

## **Oracle Linux System Administration**

**Student Guide**

D74508GC20

Edition 2.0

February 2014

D85361

**ORACLE®**

## **Author**

Craig McBride

## **Technical Contributors and Reviewers**

Avi Miller  
Elena Zannoni  
Lenz Grimmer  
Sergio Leunissen  
Waseem Daher  
Wim Coekaerts  
Al Flournoy  
Harald Van Breederode  
Joel Goodman  
Manish Kapur  
Soeren Binner  
Jeremy Smyth  
Yasar Akthar  
Javier Saiz  
Ozgur Yuksel  
Antoinette O'Sullivan  
Frank Allan  
Gavin Bowe  
Gino Kawalski  
Jeff Suchomel  
Rob Swank  
Ron Hardin  
Michele Dady  
Matt Taylor

## **Editor**

Vijayalakshmi Narasimhan

## **Publishers**

Jobi Varghese  
Veena Narasimhan

**Copyright © 2014, Oracle and/or its affiliates. All rights reserved.**

## **Disclaimer**

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

## **Restricted Rights Notice**

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

### **U.S. GOVERNMENT RIGHTS**

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

## **Trademark Notice**

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

## Contents

### 1 Course Introduction

- Course Objectives 1-2
- Course Schedule 1-3
- Lesson Objectives 1-6
- Virtualization with Oracle VM Server 1-7
- Oracle VM Server in the Classroom 1-8
- Summary 1-10
- Practices: Overview 1-11

### 2 Introduction to Oracle Linux

- Objectives 2-2
- Linux Kernel 2-3
- The GNU Project 2-5
- GNU General Public License (GPL) 2-6
- Linux Kernel Development Model 2-8
- Continuous Mainline Kernel Development 2-10
- Linux Distributions 2-11
- Oracle Linux 2-13
- Oracle's Technical Contributions to Linux 2-14
- Oracle Linux: Compatible with Red Hat Enterprise Linux (RHEL) 2-16
- Unbreakable Enterprise Kernel 2-18
- Unbreakable Enterprise Kernel Release 1 2-19
- Unbreakable Enterprise Kernel Release 2 2-22
- Unbreakable Enterprise Kernel Release 3 2-25
- Tracking Mainline 2-27
- DTrace 2-28
- Btrfs File System 2-29
- Oracle Linux Release Notes 2-30
- Summary 2-31
- Quiz 2-32
- Practice 2: Overview 2-34

### 3 Installing Oracle Linux

- Objectives 3-2
- Obtaining Oracle Linux 3-3

Oracle Software Delivery Cloud	3-4
Anaconda Installer	3-5
Boot Menu	3-6
Boot Options	3-7
Media Test	3-8
Language Selection	3-9
Keyboard Selection	3-10
Storage Devices Selection	3-11
Setting the Host Name	3-12
Configuring Network	3-13
IPv4 Settings	3-14
Time Zone Selection	3-15
Setting Root Password	3-16
Disk Partitioning Setup	3-17
Storage Devices	3-18
Default Partition Layout	3-19
Creating a Custom Layout	3-20
Standard Partition	3-21
Confirming Partitions	3-23
Boot Loader Selection	3-24
Quiz	3-25
Software Package Selection	3-26
Customizing the Package Selection	3-28
Software Installation	3-29
FirstBoot Tool	3-30
Quiz	3-31
Summary	3-32
Practice 3: Overview	3-33

#### **4 Linux Boot Process**

Objectives	4-2
Linux Boot Process	4-3
Master Boot Record (MBR)	4-4
GRUB Bootloader	4-5
GRUB Configuration File	4-7
GRUB Menu	4-10
Editing a GRUB Menu Option	4-11
Kernel Boot Parameters	4-12
GRUB Command Line	4-14
/sbin/init Process	4-15
SysV init Run Levels	4-16

Working with Run Levels	4-17
/etc/inittab File	4-19
/etc/rc.d Directory	4-22
Stopping and Starting Services	4-24
Configuring Services	4-25
ntsysv Utility	4-27
Xinetd Service	4-28
Upstart	4-29
Summary	4-31
Quiz	4-32
Practice 4: Overview	4-35

## 5 System Configuration

Objectives	5-2
Configuring System Time	5-3
Using Network Time Protocol	5-5
/etc/sysconfig Directory	5-7
proc File System	5-9
Top-Level Files Within /proc	5-11
Process Directories in /proc	5-13
Other Directories in /proc	5-14
sysfs File System	5-16
sysctl Utility	5-18
Quiz	5-20
Summary	5-23
Practice 5: Overview	5-24

## 6 Package Management

Objectives	6-2
Introduction to Package Management	6-3
rpm Utility	6-4
Oracle Public yum Server	6-6
yum Configuration	6-9
yum Utility	6-11
yum Groups	6-13
Unbreakable Linux Network (ULN)	6-14
ULN Channels	6-15
UEK R3 Kernel Image and User Space Packages on ULN	6-17
Switching from RHN to ULN	6-18
Quiz	6-20

Summary 6-22  
Practice 6: Overview 6-23

## 7 Ksplice

Objectives 7-2  
Introduction to Ksplice 7-3  
How Ksplice Works 7-4  
Ksplice Implementation 7-5  
Ksplice Packages on ULN 7-6  
Using Ksplice Uptrack 7-7  
Ksplice Uptrack Command Summary 7-8  
System Status 7-9  
System Updated 7-10  
Ksplice Offline Client 7-11  
Modifying a Local Yum Server to Act as a Ksplice Mirror 7-12  
Updating a Local Yum Server with Ksplice Channels 7-13  
Configuring Ksplice Offline Clients to Use the Local Ksplice Mirror 7-14  
Quiz 7-15  
Summary 7-16  
Practice 7: Overview 7-17

## 8 Automating Tasks

Objectives 8-2  
Automating System Tasks 8-3  
Configuring cron Jobs 8-4  
Other cron Directories and Files 8-6  
crontab Utility 8-8  
Configuring anacron Jobs 8-9  
at and batch 8-11  
Quiz 8-13  
Summary 8-14  
Practice 8: Overview 8-15

## 9 Kernel Module Configuration

Objectives 9-2  
Loadable Kernel Modules (LKM) 9-3  
Loading and Unloading Kernel Modules 9-5  
Kernel Module Parameters 9-8  
Quiz 9-10  
Summary 9-11  
Practice 9: Overview 9-12

## **10 User and Group Administration**

- Objectives 10-2
- Introduction to Users and Groups 10-3
- User and Group Configuration Files 10-4
- Adding a User Account 10-6
- Modifying or Deleting User Accounts 10-9
- Group Account Administration 10-10
- User Private Groups 10-12
- Password Configuration 10-14
- /etc/login.defs File 10-16
- User Manager Tool 10-17
- Restricting Use of the su Command 10-18
- Allowing Use of the sudo Command 10-19
- User/Group Administration in the Enterprise 10-20
- Quiz 10-21
- Summary 10-23
- Practice 10: Overview 10-24

## **11 File Systems**

- Objectives 11-2
- Disk Partitions 11-3
- Partition Table Manipulation Utilities 11-5
- fdisk Utility 11-6
- Using the fdisk Utility 11-8
- cfdisk Utility 11-11
- parted Utility 11-12
- File System Types 11-14
- Making File Systems 11-16
- Mounting File Systems 11-18
- /etc/fstab File 11-21
- Maintaining File Systems 11-22
- Swap Space 11-24
- Quiz 11-26
- Summary 11-28
- Practice 11: Overview 11-29

## **12 Storage Administration**

- Objectives 12-2
- Logical Volume Manager (LVM) 12-3
- LVM Configuration: Example 12-4
- Physical Volume Utilities 12-5

Volume Group Utilities 12-7  
Logical Volume Utilities 12-9  
Making Logical Volumes Usable 12-11  
Backing Up and Restoring Volume Group Metadata 12-13  
Redundant Array of Independent Disks (RAID) 12-14  
mdadm Utility 12-16  
Making RAID Devices Usable 12-18  
Quiz 12-19  
Summary 12-20  
Practice 12: Overview 12-21

## **13 Network Configuration**

Objectives 13-2  
Network Interfaces 13-3  
Additional Network Configuration Files 13-5  
Command-Line Network Interface Utilities 13-7  
Address Resolution Protocol (ARP) 13-9  
Network Interface Bonding 13-11  
Virtual Local Area Networks 13-13  
route Utility 13-15  
NetworkManager 13-17  
Network Connections Window 13-20  
system-config-network Utility 13-21  
Device Configuration 13-22  
DNS Client Configuration 13-23  
Quiz 13-24  
Summary 13-25  
Practice 13: Overview 13-26

## **14 File Sharing**

Objectives 14-2  
Introduction to NFS 14-3  
NFS Server Configuration 14-6  
Starting the NFS Service 14-8  
exportfs Utility 14-9  
NFS Client Configuration 14-10  
Automounting File Systems 14-12  
Introduction to vsftpd 14-15  
vsftpd Configuration Options 14-16  
Quiz 14-18

Summary 14-19  
Practice 14: Overview 14-20

## **15 OpenSSH**

Objectives 15-2  
Introduction to OpenSSH 15-3  
OpenSSH Configuration Files 15-4  
OpenSSH Configuration 15-6  
Using OpenSSH Utilities 15-7  
Using the ssh Command 15-9  
Using the scp Command 15-10  
Using the sftp Command 15-11  
Using the ssh-keygen Command 15-12  
Using ssh-agent 15-14  
Quiz 15-15  
Summary 15-16  
Practice 15: Overview 15-17

## **16 Pluggable Authentication Modules (PAM)**

Objectives 16-2  
Introduction to PAM 16-3  
PAM Module Types 16-5  
PAM Control Flags 16-6  
PAM: Example #1 16-8  
PAM: Example #2 16-10  
Quiz 16-12  
Summary 16-13  
Practice 16: Overview 16-14  
Introduction to SELinux 16-15

## **17 Security Administration**

Objectives 17-2  
chroot Jail 17-3  
chroot Utility 17-4  
Implementing a chroot Jail 17-5  
Running Services in a chroot Jail 17-7  
Introduction to iptables 17-9  
Firewall Configuration Tool 17-10  
iptables Terminology 17-11  
Beginning iptables Maintenance 17-13  
Adding a Rule by Using the iptables Utility 17-15

- iptables Rule Specs 17-17
- More iptables Options 17-18
- NAT Table 17-19
- TCP Wrappers 17-21
- TCP Wrappers Configuration 17-22
- TCP Wrapper Command Options 17-24
- Quiz 17-26
- Summary 17-27
- Practice 17: Overview 17-28

## **18 Oracle on Oracle**

- Objectives 18-2
- Oracle Software User Accounts 18-3
- Oracle Software Group Accounts 18-4
- System Resource Tuning 18-5
- Linux Shared Memory 18-6
- Semaphores 18-7
- Network Tuning 18-9
- Setting the File Handles Parameter 18-10
- Asynchronous IO (AIO) 18-11
- Oracle-Related Shell Limits 18-12
- HugePages 18-14
- Configuring HugePages 18-16
- Oracle Database Smart Flash Cache (DBSFC) 18-18
- Oracle Pre-Install RPM 18-19
- Oracle ASM 18-21
- ASM Library Driver (ASMLib) 18-23
- Using ASMLib Commands 18-25
- Quiz 18-27
- Summary 18-28
- Practice 18: Overview 18-29

## **19 System Monitoring**

- Objectives 19-2
- sosreport Utility 19-3
- iostat Utility 19-5
- mpstat Utility 19-7
- vmstat Utility 19-9
- sar Utility 19-11
- top Utility 19-13
- iotop Utility 19-15

strace Utility 19-16  
netstat Utility 19-17  
tcpdump Utility 19-19  
Wireshark 19-21  
OSWatcher Black Box (OSWbb) 19-22  
OSWbb Diagnostic Data Output 19-24  
OSWatcher Black Box Analyzer (OSWbba) 19-28  
Analyze OSWbb Archive Files 19-31  
Enterprise Manager Ops Center 19-33  
Enterprise Manager Ops Center GUI 19-35  
Enterprise Manager Ops Center Provisioning 19-36  
Enterprise Manager Ops Center Patching 19-37  
Enterprise Manager Ops Center Monitoring 19-38  
Spacewalk 19-39  
Quiz 19-41  
Summary 19-42  
Practice 19: Overview 19-43

## **20 System Logging**

Objectives 20-2  
System Log File Configuration 20-3  
Facility/Priority-Based Filters 20-5  
rsyslog Actions 20-7  
rsyslog Templates 20-9  
Configuring Log Rotation (logrotate) 20-11  
logwatch 20-13  
Quiz 20-14  
Summary 20-15  
Practice 20: Overview 20-16

## **21 Troubleshooting**

Objectives 21-2  
Two-Phased Approach to Troubleshooting 21-3  
Gathering Information 21-4  
Operating System Logs 21-5  
dmesg Utility 21-6  
Troubleshooting Resources 21-7  
My Oracle Support 21-8  
Causes of Common Problems 21-9  
Troubleshooting Boot Problems 21-11  
Typical Causes of NFS Problems 21-12

Quiz 21-13

Summary 21-14

Practice 21: Overview 21-15

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# 1

## Course Introduction

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

## Course Objectives

After completing this course, you should be able to:

- Describe Oracle's contributions and commitment to Linux
- Install and perform initial configuration of Oracle Linux
- Describe and configure Oracle's Unbreakable Enterprise Kernel
- Configure users, storage, and network interfaces on Oracle Linux
- Describe the preparation of Oracle Linux server for installation of Oracle database
- Monitor and troubleshoot Oracle Linux



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This course is designed to give you hands-on experience in installing Oracle Linux 6.5 and performing various system administration tasks. You are introduced to Linux in general and Oracle Linux in particular, as well as Oracle's commitment and technical contributions to the Linux operating system.

You will install Oracle Linux 6.5 and perform the initial system administration tasks that are required after an installation. You will be introduced to the Unbreakable Enterprise Kernel and configure kernel modules. You will configure users, storage, and network interfaces and learn how to monitor and troubleshoot your systems to ensure successful implementation.

# Course Schedule

Session	Module
Day 1	Lesson 1: Course Introduction Lesson 2: Introduction to Oracle Linux Lesson 3: Installing Oracle Linux Lesson 4: Linux Boot Process Lesson 5: System Configuration
Day 2	Lesson 6: Package Management Lesson 7: Ksplice Lesson 8: Automating Tasks Lesson 9: Kernel Module Configuration Lesson 10: User and Group Administration

# Course Schedule

Session	Module
Day 3	Lesson 11: File Systems Lesson 12: Storage Administration Lesson 13: Network Configuration Lesson 14: File Sharing
Day 4	Lesson 15: OpenSSH Lesson 16: Pluggable Authentication Modules (PAM) Lesson 17: Security Administration Lesson 18: Oracle on Oracle

# Course Schedule

Session	Module
Day 5	Lesson 19: System Monitoring Lesson 20: System Logging Lesson 21: Troubleshooting

## Lesson Objectives

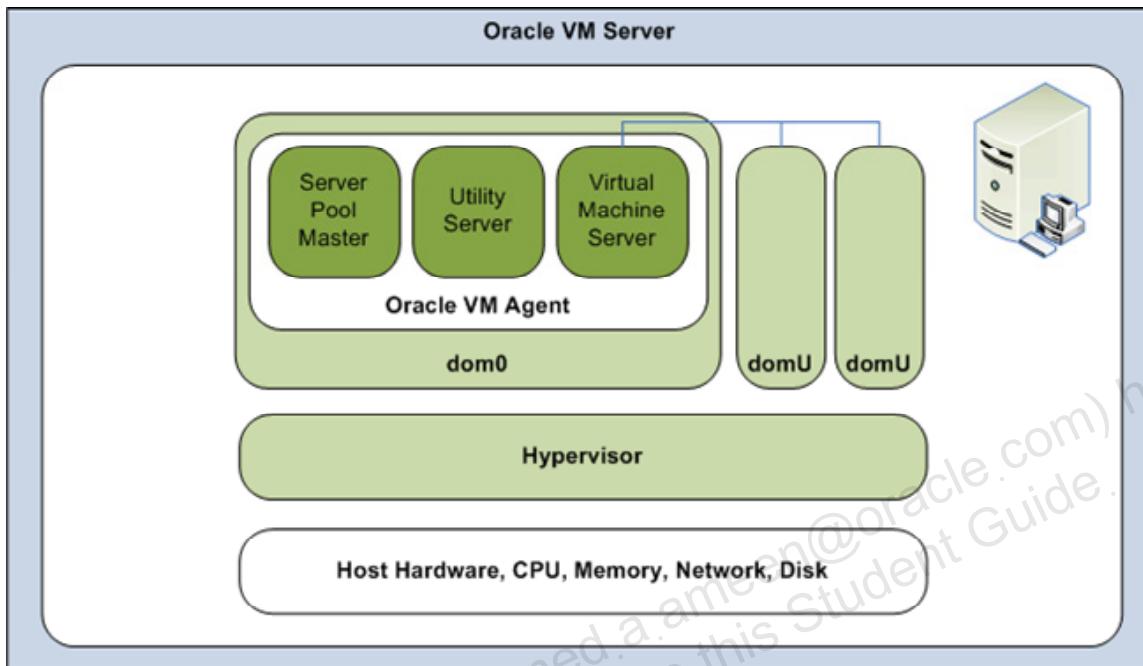
After completing this lesson, you should be able to:

- Describe the classroom environment used for the practice sessions
- Start, log in to, and stop a virtual machine on your student desktop



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Virtualization with Oracle VM Server



ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Virtualization

Virtualization allows you to use one server and its computing resources to run one or more guest operating systems and application images concurrently, sharing those resources among the guests.

