   ERROR_ARG8                    <NULL>
   ERROR_ARG9                    <NULL>
   ERROR_ARG10                   <NULL>
   ERROR_ARG11                   <NULL>
   ERROR_ARG12                   <NULL>
   SIGNALLING_COMPONENT          <NULL>
   SIGNALLING_SUBCOMPONENT       <NULL>
   SUSPECT_COMPONENT             <NULL>
   SUSPECT_SUBCOMPONENT          <NULL>
   ECID                          <NULL>
   IMPACTS                       0
   CON_UID                       1
   PROBLEM_KEY                   ORA 7445 [semtimedop]
   FIRST_INCIDENT                61553
   FIRSTINC_TIME                 2017-09-06 15:59:10.480000 +00:00
   LAST_INCIDENT                 61553
   LASTINC_TIME                  2017-09-06 15:59:10.480000 +00:00
   IMPACT1                       0
   IMPACT2                       0
   IMPACT3                       0
   IMPACT4                       0
   KEY_NAME                      ProcId
   KEY_VALUE                     14.1
   KEY_NAME                      Service
   KEY_VALUE                     SYS$BACKGROUND
   KEY_NAME                      PdbName
   KEY_VALUE                     CDB$ROOT
   KEY_NAME                      Client ProcId
   KEY_VALUE
oracle@oracle1.e2e.bos.redhat.com.8480_140644811120448
   KEY_NAME
SID/u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_61553/cdb_dbrm
_8480_i61553.trc
   KEY_VALUE                     1695.17513
   OWNER_ID                      1
   INCIDENT_FILE
/u01/app/oracle/diag/rdbms/cdb/cdb/trace/cdb_dbrm_8480.trc
   OWNER_ID                      1
```

```
   INCIDENT_FILE
/u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_61553/cdb_dbrm_84
80_i61553.trc
1 row fetched
```

NOTE

The two parameters of important here are the *PROBLEM_ID* and *INCIDENT_FILE*.

9. The incident file can be examined further via:

```
adrci> show trace
/u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_61553/cdb_dbrm_84
80_i61553.trc
Output the results to file: /tmp/utsout_46828_14046_2.ado
/bin/bash: adrci: command not found
```

10. Open the */tmp/utsout_46828_14046_2.ado* with an editor such as **vi**.

```
/u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_61553/cdb_dbrm_84
80_i61553.trc
------------------------------------------------------------
LEVEL PAYLOAD
----- ------------------------------------------------------------
------------------------------------------------------------------
----------------------
      Dump file
/u01/app/oracle/diag/rdbms/cdb/cdb/incident/incdir_61553/cdb_dbrm_84
80_i61553.trc
      Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 -
64bit Production
      Build label:    RDBMS_12.2.0.1.0_LINUX.X64_170125
      ORACLE_HOME:    /u01/app/oracle/product/12.2.0/dbhome_1
      System name:      Linux
      Node name:        oracle1.e2e.bos.redhat.com
      Release:  3.10.0-693.1.1.el7.x86_64
      Version:  #1 SMP Thu Aug 3 08:15:31 EDT 2017
      Machine:  x86_64
      Instance name: cdb
      Redo thread mounted by this instance: 1
      Oracle process number: 14
      Unix process pid: 8480, image:
oracle@oracle1.e2e.bos.redhat.com (DBRM)


      *** 2017-09-06T15:59:10.488444+00:00
      *** SESSION ID:(1695.17513) 2017-09-06T15:59:10.488494+00:00
      *** CLIENT ID:() 2017-09-06T15:59:10.488508+00:00
      *** SERVICE NAME:(SYS$BACKGROUND) 2017-09-
06T15:59:10.488521+00:00
      *** MODULE NAME:() 2017-09-06T15:59:10.488534+00:00
      *** ACTION NAME:() 2017-09-06T15:59:10.488546+00:00
      *** CLIENT DRIVER:() 2017-09-06T15:59:10.488557+00:00
      *** CONTAINER ID:(1) 2017-09-06T15:59:10.488570+00:00
```

```
[ ... 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 be then 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

adrci> ips generate package 1 in "/home/oracle"
Generated package 1 in file
/home/oracle/ORA7445se_20170906161951_COM_1.zip, mode complete
```

NOTE

Problem 1 is the *Problem_ID* found in a previous step. *Package 1* is the *package ID* captured from the **`ips create`** output command.

For more information about *ADRCI* visit: http://docs.oracle.com/database/122/SUTIL/oracle-ADR-command-interpreter-ADRCI.htm

APPENDIX K. REFERENCES

- TECH: Unix Semaphores and Shared Memory Explained (DOC ID: 15566.1)
- Oracle Grid Infrastruture, Oracle Documentation
- Tuning Red Hat Enterprise Linux For Oracle & Oracle RAC by Scott Crot, Sr.Consultant, Red Hat, Inc.)
- USE_LARGE_PAGES To Enable HugePages (ID 1392497.1)
- Large Pages Information in the Alert Log (ID 1392543.1)
- Tuning Virtual Memory - via Kernel Doc documentation (kernel-doc package)
- Maximum SHMMAX values for Linux x86 and x86-64 (ID 567506.1)
- About the Oracle Database Fault Diagnosability Infrastructure
- Oracle 12c Database Installation Guide
- Pro Oracle Database 11g RAC on Linux – Installation, Administration, Performance by Steve Shaw and Martin Bach

APPENDIX L. REVISION HISTORY

Revision 1.1-0 **2018-04-05** **Roger Lopez**

- **Updated NTP Configuration to instead use Chrony.**

Revision 1.0-0 **2017-10-01** **Roger Lopez**

- **Initial Release**