## Supervisor Versus Hypervisor

The sharing of the host server's resources can be accomplished through two different approaches:

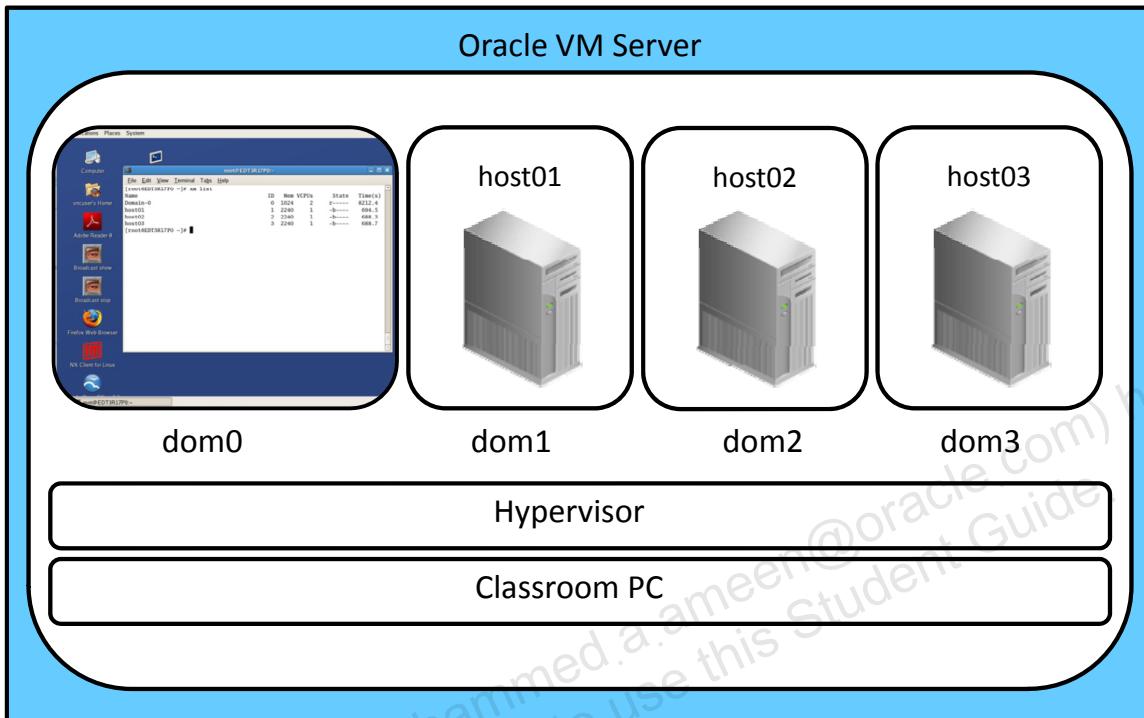
- A supervisory application such as VirtualBox or VMware that runs on the host operating system and in turn runs the guest virtual machines
- A hypervisor such as Oracle VM Server or VMware ESX that provides a small-footprint host operating system and exposes the server's resources to the guest virtual machines that run directly on top of the hypervisor

A hypervisor removes the middleman represented by the supervisory application, thereby freeing up more resources for the guest virtual machines to share.

## Oracle VM Server Domains

Oracle VM Server guests are also referred to as domains. Dom0 is always present, providing management services for the other domains running on the same server.

# Oracle VM Server in the Classroom



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Self-Contained Multi-Host Environment

Your student PC is running Oracle VM Server where you can run up to three guests as required to work through the practice sessions. Guests running on your machine can see each other, and you configure one guest to see outside the environment. Out of the box, Oracle VM Server does not offer a GUI front end; however, your dom0 has been modified to include the Gnome interface. When you log in to the machine, you are presented with a graphical interface that can also act as an X-server for your guests.

## Logging In to Your Machine

Log in as user `vncuser` (password is `vnctech`). This will log you in to dom0 and the Gnome GUI. When you are logged in, the simplest way to control your machine is from terminal sessions initiated from the Gnome desktop.

## Where to Find Your Guests

The guest VMs reside in their own directories under `/OVM/running_pool` and your practice guides will inform you which VMs are required to be running for each exercise. Note that the name of a VM does not necessarily have to be the same as the host name of the server that runs within the VM.

## Starting, Stopping, and Listing Guests

When you are logged in to dom0, you can switch to `root` (password is `oracle`) in a terminal session and use the `xm` command-line tool to manually manage guests on the machine.

- `xm list`: Lists all the currently active guests, including dom0 itself
- `xm create <config_file>`: Creates a running instance of the specified VM
- `xm shutdown -w <VM name>`: Shuts down the specified VM and waits for the action to complete before returning control to you
- `xm reset <VM name>`: Resets the specified VM. Use this command when unable to connect to the VM.

Your activity guide will detail which guests are required for each practice, and scripts are provided for ensuring that they are started. You will need to create one new guest for the Oracle Installation practice, and are expected to be proficient in using basic UNIX commands and the `vi` text editor.

## Connecting to Guests and Running GUI Utilities

The practice exercises direct you to use `vncviewer` to connect from dom0 to your guests. Use the `xm list -l VM_name | grep location` to obtain the port number required by `vncviewer`.

Alternatively, use secure shell to create a connection as the `root` user from dom0 to your guests, for example:

```
# ssh root@host01
```

The `root` password is `oracle` (lowercase) on all the guests.

## Summary

In this lesson, you should have learned how to:

- Log in to your classroom PC
- Control the guest VMs on your classroom PC
- Log in to the guest VMs on your classroom PC

## Practices: Overview

In the practices for this lesson, you:

- Log in to your classroom PC \*
- Explore dom0 configuration and directory structure
- Start, stop, and list VM guests
- Connect to a VM guest
- Log off from your classroom PC

\* See the appendix titled “Remote Access Options” for information about connecting to the classroom PC remotely.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

## Introduction to Oracle Linux

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

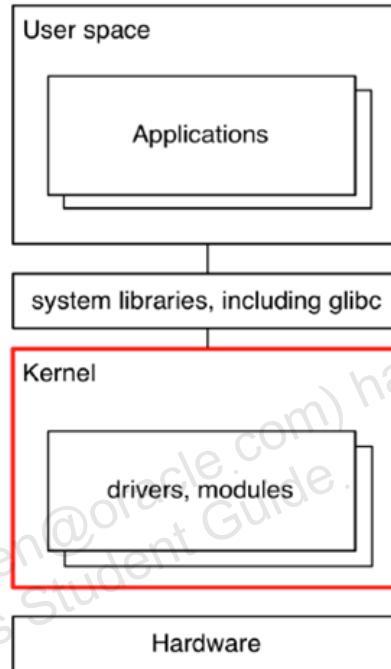
# Objectives

After completing this lesson, you should be able to describe:

- The history of the Linux operating system
- The Linux kernel development model
- Linux distributions
- Oracle's comprehensive Linux solution
- Oracle's contributions to the Linux community
- Oracle Linux's compatibility with Red Hat Enterprise Linux (RHEL)
- Unbreakable Enterprise Kernel

# Linux Kernel

- Linux is modular in design:
  - User space
  - Kernel
- Modular design allows for a large development community, better fault isolation, and security.
- Linus Torvalds developed the original Linux kernel.
- Linux version 0.01 was released in September 1991.
- The name Linux is a combination of Linus and UNIX.



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Linux operating system is a modular system. At the lowest level, the kernel interacts with the hardware and controls and schedules access to resources (CPU, memory, storage, network, and so on) on behalf of applications. Applications run in what is called the user space and only call a stable set of system libraries to ask for kernel services. The glibc library is the GNU C library that defines the system calls and other basic facilities, such as open, malloc, printf, and others. Nearly all applications, including Oracle Database, use this library.

This modular design allows different components of Linux to originate from different developers, each of which has their own specific design goals in mind. A modular design also means that the Linux kernel is independent of applications and interfaces. The result is that application crashes and security vulnerabilities in applications tend to remain isolated, rather than affecting the system as a whole.

The Windows operating system, alternatively, has a high degree of integration with applications and interfaces. This can have significant security and stability consequences. For example, the Windows kernel is heavily integrated with the graphical user interface.

In Linux, each component is configured separately, typically by using text-based configuration files. Configurations are not in a cryptic database (the Windows Registry). Reading and writing configuration information can be done by scripts or applications by using simple text parsing engines. No special application programming interface (API) is required to interface with the system configuration data.

Linus Torvalds developed the original Linux kernel while he was a student at the University of Helsinki in Finland. He had been using MINIX, but MINIX was licensed for educational use only and was not free. He began writing his own kernel and, in August 1991, posted his now famous announcement to the comp.os.minix newsgroup:

"Hello everybody out there using minix –

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus ([torvalds@kruuna.helsinki.fi](mailto:torvalds@kruuna.helsinki.fi))"

# The GNU Project

- The GNU Project was launched in 1983 by Richard Stallman.
- The goal was to create a free, UNIX-compatible operating system.
- GNU stands for “GNUs Not UNIX.”
- The GNU Project created many programs but no kernel.
- The Linux kernel filled the last gap in the GNU system.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Richard Stallman started the GNU project in 1983. The goal of the project was to develop a free UNIX-compatible operating system. Stallman also founded the Free Software Foundation, which continues to sponsor the GNU project. Stallman presented the GNU Manifesto that began with, “GNU, which stands for GNUs Not UNIX, is the name for the complete UNIX-compatible software system which I am writing so that I can give it away free to everyone who can use it.”

Stallman also addressed why he must write GNU, “I consider that the golden rule requires that if I like a program I must share it with other people who like it. Software sellers want to divide the users and conquer them, make each user agree not to share with others. I refuse to break solidarity with other users in this way.”

By 1991, the GNU project had created many programs and utilities with contributions from developers around the world. The Linux kernel was added in 1992, achieving the GNU Project’s goal of developing a free operating system.

GNU’s own kernel, called the Hurd, is not ready for production use. The GNU Hurd is under active development but there is no stable version available.

## GNU General Public License (GPL)

- Richard Stallman wrote the GPL for the GNU Project.
- GPL provides basic software freedoms:
  - Freedom to copy, change, and redistribute software
- Distributors must provide the source code at no cost.
- The Linux kernel version 0.12 was licensed under the GNU GPL.
- The Linux community participates in the advancement of Linux.
- Other free software licenses exist, under which different Linux software packages are licensed.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Richard Stallman also wrote the GNU General Public License (GNU GPL, or simply GPL) for the GNU Project. The GPL provides for some basic software freedoms:

- The freedom to use the software for any purpose
- The freedom to share the software
- The freedom to change the software to suit your needs
- The freedom to share the changes that you make

Software licensed under the GPL can be copied, modified, and redistributed. However, any changes made to the code and redistributed must be distributed with the license. Distributors of GPL software must provide the source code at no cost. This arrangement has been termed a copyleft, because it is the reverse of the way a normal copyright works.

Linus Torvalds released Version 0.11 of his kernel under a freeware license of his own, but in 1992, version 0.12 was relicensed under the GNU GPL, paving the way for programmers around the world to participate in Linux development. These users and developers of Linux are generally referred to as the Linux community.

Linux software is licensed under several other free software licenses in addition to GPL. Following is a partial list of software licenses:

- CPL (Common Public License)
- BSD (Berkeley Software Distribution) License
- AFL (Academic Free License)
- LGPL (GNU Lesser General Public License )
- CC0, CC BY (Creative Commons) License
- Artistic License
- ASL or ASF (Apache Software Foundation) License
- MIT (Massachusetts Institute of Technology) License
- MPL (Mozilla Public License)
- SISSL (Sun Industry Standards Source License)
- AGPL (GNU Affero General Public License)
- Arphic Public License
- LPPL (LaTeX Project Public License)
- UCD (University of California, Davis) License
- Utopia License
- W3C License
- CNRI (Corporation for National Research Initiatives) License
- PSF (Python Software Foundation) License
- Jython License: Python for the Java Platform
- Baekmuk License
- Bitstream Vera Licensing
- Boost Software License
- AMDPLPA License
- GFDL (GNU Free Documentation License)
- IJG (Independent JPEG Group)
- ImageMagick License
- ZPL (Zope Public License)
- Clarkware License
- DMTF (Distributed Management Task Force) License
- EPL (Eclipse Public License)
- Exolab Software License
- FTL (Freetype Project License)

## Linux Kernel Development Model

- Thousands of developers contribute to frequent releases of the kernel.
- Features are pushed upstream through mail lists and IRC.
- New releases deliver stable updates, new features, and performance improvements.
- The Linus Torvalds-led team makes the new releases.
- Mainline kernels are released approximately every three months.
- Kernel branches are available at <http://www.kernel.org>.
- Linux kernel development uses Git as the source-code control system.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Thousands of developers representing hundreds of corporations contribute to frequent releases of the Linux kernel. The development effort has been called one of the largest cooperative software projects ever attempted. Stable updates are created regularly, which include new features, support for new devices, and performance improvements.

The Linux community collaborates through various mailing lists that are set up to handle kernel development. Features are pushed upstream, through these mail lists and Internet Relay Chat (IRC). Upstream is the term used for a community-owned version of a specific project. This is where the development happens and always has the most recent changes. You can subscribe to some of these development mailing lists at <http://vger.kernel.org/vger-lists.html>.

Linus Torvalds leads a team that releases new versions, called “vanilla” or “mainline” kernels. A new version of this mainline kernel is officially released approximately every three months. The mainline branch of development incorporates new features, security fixes, and bug fixes. It is not considered a “stable” branch until it undergoes thorough testing. Separate stable branches for each released version exist. The stable branches do not include the latest features, but do include bug fixes.

A number of kernel versions are currently being maintained as stable kernels. These kernels have patches that are backported to them. These patches are primarily driver updates and security fixes. Kernel branches are available at <http://www.kernel.org>.

Features get pushed into the kernel in different ways. If a kernel feature is not available in an Oracle-supplied kernel, you can submit an RFE (request for enhancement) through Oracle's Bugzilla system at <http://bugzilla.oracle.com/bugzilla> with a detailed explanation of why the feature is being requested. Not all RFEs get implemented, especially if they are known to cause conflicts in other environments or are known to be unstable.

If the feature is something for the mainline kernel, you need to open a discussion in a mail list to debate the merits of a feature and get a consensus agreement that it is a good thing and should be merged with mainline.

Bugs fixes and request for enhancements to kernel features developed by Oracle get submitted directly to Oracle, which then merges and commits the code upstream. Commits are submitted by the maintainers to their respective areas in the Git tree and then peer reviewed, after which they get signed off as being reasonable and are pulled into the mainline tree.

Linux kernel development uses Git as the source code control system. Git provides complete history and revision tracking capabilities. See <http://git-scm.com/> for more information. Oracle has an external Git repository on <http://oss.oracle.com>. All kernel changes are pushed to this Git repository. For example, Oracle's public GIT repository for kernel version 2.6.39 is available at <https://oss.oracle.com/git/?p=linux-uek-2.6.39.git;a=summary>.

# Continuous Mainline Kernel Development

- New hardware brings massive scalability changes and challenges.
  - High Input/Output Operations Per Second (IOP/s) in networking and storage
  - Dramatic bottlenecks in large symmetric multiprocessing (SMP) systems
- Performance is very dependent on power management.
- Mainline kernels have changes to address performance on new hardware.
- Oracle Linux is supported on hardware architectures:
  - x86 (32 bit)
  - x86-64 (64 bit)



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mainline Linux kernel development will never be complete, because new hardware brings in new requirements and new and different challenges. There are massive scalability differences in the way the new hardware works, particularly with regard to high-speed networking and storage. New devices are dramatically faster than the original kernel stack was designed for. A huge number of changes have been put into new kernels to deal with these devices efficiently. However, older enterprise kernels are not able to take advantage of that.

Another hardware advancement of concern for the Linux kernel is the CPU. CPUs rely on power management to perform well. Each individual CPU socket has several cores and several threads, and they all have to work in tandem with awareness from the operating system to maintain power and thermal management. The newer kernels have had a lot of work done to them to address these things. But kernel development will continue due to advances in computer hardware.

Oracle Linux is currently supported on the following hardware architectures:

- x86 (32 bit)
- x86-64 (64 bit)

For a list of Oracle Linux supported releases by hardware platform, see

<https://linux.oracle.com/supported.html>.

# Linux Distributions

- Linux distributions:
  - Are built on top of the Linux kernel
  - Are complete operating systems and more
  - Include compiled binaries and source code
- There are hundreds of Linux distributions.
  - Commercially backed distributions
  - Linux community–driven distributions
- Example:
  - Oracle Linux, Debian, Fedora, Red Hat Enterprise Linux (RHEL), Ubuntu, and many others



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

A Linux distribution (distro for short) is a collection of software built on top of the Linux kernel and offered as a complete package. Distributions are full operating systems plus some additional applications, such as graphics packages, OpenOffice, and others. The kernel is just one component of a Linux distribution.

A typical Linux distribution comprises a Linux kernel, GNU tools and libraries, additional software, documentation and a window system, proprietary applications, free applications, distribution-specific applications for configuration and installation, user manuals, and support information. Most of the software is distributed both as compiled binaries and source code. This allows users to modify and compile the original source code.

There are hundreds of Linux distributions, both commercially backed distributions from companies, such as Red Hat and Novell, as well as Linux community–driven distributions. Some of the more well-known distributions include:

- Oracle Linux
- Debian
- Fedora (a Red Hat–sponsored and community-supported distribution)
- Red Hat Enterprise Linux (RHEL): RHEL is the commercial version of Fedora.
- Ubuntu: Canonical is the vendor behind Ubuntu.

C functions and C++ classes and methods that can be shared by more than one application are broken out of the application's source code and compiled and bundled into a library. The library components are then called by various applications for use when needed.

The GNU toolchain is also included in a distribution. These are a collection of programming tools produced by the GNU Project for developing applications and operating systems. Some of the projects in the GNU toolchain include:

- GNU make: Automation tool for compilation and build
- GNU Compiler Collection (GCC): Suite of compilers for several programming languages
- GNU Binutils: Suite of tools including linker, assembler, and other tools
- GNU Debugger (GDB): Code debugging tool

The X Window System is included, which provides a basis for a graphical user interface (GUI). The window system includes GNOME, KDE, and other GUI components. Proprietary applications such as Adobe Reader and graphics drivers are included. Examples of free applications in a distribution include OpenOffice and Apache.

Oracle Linux offers many software packages (RPMs) and services. Many of these are available on the Unbreakable Linux Network (ULN) in other \_addons channels. All of the Oracle RPMs as well as errata released in between installation DVDs are also available via <http://public-yum.oracle.com>. See the following for information about how to subscribe to the free Oracle Linux errata yum repositories:

[https://blogs.oracle.com/OTNGarage/entry/how\\_to\\_subscribe\\_to\\_the](https://blogs.oracle.com/OTNGarage/entry/how_to_subscribe_to_the)

Some of the different groups of packages for Oracle Linux are listed as follows:

- Administration Tools, Authoring and Publishing
- Development Libraries, Development Tools, Editors
- GNOME Desktop Environment, GNOME Software Development
- Games and Entertainment
- Graphical Internet, Graphics
- Legacy Network Server, Legacy Software Development, Legacy Software Support
- Mail Server, Network Servers, DNS Name Server
- Office/Productivity, Printing Support
- Server Configuration Tools, Sound and Video, System Tools
- Text-based Internet, Web Server
- X Software Development, X Window System
- Cluster Storage, Clustering
- Engineering and Scientific, FTP Server
- Java Development
- KDE (K Desktop Environment), KDE Software Development
- MySQL Database, PostgreSQL Database
- News Server, OpenFabrics Enterprise Distribution
- Windows File Server, Xen



LINUX

## Oracle Linux

- Brings the latest Linux innovations to customers
- Is the best-performing, most modern and reliable Linux OS
- Tracks mainline closely
- Influences Linux roadmap upstream via direct code contributions
- Provides highest-value, enterprise-class support
- Deployment best practices: Full stack tested with real-world workloads
- Provides comprehensive legal indemnification
- Lowers cost
- Ksplice: Apply kernel updates on a running system



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohamed Ameen (mohammed.ameen@oracle.com) has a license to use this Student Guide.

Oracle offers a comprehensive Linux solution including:

- Dedicated development team
- Dedicated QA team
- Dedicated support team
- Dedicated ISV and IHV team
- Oracle Linux training and certification
- Oracle Linux consulting services

Ksplice allows you to do kernel updates without having to reboot the system. A kernel update comes from either Oracle or from the kernel community. The Ksplice team takes the update and works it into a binary patch that is inserted into a running kernel. You apply it by using the Ksplice tools and the patch is up and running.

Security updates are announced to the world, and there is typically a time period between when a security problem is globally known and when system administrators have an opportunity to patch their systems. Ksplice allows you to apply security updates without having to wait for your users to tell you it is okay to take down the system. This problem is even more significant when running a large number of systems. Ksplice allows you to maintain highly available systems that are also very secure.

# Oracle's Technical Contributions to Linux

- Oracle has a dedicated Linux kernel development team.
- Oracle's technical contributions to Linux include:
  - ASMLib
  - Asynchronous IO (AIO) Kernel Subsystem
  - Btrfs file system
  - Oracle Cluster Filesystem (OCFS2)
  - Linux data integrity based on the T10-PI standard
  - Xen Hypervisor
- All Oracle Linux code is available to the Linux community.
- The Git source tree with change logs and commit messages is available at:
  - <http://oss.oracle.com/git/>



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle's long-term vision is focused on continuing to enhance and extend the enterprise-class capabilities of Linux, and this vision is manifest through the many projects and code contributions that Oracle shares with the Linux community. Oracle continues to strengthen its involvement in the Linux community by providing enhancements that facilitate the development and deployment of enterprise Linux solutions. With Oracle Linux, regarding the code developed, 100% of that code becomes available to the open source community for Linux.

The following list includes just some of Oracle's many technical contributions to the Linux community—contributions that benefit anyone using Linux:

- **ASMLib:** A support library for the Automatic Storage Management feature of Oracle Database that simplifies database administration
- **Asynchronous IO (AIO) Kernel Subsystem:** Used to make system calls asynchronous in a generic fashion to ensure that Oracle databases run properly on Linux
- **Btrfs file system:** Designed to address the expanding scalability requirements of large storage subsystems
- **Ext3 file system data-guarded mode:** Oracle's Linux Engineering Team has proposed this new ext3 mode. It maintains all the security protections of data-ordered mode, without requiring all the dirty data on the file system to be written during a single fsync.

- **Kernel I/O Subsystem Tuning:** Oracle Linux kernel engineers are working on creating novel approaches in the area of block I/O, to fully exploit the higher disk speeds of Solid State Disks (SSD).
- **Libstdc++:** Oracle is a major contributor and maintainer of this GNU standard C++ library.
- **NFS on IPv6:** Oracle Linux kernel engineers are working to enable the Linux Network Filesystem (NFS) to run natively on IPv6 networks. The maintainer of NFS, Chuck Lever, is an Oracle employee.
- **Oracle Cluster Filesystem (OCFS) 2 v.1.4:** OCFS2 is an open source, general-purpose, extent-based clustered file system that Oracle developed and contributed to the Linux community. It was accepted into Linux kernel 2.6.16.
- **Oracle Linux Test (OLT) Kit:** Available as open source under the GPL and Artistic licenses, the Oracle Linux Test Kit, derived from the Oracle Validated Configurations program, is designed to verify Linux kernel functionality and stability essential for Oracle Database.
- **Oracle-Validated Configurations for Linux and Virtualization:** These are pre-tested, validated architectures with software, hardware, storage, and networking components with documented best practices for deployment included.
- **PHP:** The Oracle Linux engineering team devotes resources to the improvement and maintenance of PHP and its Oracle-specific extensions. Newer PHP packages in RPM format are available to Linux users for free download.
- **RDS:** Reliable Datagram Sockets (RDS) is an effort to provide a socket API that is uniquely suited to the way Oracle does network Interprocess Communication (IPC). The Oracle Linux kernel development team created an open source implementation of the API for the Linux kernel. The code is now integrated into the OpenFabrics Enterprise Distribution (OFED) stack. OFED aims to deliver a unified, cross-platform, transport-independent software stack for RDMA (remote directory memory access), including a range of standard protocols.
- **T10 Protection Information Model (also known as DIF):** Oracle, in collaboration with Emulex, is implementing a leading, first-of-its-kind initiative to bring enterprise-class data integrity to the Linux platform. An open source interface is being implemented by Oracle to expose the T10 Protection Information Model (also known as DIF—data integrity framework) standard to the Linux kernel and end-user applications. See the following for more information: <http://oss.oracle.com/~mfp/>.
- **Testing of Open Source Projects:** Testing the mainline kernel is essential so the Linux community can get a long-term regression picture of how the kernel performs and works. Mainline kernel testing and quality assurance (QA) benefit the entire community.
- **SELinux:** The maintainer, James Morris, is an Oracle employee.
- **Yet Another Setup Tool (YaST):** YaST helps make system administration easier by providing a single utility for configuring and maintaining Linux systems. Available under GPL, this code can be freely accessed by anyone. The Oracle Linux Engineering team ported the YaST tool to OL from SUSE. Oracle Linux support customers have access to the YaST functionality integrated with the Oracle Management Pack for Linux.
- **Xen Hypervisor:** Consisting of Xen's open source server software and an integrated web browser-based management console, Oracle VM is free, scalable server virtualization software that supports Oracle and non-Oracle applications. Oracle's engineering team contributes heavily to feature development of Xen mainline software. The Xen Hypervisor interface is maintained by Konrad Wilk, an Oracle employee.

# Oracle Linux: Compatible with Red Hat Enterprise Linux (RHEL)

- Source and binaries are fully compatible with RHEL.
- Applications that run on RHEL run on Oracle Linux.
- Trademarks and logos have been removed, but there are no compatibility issues.
- /etc/oracle-release was added to identify code obtained from Oracle.
- Oracle continues to track RHEL releases with Oracle Linux ISO releases and errata stream.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle Linux is fully compatible—both source and binary—with Red Hat Enterprise Linux. Applications that run on RHEL run on Oracle Linux.

## Strict Binary Compatibility

Tremendous effort has gone into assuring that there is no divergence from the original Red Hat source code, given that the main goal of Oracle Linux and the Oracle Linux Support program is to not fragment the Linux code base, but to improve Linux quality and support.

Oracle Linux is built from the very same source code as Red Hat Enterprise Linux (RHEL). A byte-by-byte comparison of the source code against RHEL reveals no differences, the only changes being the removal of trademarks and copyrights.

Trademarks and logos have been removed from a small number of the packages. These are non-functional text or graphic changes that in no way affect any program code, and they do not generate any compatibility issues. Oracle has added its own text file, /etc/oracle-release, so support teams can easily identify that they obtained the code from Oracle.

RHEL provides a text file called /etc/redhat-release, which contains a one-line string identifying the specific distribution release. This file is part of the redhat-release package. Oracle Linux also contains a text file called /etc/redhat-release, which is installed by a package called oraclelinux-release.

Oracle Linux does not include the `redhat-release` package, but the `oraclelinux-release` package provides a set of files equivalent to those in the `redhat-release` package on RHEL.

The Oracle Linux source code is recompiled into binaries and made available for download and produced into CD images. Oracle also applies a number of bug fixes on top of the original code. These fixes are critical for customers to have as soon as possible in their production deployment.

Linux is available under the GPL license, which requires free distribution of the source code. A significant amount of code that is shipped by Red Hat as part of its distribution is actually created by developers outside of Red Hat. Oracle takes the source code that Red Hat makes available under GPL. To offer the Red Hat Compatible Kernel, Oracle tracks the Red Hat distribution closely to ensure compatibility for users.

### Fully Compatible Updates and Errata

Oracle synchronizes bug fixes at regular intervals with RHEL to maintain full compatibility. Whenever a new version of an individual package (an erratum) gets released by Red Hat, not just as part of an update release, the corresponding package for Oracle Linux is made available very quickly, in a matter of hours. If a package has no trademarks and no Oracle-specific patches, it is simply recompiled and re-issued for Oracle Linux immediately after going through testing.

If a package has trademarks or Oracle Linux–specific changes, Oracle examines the source code and compares it against the bug fixes that have been already applied and released as part of Oracle Linux. If the Oracle patches are still relevant, they are re-applied, but if the problems have been fixed in the Red Hat version, whether in the same or in a different way, the Oracle-specific patches are dropped and the package is recompiled (always checking for trademarks and copyrights issues) and released as part of Oracle Linux via the Unbreakable Linux Network (ULN).

For official updates of existing major releases, for example RHEL 6 Update 1, Oracle rebundles the Red Hat patches in the update and re-issues them as Oracle Linux 6 Update 1, including free ISOs, almost immediately.

Bug fixes and security errata are available for free on <http://public-yum.oracle.com>. You may subscribe to the Oracle Linux errata mailing list (el-errata) from this site as well.

As a new major RHEL release is issued, there is usually the need to do some additional testing before Oracle can consider it an official Oracle Linux version because Red Hat does not conduct Oracle related testing. For instance, when RHEL 5 was released, Oracle ensured that the corresponding Oracle Linux product had been well tested before issuing its own version of it, because in the past, critical bugs were discovered and fixed during this process.

For more information on compatibility, download an independent third-party white paper (PDF) from the Edison Group, Oracle Linux: True Enterprise-Quality Linux Support. The white paper is available at: <http://www.oracle.com/us/technologies/linux/UBL-edison-066204.pdf>.

# Unbreakable Enterprise Kernel

- Oracle announced the Unbreakable Enterprise Kernel in September 2010.
- It is used by Exadata and Exalogic for extreme performance.
- It is available in both Oracle Linux 5 and Oracle Linux 6.
- Beginning with Oracle Linux 5.5, you have a choice:
  - Red Hat Compatible Kernel
  - Unbreakable Enterprise Kernel
- Oracle is committed to offering compatibility with Red Hat.
- Full support is offered for customers running either kernel.
- Existing applications run unchanged.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

In September 2010, Oracle announced the new Unbreakable Enterprise Kernel for Oracle Linux as a recommended kernel to deploy with Oracle Linux 5 or Red Hat Enterprise Linux (RHEL) 5. Beginning with Oracle Linux 5.5 (default in 5.6), you now have a choice when it comes to the kernel, either the Red Hat Compatible Kernel or the Unbreakable Enterprise Kernel. In Oracle Linux 5.6, the Unbreakable Enterprise Kernel became the default kernel.

The initial motivation for creating the Unbreakable Enterprise Kernel was to have a modern and best-performing Linux kernel for the Exadata and Exalogic engineered systems. The kernel needed to scale with the larger number of CPUs, memory, and InfiniBand connects.

Unbreakable Enterprise Kernel is heavily tested with Oracle workloads and therefore recommended for Oracle deployments and all other enterprise deployments. Oracle is committed to offering compatibility with Red Hat, and continues to release and support the Red Hat Compatible Kernel as part of Oracle Linux, for customers that require strict RHEL compatibility. Under the Oracle Linux Support Program, customers can receive full support for Oracle Linux running with either kernel.

Using the Unbreakable Enterprise Kernel instead of the Red Hat compatible kernel changes only the kernel. Nothing changes in the user space. Existing applications run unchanged regardless of which kernel is used. Using a different kernel does not change system libraries such as glibc. The glibc version in Oracle Linux 6 is 2.12, regardless of the kernel version.

# Unbreakable Enterprise Kernel Release 1

- Unbreakable Enterprise Kernel R1 is based on a stable 2.6.32 Linux kernel.
- Unbreakable Enterprise Kernel R1 features include:
  - Latest InfiniBand software stack, OFED 1.5.1
  - Advanced support for large NUMA systems
  - Receive packet steering and receive flow steering
  - SSD detection
  - Data integrity up to the storage area network (SAN)
  - OCFS2 1.6



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Version 1 of the Unbreakable Enterprise Kernel is based on a stable 2.6.32 Linux kernel and provides additional performance improvements, including:

- Improved IRQ (interrupt request) balancing
- Reduced lock contention across the kernel
- Improved network I/O via receive packet steering and RDS improvements
- Improved virtual memory performance

Unbreakable Enterprise Kernel includes optimizations developed in collaboration with Oracle's Database, Middleware, and Hardware engineering teams to ensure stability and optimal performance for the most demanding enterprise workloads.

In addition to performance improvements for large systems, the following list includes new Unbreakable Enterprise Kernel features that are relevant to Linux running in the data center:

## Latest OFED Stack (1.5.1)

Unbreakable Enterprise Kernel R1 includes the latest InfiniBand Stack (OFED) 1.5.1. OpenFabrics Enterprise Distribution (OFED) implements Remote Direct Memory Access (RDMA) and kernel bypass mechanisms to deliver high-efficiency computing, wire-speed messaging, ultra-low microsecond latencies and fast I/O for servers, block storage, and file systems.

This also includes an improved RDS (reliable datagram sockets) stack for high-speed, low-latency networking. As an InfiniBand Upper Layer Protocol (ULP), RDS is used to send IPC datagrams (up to 1 MB) reliably, and is used currently in Oracle Real Application Clusters (RAC) and Exadata/Exalogic products.

## Advanced Support for Large NUMA Systems

Unbreakable Enterprise Kernel R1 includes a number of additional patches to significantly improve performance on Non-Uniform Memory Access (NUMA) systems with many CPUs and cores. These include:

- A patch to list message-signaled interrupts (MSI) for each device in the sysfs file system:
  - Before, when MSI-X mode was enabled for a PCI device, there was no entry in sysfs that displayed the IRQs.
  - The interrupts were only displayed in /proc/interrupts, but it was impossible to determine which interrupts were used by which device when there were multiple identical devices in the system.
- A modified irqbalance utility helps ensure that IRQs are kept on NUMA local CPUs.
- Reduced runqueue lock contention by making improvements related to IPC semaphores.
- A patch tries to reduce runqueue lock contention by ordering the wakeups based on the CPU the waiting process was on when the process went to sleep.

## Receive Packet Steering (RPS)

RPS improves overall networking performance, especially at high loads. RPS distributes the load of received network packet processing across multiple CPUs and ensures that all packets for a specific IP address/port combination are handled by the same CPU core. This allows protocol processing (for example, IP and TCP) to be performed on packets in parallel and avoids performance penalties that can occur due to the resulting cacheline bouncing. This solution removes a bottleneck where a single CPU core could be saturated from processing network interrupts. This feature has been backported from the mainline Linux 2.6.35 kernel.

To enable receive packet steering, you have to place a CPU mask into:

```
/sys/class/net/xxx/queues/rx-0/rps_cpus (where xxx is your interface name)
```

The CPU mask takes the same form as the masks for the taskset command, for example:

```
echo 0x55 > /sys/class/net/eth0/queues/rx-0/rps_cpus
```

## Receive Flow Steering (RFS)

RFS can be considered the second stage of receive packet steering. RFS is an extension of RPS that makes sure that now that processing of network packets is happening in parallel, it is done in a coordinated fashion. Instead of performing an IP/Port match, it is doing an application match, directing the flow of traffic to where the application is waiting for it. If an application issues system calls that trigger network packets to be sent and received, its footprint is logged to the CPU currently executing it and any incoming packets for this application meet up on the same CPU, thus improving CPU locality and minimizing the performance penalty. This is more directed than receive packet steering alone.

Together with RPS, this can result in tremendous performance improvements—Oracle-internal tests have shown 50% faster IP over InfiniBand results on a two-socket system.

## SSD Detection

Unbreakable Enterprise Kernel R1 includes code in the kernel block layer to detect solid state disks and tune itself accordingly. The result of detection can be found in:

/sys/block/xxx/queue/rotational (where xxx is the block device)

When the kernel sees that it has an SSD, it is able to bypass optimization code for spinning media and do things that make more sense on SSDs. Most of the changes are about being able to dispatch I/O quickly and without delay to the SSD. With spinning media, the kernel goes to great lengths to batch up I/O and to not make the device seek. The SSDs want to process as many I/Os, especially writes, in parallel at a time. They perform dramatically better if the I/Os are not held back and, instead, are sent immediately to the device.

## Data Integrity Up to SAN

Unbreakable Enterprise Kernel R1 includes data-integrity features. Data is verified from the database, all the way down to the individual storage spindle or device, to make sure that the data has not changed. This picks up corruptions from memory, and corruptions from any piece of the storage stack between the database, for example, and the actual spindle.

The Linux data integrity framework (DIF) enables applications or kernel subsystems to attach metadata to I/O operations, allowing devices that support DIF to verify the integrity before passing them further down the stack and physically committing them to disk.

Data Integrity Extensions (DIX) is a hardware feature that enables the exchange of protection metadata between the host operating system and the host bus adapter (HBA) and helps to avoid corrupt data from being written (silent data corruption), allowing a full end-to-end data integrity check.

The data integrity–enabled ASM kernel driver protects against data corruption from application to disk platter. ASMLib is a library add-on for the Automatic Storage Manager of Oracle Database.

## OCFS2 1.6

Unbreakable Enterprise Kernel R1 includes the Oracle Cluster Filesystem 2 (OCFS2) version 1.6 kernel module. New features include:

- JBD2 support: This gives 64-bit block numbers and, theoretically, support for 4 petabyte (4 PB) file systems. With JBD1 the limit was 16 terabytes (16 TB) per file system. JBD stands for “journal block device.”
- Quota support
- Extended attributes: The value of each extended attribute can be as large as a regular file. Which is larger than even ext3 (“third extended file system”) can do.
- POSIX ACL support
- Support for user space cluster stacks
- Security attributes
- Metadata checksums and ECC: All metadata blocks in OCFS2 now have a checksum field.
- Improved inode allocation: For file systems with a huge number of files
- Indexed directories: This improves performance of lookups of a single name.
- Reflinks: This creates a target inode that shares the data extents of the source inode in a copy-on-write fashion.

# Unbreakable Enterprise Kernel Release 2

- Version 2.6.39
- Based on upstream Linux Kernel 3.0.16
- Many scalability improvements and new features:
  - DTrace
  - Btrfs file system
  - Transcendent memory
  - Resource isolation: Cgroups
  - OS isolation: Linux containers (technical preview)
  - Transparent huge pages
  - Transmit packet steering (XPS)
  - Built-in virtual switch (technical preview)



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle calls the second Unbreakable Enterprise Kernel version “2.6.39,” but it is actually based on the 3.0.16 Linux kernel. Some applications expect the kernel number to start with 2.6 and do not understand any kernel number that does not begin with 2.6. There are no other compatibility problems other than this one version number.

As with Unbreakable Enterprise Kernel R1, there are a number of scalability improvements in Unbreakable Enterprise Kernel R2. Improvements include further refinements of the interrupt scalability work mentioned before. The scheduler is much better tuned, especially for Java workloads.

Dtrace and Btrfs file system are covered in more detail but some of the other features listed in the slide are described below.

## Transcendent Memory: Cleancache

Transcendent memory (tmem) is a memory area to cache clean memory pages. It eliminates costly disk reads and has been shown to improve performance on a broad range of workloads. It is exposed via the VFS layer for easy integration with existing file systems. The cache helps to keep more pages of the file system page cache longer in memory.

Transcendent memory is especially beneficial in virtualization, to make better use of the hardware.

With transcendent memory, instead of giving the memory to individual virtual machines, you keep more of it in the hypervisor. The virtual machines then ask for cached memory that they have access to. This is typically used with file data. The virtual machines can easily pull the data off a disk if the hypervisor is not able to hold onto it for them. It allows capacity and the usage level of the virtual machines to be increased.

Similarly, the transcendent memory subsystem makes it possible to have a compressed page cache, called the zcache. You gain extra memory space by compressing the pages. This additional cache uses LZO compression and results in fewer disk I/O operations.

### **Resource Isolation: Cgroups**

Cgroups provide more fine-grained control of CPU, I/O, and memory resources. You can associate a set of CPU cores and memory nodes with a group of processes that make up an application or a group of applications. This enables subsetting larger systems, more fine-grained control over memory, CPUs and devices, and isolation of applications.

For example, with very large NUMA systems, you make the best use of it by compartmentalizing. Cgroups give you a great deal of control over how you want to set up a system, which memory to give, and which CPUs to give to an individual task. You can pin processes to the same NUMA node and use NUMA-local memory. Cgroups facilitate database consolidation on large NUMA servers, I/O throttling support, and device whitelisting. Cgroups work inside virtual guests as well.

### **OS Isolation: Linux Containers**

Linux containers are a technical preview as of this writing, but they provide multiple user space versions of an operating system on the same server. Containers are the next step up from cgroups for using system resources more efficiently. They provide an isolated environment with its own process and network space.

Containers are similar to virtualization in that you are maintaining semi-private instances of an operating system. But in a container setup, there is no hypervisor involved. You are just compartmentalizing the system in such a way that each container thinks that it is in its own box. The advantage is you can isolate environments and control how resources are allocated without the virtualization overhead.

Tools exist to start, stop, freeze, create, and destroy containers. These tools are similar to virtualization tools but without the needs for a hypervisor.

### **Transparent Huge Pages**

Transparent huge pages better supports the memory management capabilities of modern CPUs and provides much more efficient ways to dynamically manage physical memory. Instead of using memory in 4 KB chunks, huge pages of memory are 2 MB chunks. In large systems with terabytes of RAM, using 2 MB pages dramatically reduces overhead in the virtual memory subsystem, and the kernel is able to keep track of the pages much more efficiently. Frequently accessed virtual addresses for memory-intensive workloads can be better cached.

Transparent huge pages, in the background, collect pages that you are using and turn them into huge pages. This allows you to get all that efficiency without making changes in your applications.

### **Transmit Packet Steering (XPS)**

Transmit packet steering spreads outgoing network traffic across CPUs on multiqueue devices. XPS selects a transmit queue during packet transmission based on configuration by mapping the CPU transmitting the packet to a queue.

XPS is kind of the opposite of receive packet steering, as XPS is oriented towards picking the most efficient place to send the packet, where efficiency is defined as lock contention and NUMA cost inside the CPU itself. XPS is able to control which CPUs are used to transmit a packet for a given networking queue or networking device. This can have a very dramatic impact on performance.

### **Networking: Built in Virtual Switch**

The built-in virtual switch is also a technical preview as of this writing. It is based on the Open vSwitch project and is used to create virtual networks, Virtual Network Cards (VNICs), VLANs, and virtual switches. Features include resource management, QoS, and sFlow monitoring. The built-in virtual switch is exceptionally useful in virtualized environments in terms of maintaining good network setups across all the virtual machines. It provides a much more powerful and much easier-to-use way to dynamically create and manage virtual network devices. It is a replacement for the in-kernel bridging code of the Linux kernel. It can operate as a soft switch within the operating system or as a control stack for switching silicon.

### **Other Scalability Improvements**

Some of the other scalability improvements in the Unbreakable Enterprise Kernel R2 include NUMA fixes and lock contention optimizations. The VFS subsystem for the kernel is much more efficient with directory cache improvements for multithreaded and single-threaded workloads. The VFS subsystem does all the mappings from a file name to the individual file system responsible for that file. The file systems, specifically ext4, XFS, and Btrfs, have had major improvements as well. The Big Kernel Lock (BKL) was replaced with much more fine-grained locking code. The BKL was added for the very first SMP-capable Linux kernel. It was a very crude lock that had a lot of contention on it and it is now gone from the kernel. One of the last uses of BKL was in the NFS subsystem and also in file locking code, such as the flock or the fcntl locks on a file. These are now dramatically more scalable.

# Unbreakable Enterprise Kernel Release 3

- Based on mainline Linux Kernel 3.8.13
- Many scalability improvements and new features:
  - Inclusion of DTrace 0.4 for Linux
  - Btrfs file system improvements
  - Improved support for control groups and Linux containers.
  - Transmit packet steering (XPS)
  - Built-in virtual switch (technical preview)



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Unbreakable Enterprise Kernel Release 3 (UEK R3) is Oracle's third major release of its heavily tested and optimized operating system kernel for Oracle Linux 6 on the x86\_64 architecture. It is based on the mainline Linux version 3.8.13.

The 3.8.13-13 release also updates device drivers and includes bug and security fixes. Some notable improvements in functionality and new features include:

- Numerous stability and scalability enhancements
- Inclusion of DTrace 0.4 for Linux into the kernel (no longer a separate kernel image). DTrace for Linux now supports probes for user space statically defined tracing (USDT) in programs that have been modified to include embedded static probe points.
- Btrfs file system improvements (subvolume-aware quota groups, cross-subvolume reflinks, Btrfs send/receive to transfer file system snapshots or incremental differences, file hole punching, hot replacing of failed disk devices)
- Improved support for Control Groups (cgroups) and Linux containers (LXC)
- The ext4 file system can now store the content of a small file inside the inode (inline\_data).
- TCP fast open (TFO) can speed up the opening of successive TCP connections between two endpoints.

## Notable Improvements (continued)

- XFS journals implement checksums for verifying log integrity.
- A zero huge page complements the existing implementation of zero 4-KB pages as a performance optimization.
- Automatic balancing of memory allocation for NUMA nodes.
- The value of the SCSI error-handling timeout is now tunable. If a SCSI device times out while processing file system I/O, the kernel attempts to bring the device back online by resetting the device, followed by resetting the bus, and finally by resetting the controller. The error-handling timeout defines how many seconds the kernel should wait for a response after each recovery attempt before performing the next step in the process.

## Technology Preview

The following features included in the Unbreakable Enterprise Kernel Release 3 are still under development, but are made available for testing and evaluation purposes.

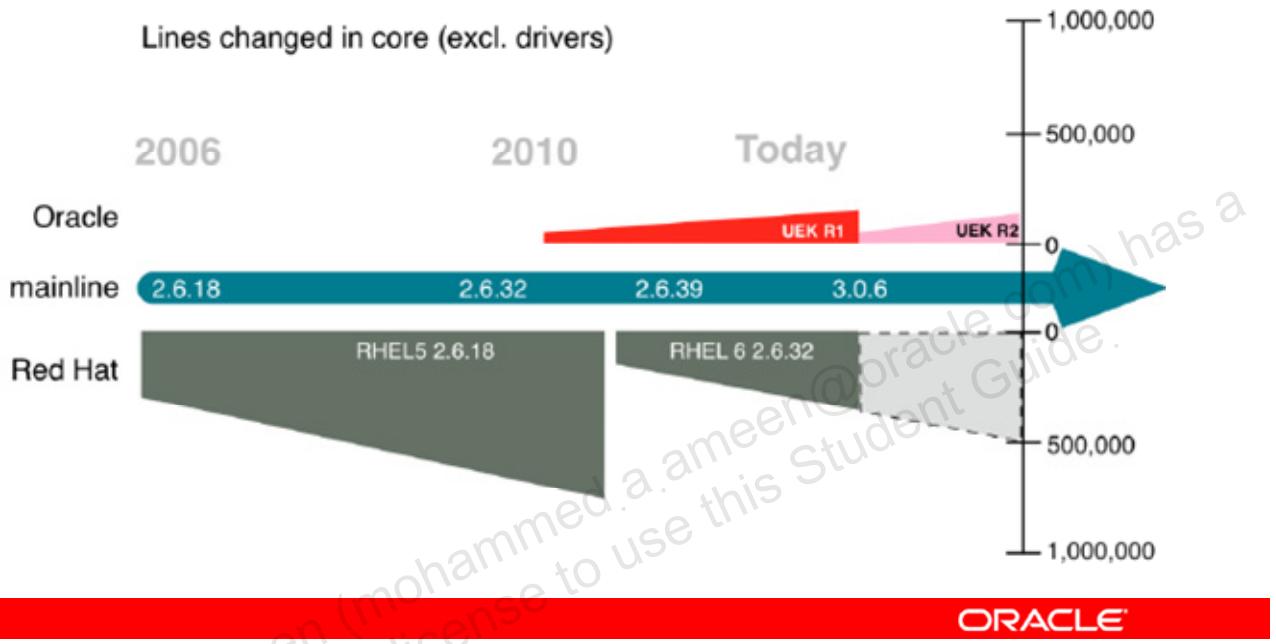
- Kernel module signing facility: Applies cryptographic signature checking to modules on module load, checking the signature against a ring of public keys compiled into the kernel. GPG is used to do the cryptographic work and determines the format of the signature and key data.
- NFS version 4.1 client: Supports Sessions, Directory Delegations, and parallel NFS (pNFS) as defined in RFC 5661
- Transcendent memory: Transcendent memory (tmem for short) provides a new approach for improving the utilization of physical memory in a virtualized environment by claiming underutilized memory in a system and making it available where it is most needed. From the perspective of an operating system, tmem is fast pseudo-RAM of indeterminate and varying size that is useful primarily when real RAM is in short supply. To learn more about this technology and its use cases, see the Transcendent Memory project page at <http://oss.oracle.com/projects/tmem/>.

Unbreakable Enterprise Kernel Release 3 can be installed on Oracle Linux 6 Update 4 or newer, running either the Red Hat-compatible kernel or a previous version of Unbreakable Enterprise Kernel. Unbreakable Enterprise Kernel Release 3 is supported on the x86\_64 architecture but not on x86.

Release notes are available at: [http://docs.oracle.com/cd/E37670\\_01/E48380/html/index.html](http://docs.oracle.com/cd/E37670_01/E48380/html/index.html).

## Tracking Mainline

Unbreakable Enterprise Kernel is closer to the Linux mainline kernel and more modern than the Red Hat kernel.



This slide illustrates that the Unbreakable Enterprise Kernel is closer to the Linux mainline kernel and more modern than the Red Hat kernel. The enterprise kernels from Red Hat are frozen for the lifetime of the product, making it difficult to backport new features on those kernels.

# DTrace

- DTrace:
  - Is a Solaris tool, available since 2005
  - Allows static tracing by using instrumentation compiled into kernel and applications
  - Allows dynamic tracing by defining probe points “on the fly”
- Probes and actions at probe points are defined by scripts written in the “D” language.
- Many types of providers:
  - DTrace, syscall, profile, sysinfo, vminfo, fpuinfo, sched, io, iscsi, and so on
  - A Pid provider for dynamic tracing in user space applications
- Speculative tracing allows the filtering of events and data presented to the user after probes fire.

**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle Linux is the only Linux distribution to support Dtrace. Dtrace is fully integrated with Oracle Linux 6.5 and Unbreakable Enterprise Kernel Release 3. It allows static and dynamic tracing of your applications and your kernel. It does static tracing by allowing instrumentation compiled into the kernel and the applications. At specific points of execution in your code, you can activate the probes and designate actions, such as collecting and displaying information.

DTrace also allows you to dynamically define probe points on the fly. That means they are not precompiled into the kernel or into your application. Usually, probes and probe points are defined by the user using scripts written in a language called D.

DTrace has providers, which are basically categories of probes. Some of the common providers are listed in the slide.

For user space applications, probes are in various popular applications and programs like MySQL, Perl, and Java that have been working with DTrace for several years. The Pid provider also allows dynamic tracing for user space applications by probing every instruction, not just by specifying spots in your application at compile time.

Speculative tracing allows output to be filtered to drop uninteresting events or events not associated with what you are trying to trace.

## Btrfs File System

- Designed for large files and file systems
- Simplified administration
- No volume manager needed
- Easy to add and remove capacity
- Online defragmentation and scrubbing
- Built-in data integrity
- RAID
- Flexible
- File and file subvolume snapshots
- Transparent compression



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Btrfs is a new file system for Oracle Linux, although it has been around for a few years. Btrfs is a general-purpose file system that scales very well for large systems and large file systems.

Features of Btrfs include simplified administration and an internal volume manager that makes it very easy to add and remove capacity. It is also easy to do online defragmentation and online scrubbing of the drives. Btrfs has checksums on all of the data and metadata, and scrubbing is a way to read this to be sure that the values are correct. If any values are not correct, the data can be pulled off another mirror, or another copy of the drive, still maintaining the internal Btrfs RAID.

Btrfs is entirely built upon scalable snapshotting. There is no difference between the snapshot of something and the original, so they can be snapshotted again, providing flexibility.

Btrfs also has transparent compression, which is enabled by using a mount option. You can either use LZO or zlib to have everything compressed in the background.

For a demonstration of Btrfs, see <http://www.youtube.com/watch?v=hxWuaozpe2I>.

Beginning with Oracle Linux 6.3, a boot iso is available that boots up the Unbreakable Enterprise Kernel as the install kernel and uses Btrfs as the default file system for installation.

## Oracle Linux Release Notes

- Oracle publishes release notes for each version.
- Release notes and all product documentation are available at: [http://docs.oracle.com/cd/E37670\\_01/](http://docs.oracle.com/cd/E37670_01/).
- What's new in this release?
- Changes from the upstream release:
  - Packages modified, removed from upstream release
  - New packages added by Oracle
- Kernel:
  - Driver updates, new features



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Refer to the release notes for information regarding each release of Oracle Linux. For example, the following is part of “What’s new in Oracle Linux 6”:

- **ext4 file system:** The ext4 file system is installed by default.
- **XFS:** Oracle Linux 6 includes XFS as an optional file system.
- **ftrace:** ftrace is a tracing framework for analyzing performance and latency in the kernel.
- **Performance Counters for Linux (PCL):** The performance counter subsystem keeps track of hardware and software events without affecting performance.
- **Powertop:** Powertop is a new user space tool that helps you reduce server power usage by identifying power hungry processes.
- **Latencytop:** LatencyTOP is a Linux tool for software developers aimed at identifying where system latency occurs.
- **Yum-only access to Unbreakable Linux Network (ULN):** Oracle Linux 6 no longer contains up2date for access to ULN. Instead, packages are managed by using Yum.
- Additional new features are provided via Unbreakable Enterprise Kernel.

In addition, changes from the upstream release, kernel updates, and more such information are available from the Oracle Linux release notes.

## Summary

In this lesson, you should have learned:

- The history of the Linux kernel
- The relationship between the Linux kernel and the GNU Project
- The purpose of the GPL
- The development model of the Linux kernel
- The definition of a Linux distribution
- Oracle's comprehensive Linux solution
- Oracle's technical contributions to Linux
- Oracle Linux compatibility with RHEL
- Oracle Unbreakable Enterprise Kernel versions and features



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Quiz

Oracle Linux offers a Red Hat–compatible kernel as well as a kernel that is optimized for Oracle applications.

- a. True
- b. False



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Quiz

Which of the following features are included in the Unbreakable Enterprise Kernel?

- a. Ksplice
- b. Btrfs file system
- c. OS isolation by using Linux containers
- d. Built-in virtual switch



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 2: Overview

The practices cover the following topics:

- Introduction to Oracle Linux quiz
- Viewing kernel information

## Installing Oracle Linux

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Obtain Oracle Linux operating system software
- Describe the Anaconda installer
- Install Oracle Linux
- Describe the FirstBoot utility



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Many of the steps required during installation, such as configuring the network, partitioning storage devices, creating file systems, configuring the GRand Unified Bootloader (GRUB), and installing software packages are covered in later lessons. The objective of this lesson is simply for you to become familiar with the Oracle Linux installation process.

# Obtaining Oracle Linux

- Obtain Oracle Linux from:
  - Oracle Software Delivery Cloud:  
<https://edelivery.oracle.com/linux/>
  - Oracle Public Yum Server:  
<http://public-yum.oracle.com>
  - Source and debug information available from:  
<http://oss.oracle.com/>
- Obtain errata for free from <http://public-yum.oracle.com>.
  - Subscribe to Oracle Linux errata mailing list from this site.
- Oracle Linux 6.5 is available for:
  - 32-bit x86 systems
  - 64-bit x86 systems



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

True to the open source philosophy, Oracle has provided Oracle Linux software as a free download from day 1. Anyone can download the binaries, installation media, and the source code for free without a support subscription.

Oracle Linux DVD images (ISOs) can be obtained from the Oracle Software Delivery Cloud: <http://edelivery.oracle.com/linux>. Oracle also provides free access to the individual RPM packages via public yum repositories from <http://public-yum.oracle.com/>. Source and debug information packages are available from <http://oss.oracle.com/>.

A new set of ISO DVD images is made available for free download for every minor release (for example, Oracle Linux 6 Update 5) of Oracle Linux. Therefore, users of Oracle Linux will be at most one update release behind, even if they do not purchase any support contract.

You can obtain errata (bug fixes, security fixes, enhancements) for free as well as subscribe to the Oracle Linux errata mailing list from the public yum server.

Oracle Linux 6.5 is available for 32-bit and 64-bit x86 systems.

# Oracle Software Delivery Cloud

## Oracle Linux 6.5 download page: Example

The screenshot shows the Oracle Software Delivery Cloud interface. At the top, there's a navigation bar with links for 'Sign Out', 'Cloud Portal (Oracle Linux/VM)', 'Language (English)', and 'FAQs'. Below the navigation is a search bar with 'Search' and 'Download' buttons. A tip message says: 'View the Readme file(s) to help decide which files you need to download.' Another message below it says: 'Print this page with the list of downloadable files. It contains a list of the part numbers and their corresponding description that you may need to reference during the installation process.' A note at the bottom of the page reads: 'Hi Craig, by clicking the download button, you agree Oracle's Terms & Conditions apply to your use of the software on this portal. Not Craig? Do not download the software and [login with your account](#)'.

Select	Name	Part Number	Size (Bytes)
<a href="#">Download</a>	Oracle Linux Release 6 Update 5 source DVD 2	V41366-01	2.0G
<a href="#">Download</a>	Oracle Linux Release 6 Update 5 source DVD 1	V41365-01	3.1G
<a href="#">Download</a>	Oracle Linux Release 6 Update 5 for x86_64 (64 Bit)	V41362-01	3.7G
<a href="#">Download</a>	Oracle Linux Release 6 Update 5 UEF Boot ISO Image for x86_64 (64 bit)	V41364-01	227M
<a href="#">Download</a>	Oracle Linux Release 6 Update 5 Boot iso Image for x86_64 (64 bit)	V41363-01	214M
<b>Total: 5</b>			

**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This screen lists the download options for Oracle Linux 6.5 from the Oracle Software Delivery Cloud: <http://edelivery.oracle.com/linux>. As shown, Oracle Linux 6.5 comes in three DVD ISO images for x86\_64 bit architecture. One is the complete distribution (part # V41362-01) and the other two are marked as boot ISO images. The full distribution ISO is what is typically needed for installations of Oracle Linux 6.5.

The boot ISO images are slightly over 200 MB and only contain the boot kernel and installer. They do not have all the software packages (RPMs). You can use these image files to produce minimal boot media such as bootable CDs, DVDs, or USB devices with which you can boot a system when you plan to complete the installation from an installation source available on a hard disk or over a network connection. You do not need to download the source DVDs to install Oracle Linux.

To install Oracle Linux 6.5 from DVD media, download the compressed binary DVD images. Verify the downloaded media file by comparing its `sha1sum` or `md5sum` with the published `sha1sum` or `md5sum`. Use DVD burning software to write the DVD image directly to DVD. Do not simply copy the files onto the DVD. You must use a DVD burner that can accept an iso image as input, and that can create a bootable DVD from it. To test if you have burned the images correctly, insert a burned DVD and ensure that multiple files and directories are visible. Insert the DVD into your system, boot from DVD, and follow the on-screen instructions to deploy Oracle Linux.

## Anaconda Installer

Anaconda:

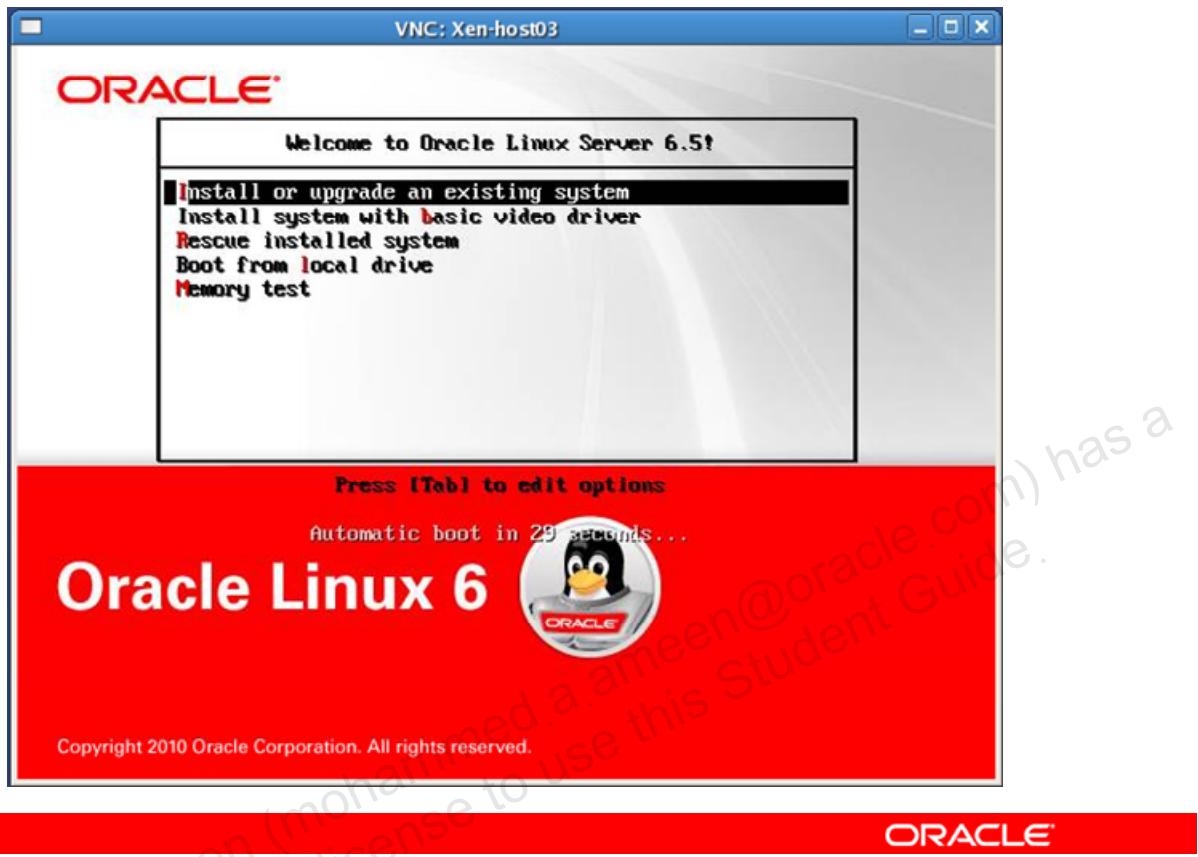
- Is the installation program used by Oracle Linux
- Runs in textual or graphical mode
- Supports installation from local or remote sources
  - CD, DVD, USB drive, or images stored on a hard drive
  - NFS, HTTP, or FTP
- Installation can be automated with Kickstart for unattended installation.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Anaconda is the installation program used by Oracle Linux and other Linux distributions. Anaconda identifies the system's hardware, creates the appropriate file systems, and installs or upgrades the operating system. Anaconda runs in textual or graphical mode and supports installation from multiple sources, both local and remote. You can install from CD or DVD, USB flash drive, from images stored on a hard drive, or from remote servers using NFS, HTTP, or FTP. Installation can be automated with Kickstart for unattended installation.

## Boot Menu



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This slide shows the Boot Menu, which is the first window to appear during an installation. The default option of “Install or upgrade an existing system” is automatically selected within 60 seconds. Press Esc at the boot menu to get a boot prompt.

### **Install or upgrade an existing system**

Select this default option to use the graphical installation program to install Oracle Linux.

### **Install system with basic video driver**

Use this option if the default option causes a distorted or blank screen, which is the result of the installation program’s inability to load the correct driver for your video card.

### **Rescue installed system**

Choose this option if you are unable to boot an installed system. Rescue mode allows you to repair partitions, edit configuration files, and fix a variety of boot problems.

### **Boot from local drive**

This option boots the system from the hard disk.

### **Memory test**

This option runs a memory test utility to verify RAM.

## Boot Options

- Press Esc at the boot menu to display the `boot:` prompt.
- Use the following syntax to provide boot options:
  - `linux option1 option2 option3`
- Some examples of boot settings include:
  - Language
  - Display resolution
  - Interface type
  - Installation method
  - Network settings



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Press Esc at the boot menu to display the `boot:` prompt. From the boot prompt, you can specify multiple boot options for advanced installations. Use the following syntax to provide boot options:

`boot: linux option1 option2 option3`

To specify multiple boot options, separate each option by a single space. Some examples of boot settings include:

- Language
- Display resolution
- Interface type
- Installation method
- Network settings

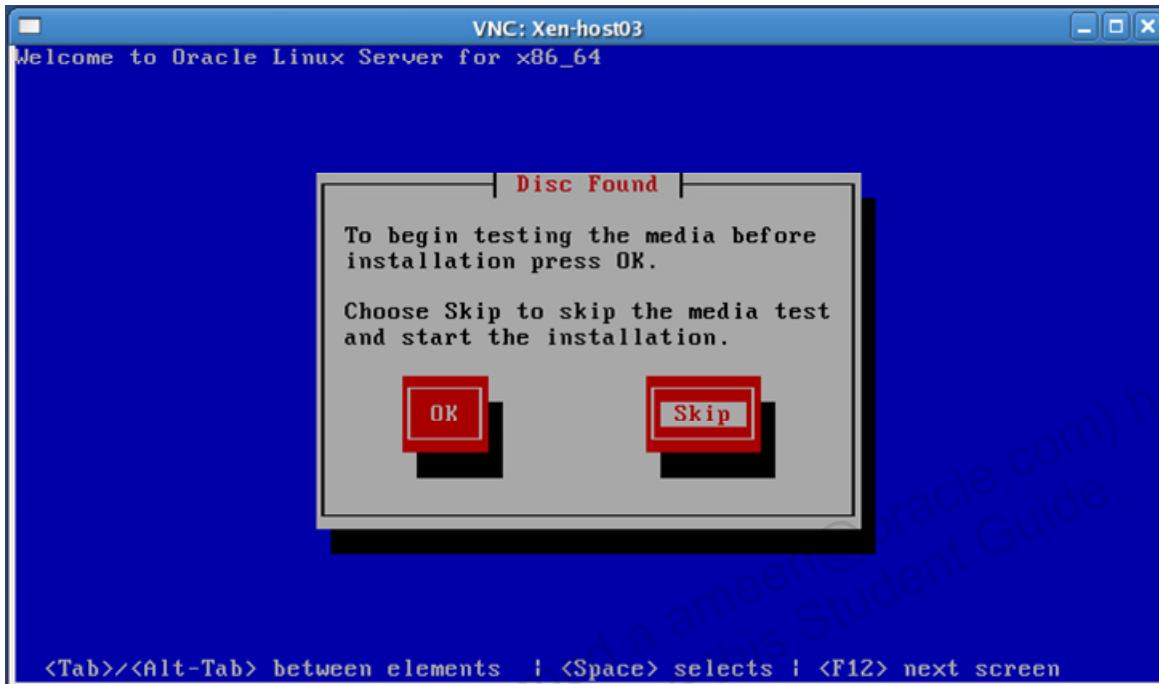
For example, to specify language and keyboard layout, enter:

`boot: linux lang=value keymap=value`

To run the installation in text mode, enter:

`boot: linux text`

# Media Test



ORACLE

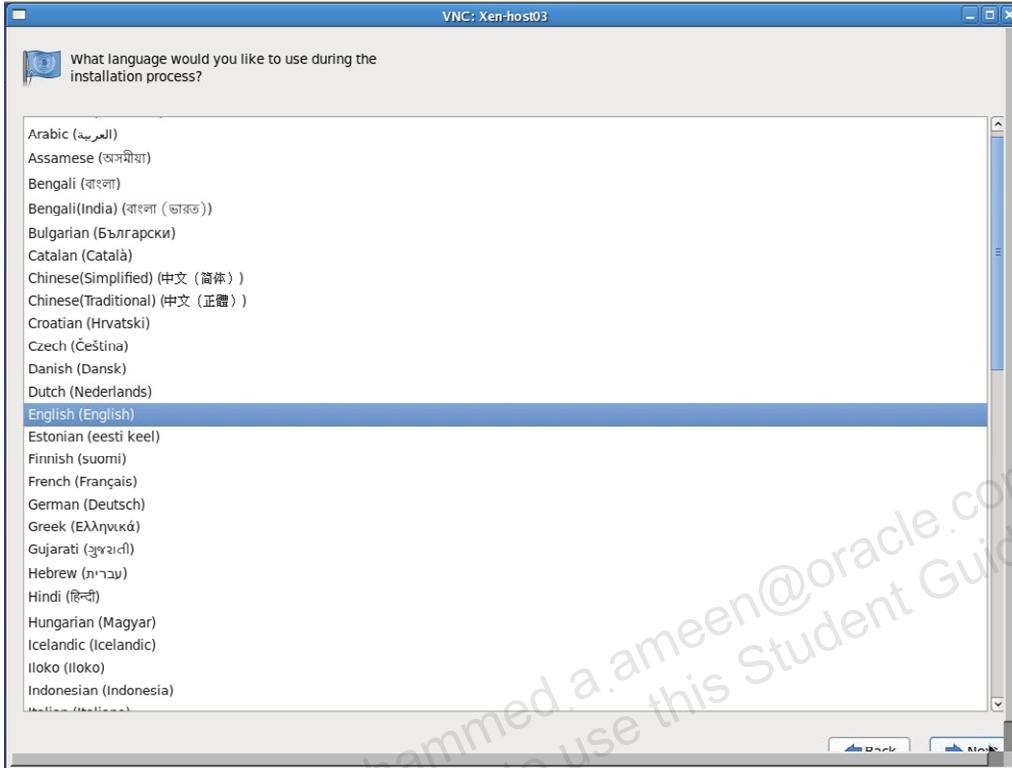
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Media Test window appears next. You have the option to test the installation media before starting the installation. Testing takes a few extra minutes but is worthwhile if you have concerns about corruption or errors with the media.

## Logo

Anaconda then displays the Logo window. Each window displayed by Anaconda provides a button labeled Next in the lower-right corner. Most windows also include a Back button. The only option on the Logo window is Next. Scroll down if needed to view the selection options.

## Language Selection

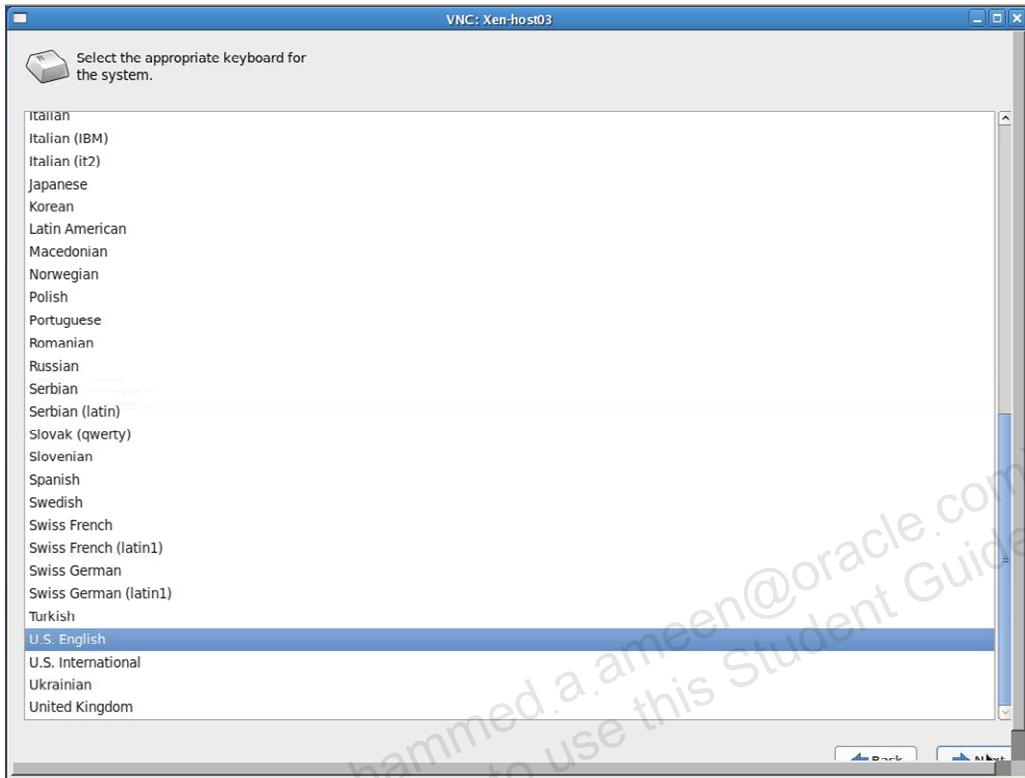


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

ORACLE

The Language Selection window appears. Select the language to use for the installation. The language becomes the default language for the operating system. The language is also used to target the time zone configuration later during the installation process. Click Next to continue or press F12 if installing in text mode.

## Keyboard Selection

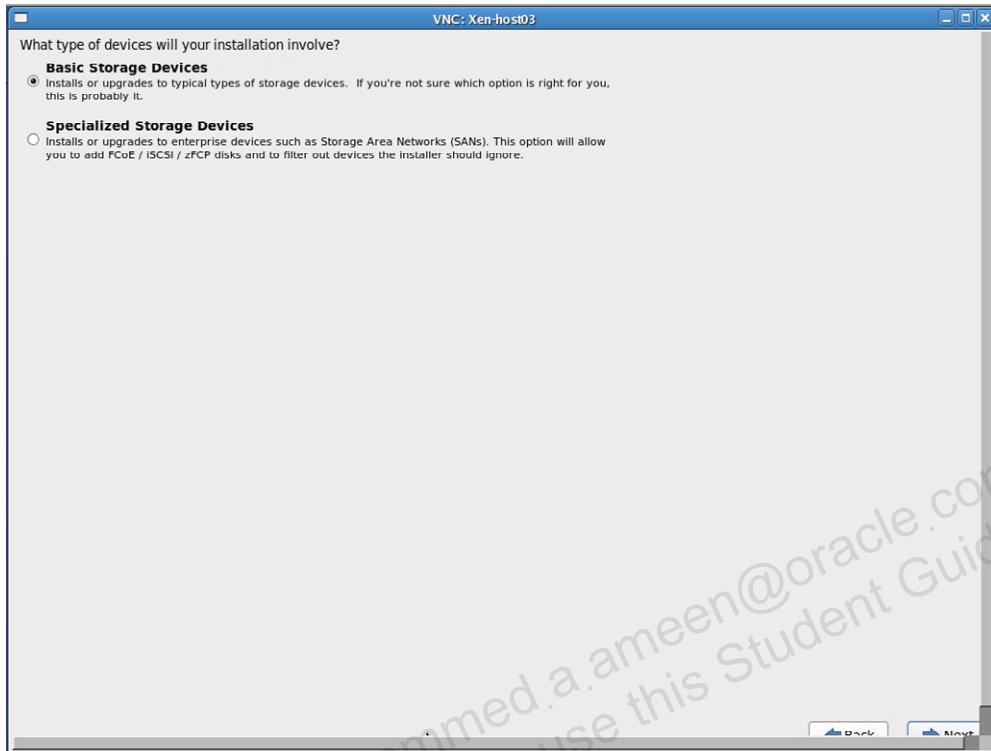


ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Keyboard Selection window appears. Select the keyboard to use for the installation. After the installation is complete, you can use the `system-config-keyboard` command to change the keyboard type. If the X Window System is not installed or not running, a text user interface (TUI) provides a list of keyboard layouts.

# Storage Devices Selection



**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Storage Devices Selection window appears. You can install Oracle Linux on different types of storage devices. Choose Basic Storage Devices to install on local hard drives or solid-state drives. Select Specialized Storage Devices to install on SANs, DASDs, devices attached to a firmware RAID controller, or multipath devices (devices accessible through more than one path).

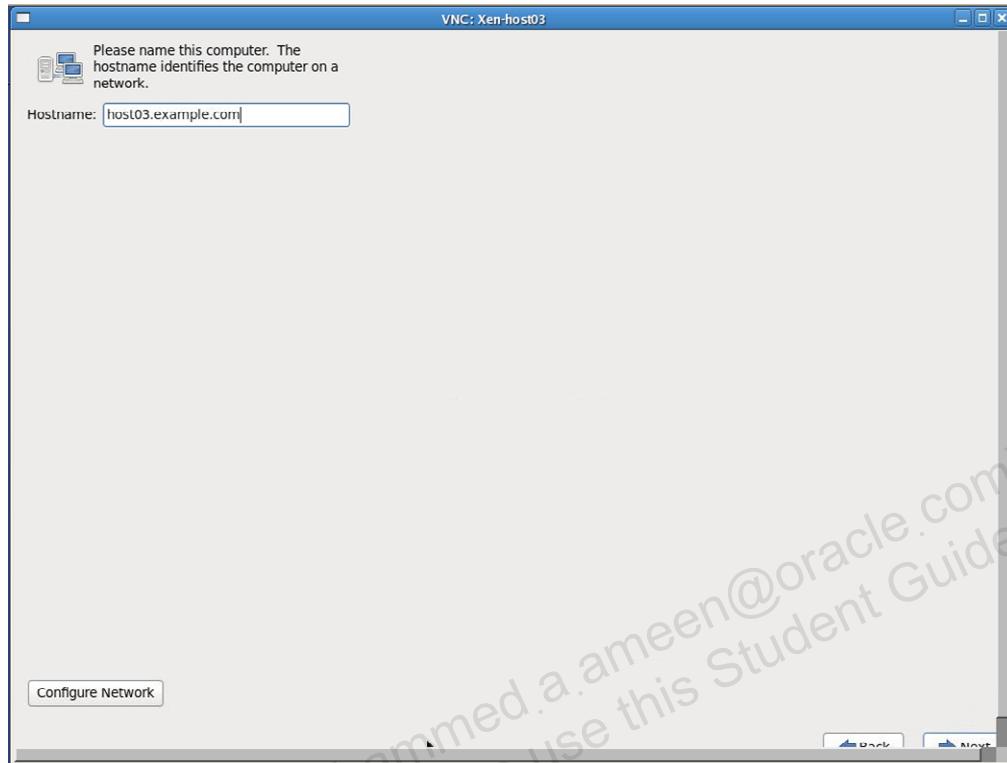
## Basic Storage Devices

This option automatically detects the attached local storage devices. A Storage Device Warning window prompts you to discard data or keep data.

## Specialized Storage Devices

Selecting this option displays a Storage Devices Selection window with tabs for each of the different types of devices. You can select the drives to install the operating system on as well as drives to be mounted automatically. You can also add devices to be mounted after the installation by modifying the `/etc/fstab` file. This window also includes an Add Advanced Target button to configure iSCSI and FCoE connections. You can also filter storage devices by WWID or by port, target, or LUN.

# Setting the Host Name



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Set Hostname window appears. Enter the host name either as a fully-qualified domain name (FQDN) in the format *hostname.domainname* (for example, `host03.example.com`) or as a short host name in the format *hostname*. When using DHCP (Dynamic Host Configuration Protocol), enter the short name to allow the DHCP service to assign the domain name.

## Configure Network

At the bottom left of this window is a Configure Network button. You need to configure a network only if network access is required when your system boots for the first time. Otherwise, you can configure the network after installation is complete.

# Configuring Network



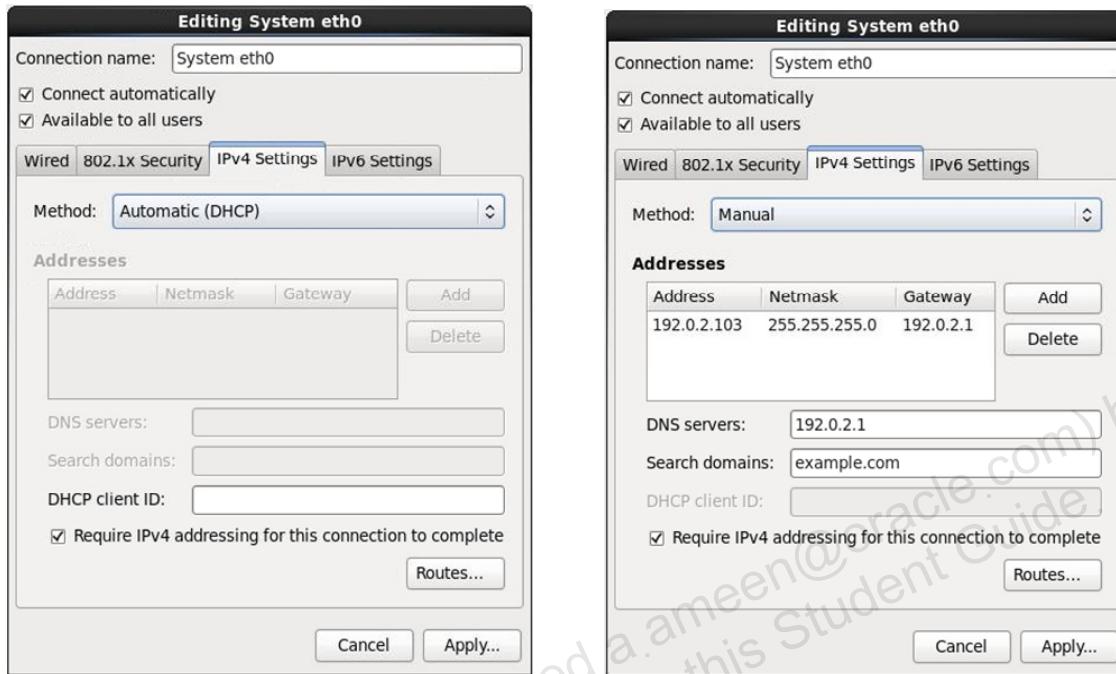
ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Network Connections window appears if you clicked the Configure Network button in the previous window. Wired connections identified by Anaconda are displayed.

This example displays a system with two wired Ethernet connections, `eth0` and `eth1`. You can add and delete connections, or edit existing connections. Select a connection and click Edit to configure an existing connection.

# IPv4 Settings



ORACLE®

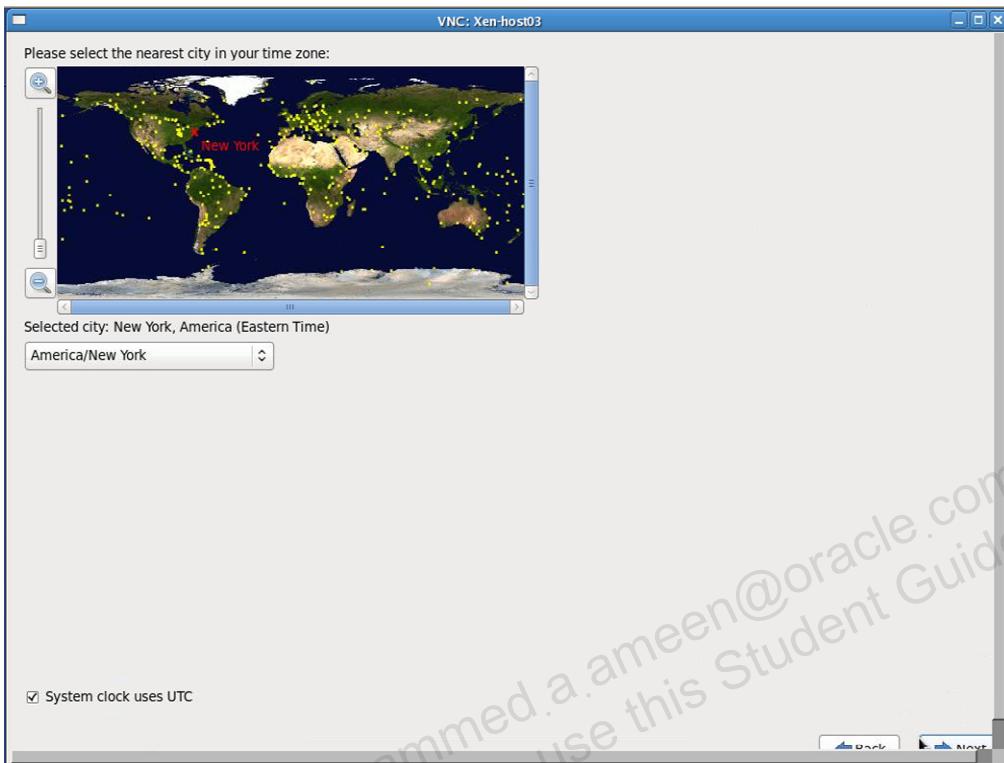
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The slide shows two examples of IPv4 settings. The Wired tab allows you to specify the MAC address and the MTU for the network adapter. The 802.1x Security tab allows you to configure 802.1X security for a connection. You can select the authentication method, either transport layer security (TLS), Flexible Authentication via Secure Tunneling (FAST), Tunneled TLS, or protected extensible authentication protocol (EAP). For each authentication method, you can provide the corresponding information required by the selected method. The IPv4 Settings and IPv6 Settings tabs allow you to configure IP settings for each version of the protocol.

The `eth0` adapter shown has “Connect automatically” enabled. This automatically starts the connection when the system boots. Automatic (DHCP) method configures IPv4 parameters for this adapter by the DHCP service.

The manual method allows you to provide the IP address, netmask, gateway, and Domain Name System (DNS) parameters.

## Time Zone Selection



ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

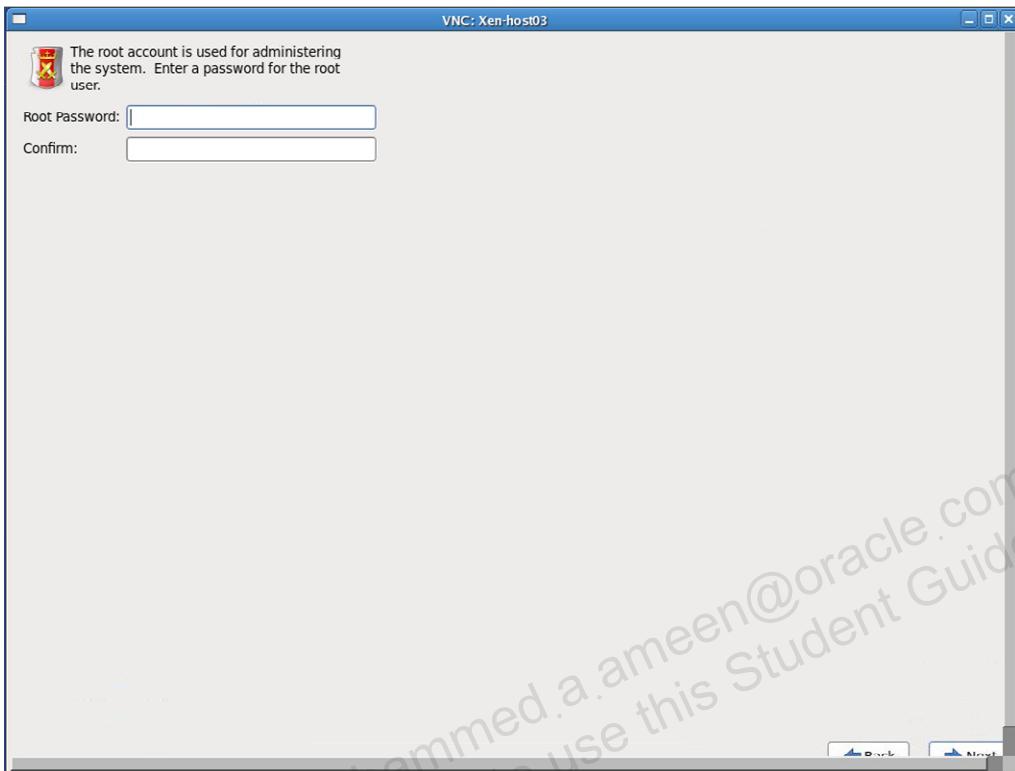
The Time Zone Selection window appears. Select the appropriate time zone by selecting the country and city closest to the location of your system. Either click the map or select from the drop-down list. Specify a time zone even if you are using Network Time Protocol (NTP). NTP is used to synchronize the clocks of computers to a time reference.

At the bottom of this window is a check box to enable “System clock uses UTC.” UTC stands for the Universal Time, Coordinated—also known as Greenwich Mean Time (GMT). Other time zones are determined by adding or subtracting from UTC time. Oracle Linux uses the time zone setting to determine the offset between the local time and UTC on the system clock.

The default is to enable “System clock uses UTC” and should only be deselected if you are running both Oracle Linux and Microsoft operating systems in a dual boot environment. Microsoft operating systems change the BIOS clock to match local time rather than UTC, which might cause unexpected behavior in Oracle Linux.

After the installation is complete, you can use the `system-config-date` command to change the time zone configuration.

# Setting Root Password

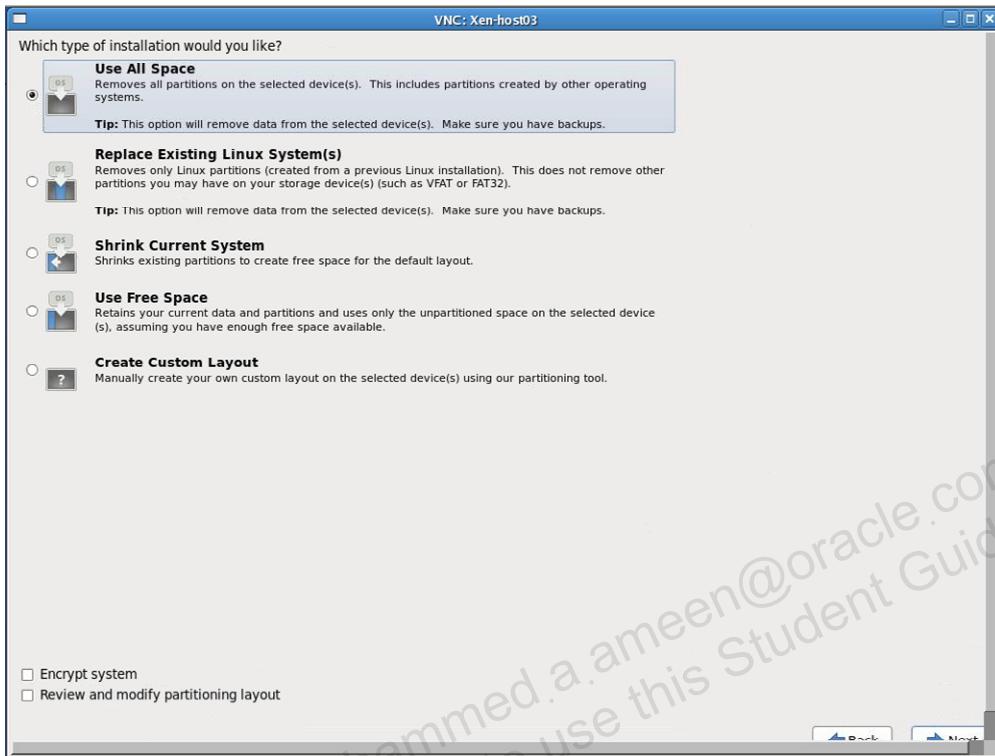


ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Set Root Password window appears. You must enter a password for the root user account. Press Tab and enter the same password in the Confirm box. Click Next. A Weak Password message may appear; however, you have the option to select Use Anyway.

# Disk Partitioning Setup



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Disk Partitioning Setup window appears; it provides disk partitioning options. Hard disks can be divided into one or more logical disks called partitions. The first four options create a default partition layout. The last option, Create Custom Layout, allows you to manually partition your storage devices. At the bottom of this window are buttons to “Encrypt system” and to “Review and modify partitioning layout.” You are prompted for a passphrase if you choose to encrypt the disk partitions.

## Use All Space

This option removes all existing partitions and all existing data.

## Replace Existing Linux System(s)

This option removes partitions created by a previous Linux installation. Other partitions and associated data are preserved.

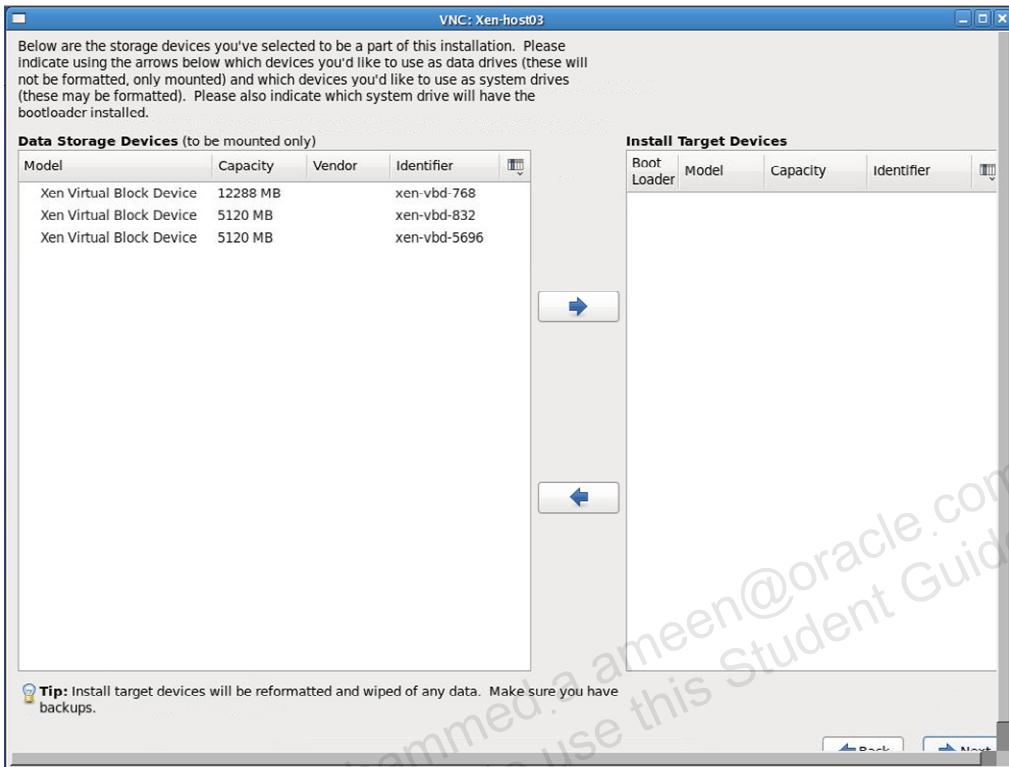
## Shrink Current System

This option allows you to manually resize existing partitions and create a default partition layout for Oracle Linux in the unused space.

## Use Free Space

This option retains existing partitions and installs Oracle Linux in the unused space.

# Storage Devices



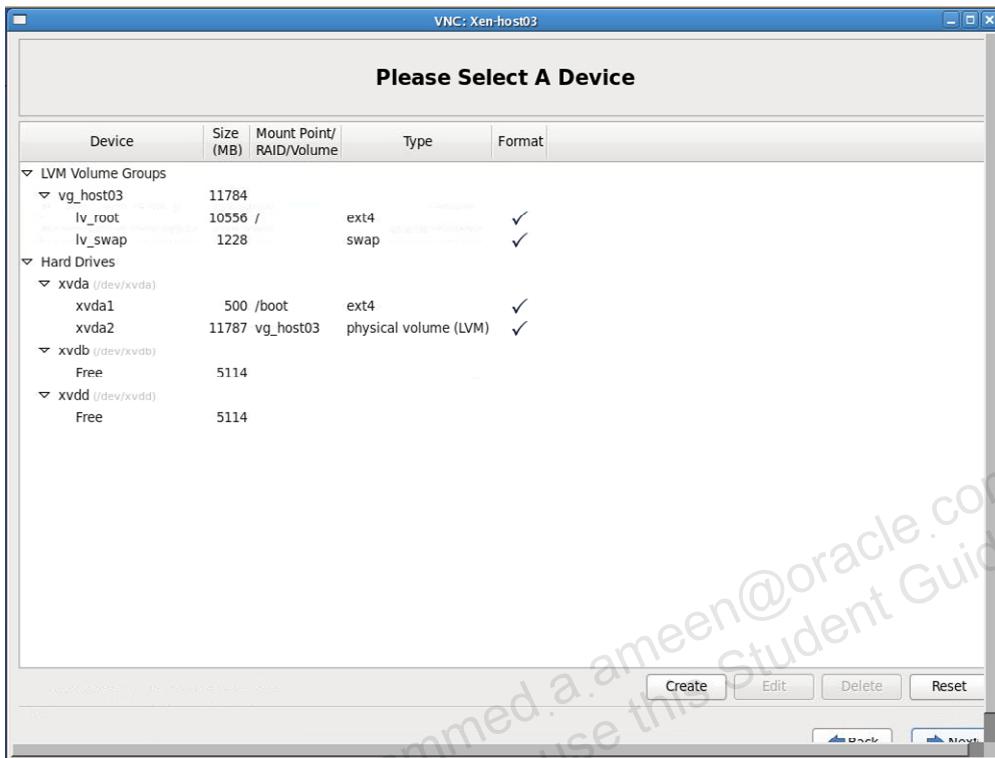
ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This Storage Devices window only appears if more than one storage device is detected. If you have only one storage device, Anaconda does not display this window.

This example shows three virtual disks. Devices on the left are not reformatted, only mounted. Devices on the right are available for installation of the operating system. Select the disks and click the right or left arrows to move them from one side to the other. Devices available as installation targets on the right also include an option button to specify the boot device for the system.

## Default Partition Layout



**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Default Partition Layout window appears if you created a default partition layout and selected “Review and modify partitioning layout.” In this example, two virtual storage devices exist.

Anaconda creates an LVM physical volume by default, with one LVM Volume Group (named `vg_host03` in this example). Two logical volumes are created for this volume group:

- 10556 MB logical volume named `lv_root` for the root partition (/)
- 1228 MB logical volume named `lv_swap` for swap

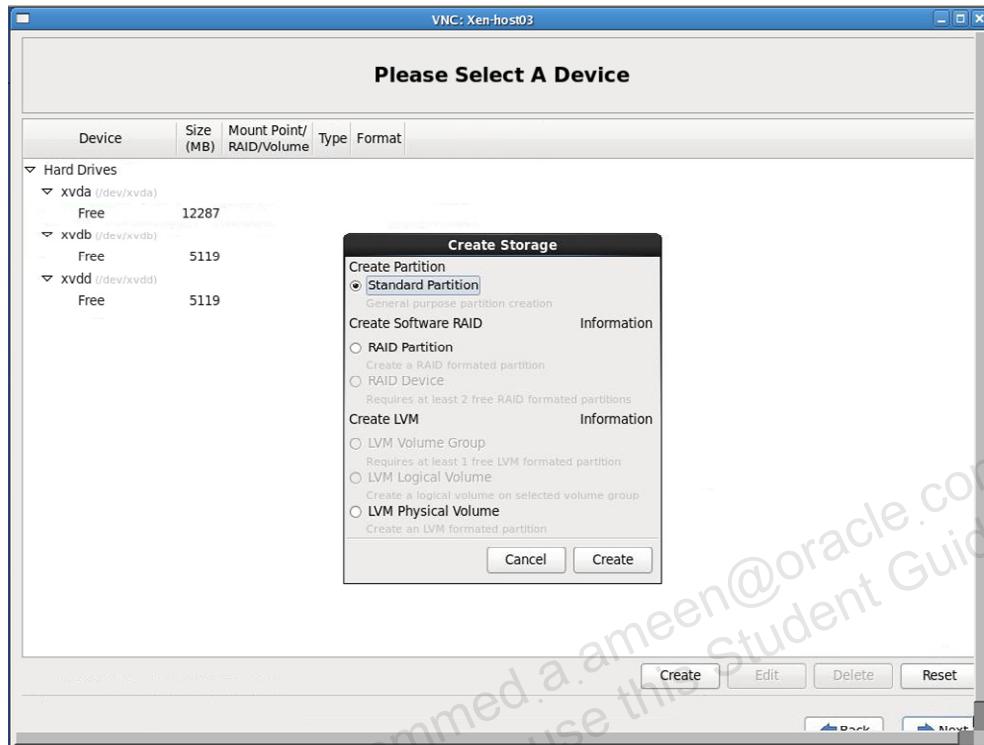
For the first disk (`xvda`), two partitions are created:

- 500 MB partition for `/boot`
- All remaining space as an LVM physical volume

The second disk (`xvdb`) and third disk (`xvdd`) are free to be used as needed.

You can re-partition the storage device by using the Edit, Delete, or Create buttons located at the bottom right of this window.

# Creating a Custom Layout



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Create Custom Layout window appears if you chose to create a custom layout when installing Oracle Linux. Click the Create button to display the Create Storage dialog box. From here you can choose:

## Standard Partition

This option creates a standard disk partition from free space.

## RAID Partition

This option creates a partition from free space to form part of a software RAID device.

## RAID Device

This option combines two or more RAID partitions into a RAID device.

## LVM Physical Volume

This option creates a physical volume from free space.

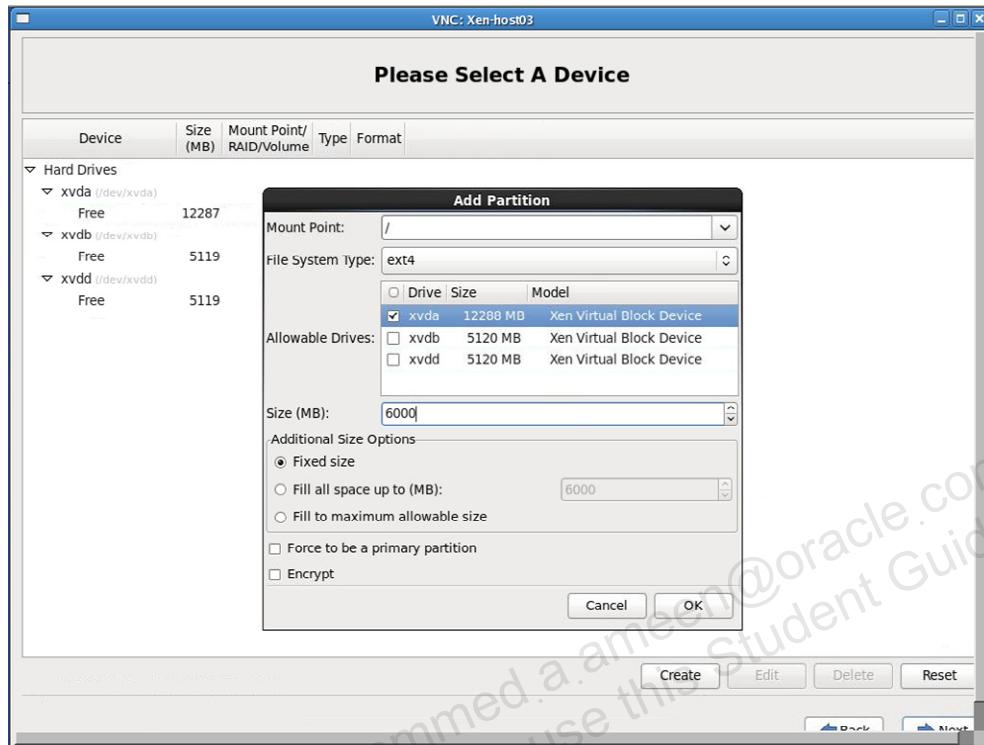
## LVM Volume Group

This option creates a volume group from one or more physical volumes.

## LVM Logical Volume

This option creates a logical volume on a volume group.

# Standard Partition



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Add Partition dialog box is displayed when choosing to create standard partitions. The following information is defined for each partition:

## Mount Point

This is the partition's mount point. Either enter the mount point (for example `/`, `/boot`, `/home`) or select the mount point from the drop-down list. No mount point is defined for the swap partition.

## File System Type

Select the file system and enter either `ext2`, `ext3`, `ext4`, LVM, RAID, swap, or `vfat`.

## Allowable Drives

This field lists the hard disks on your system.

## Size (MB)

This field defines the size (in megabytes) of the partition.

## Additional Size Options

Choose to keep the partition at a fixed size, allow the partition to grow up to a designated size, or use all remaining space on the device (fill to maximum allowable size).

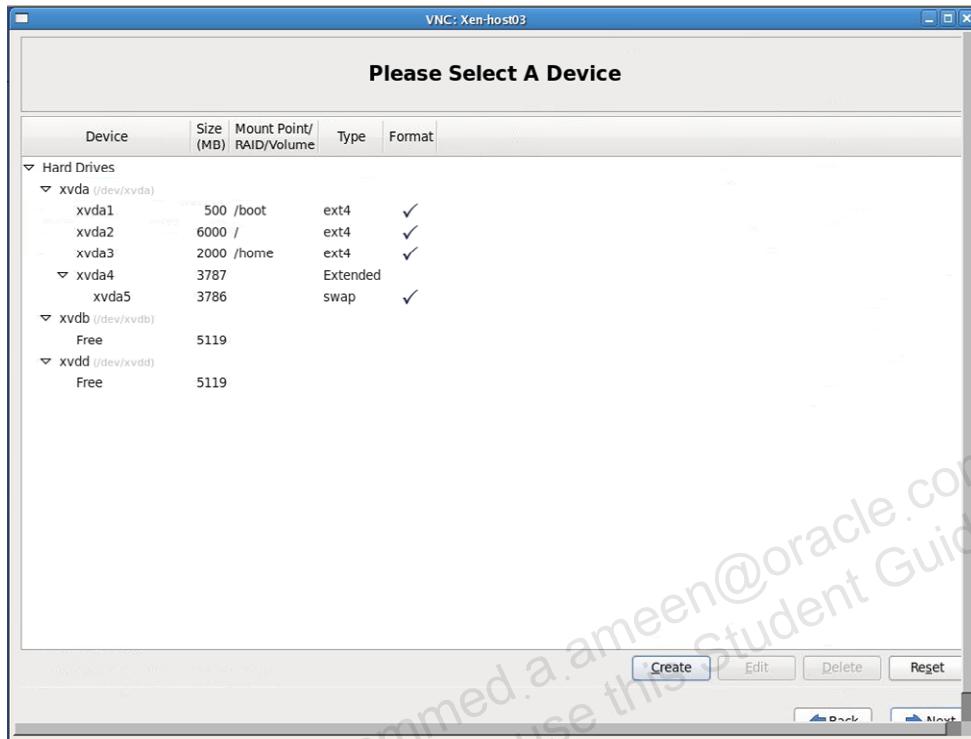
### **Force to be a primary partition**

Select to make the partition one of the first four partitions on the hard drive.

### **Encrypt**

Select to encrypt the partition. If encrypted, data on the partition cannot be accessed without a passphrase.

# Confirming Partitions



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This window shows an example of the created standard partitions.

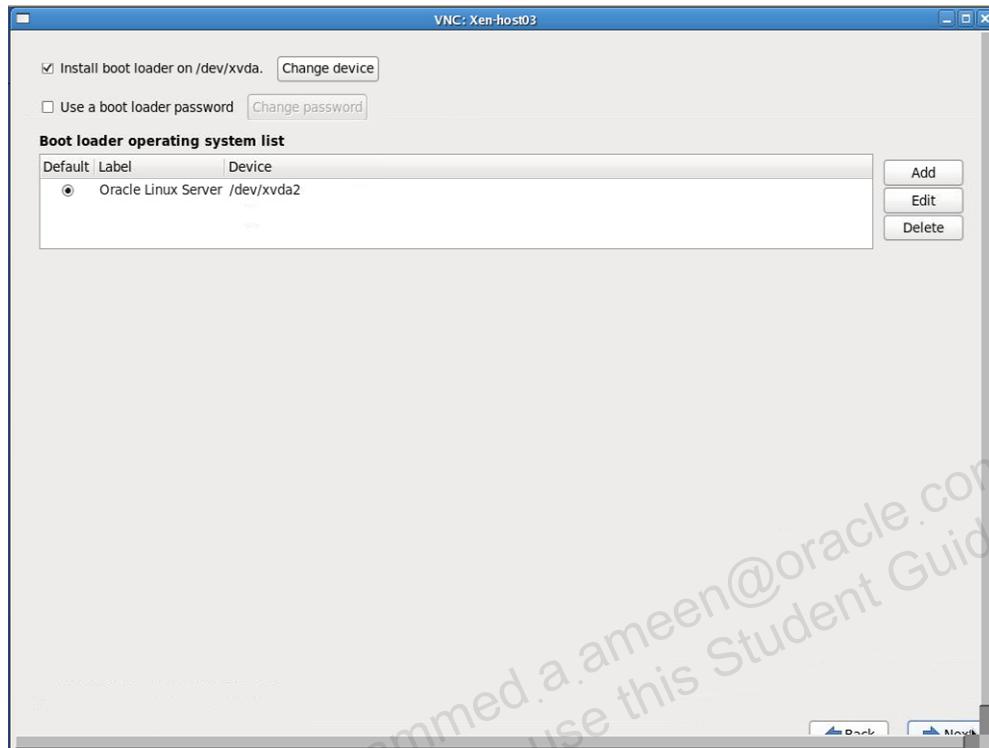
## Format Warnings

When you have created the partitions and clicked Next to continue, a warning appears indicating, "The following pre-existing devices have been selected to be formatted, destroying all data". Click Format to continue.

## Writing Storage Configuration to Disk

A confirmation message then appears indicating, "The partitioning options you have selected will now be written to disk. Any data on deleted or reformatted partitions will be lost." Click "Write changes to disk" to continue.

## Boot Loader Selection



ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Boot Loader Selection window appears. You need to install a boot loader to boot the system without the need for boot media. The boot loader runs when a computer starts and loads the kernel. Oracle Linux uses GRand Unified Bootloader (GRUB), which is installed by default on the Master Boot Record (MBR).

You also have the option to create a boot loader password. A boot loader password is not required but is recommended to enhance system security. Without a password, users can pass options to the kernel allowing them to access the system as the root user without providing the root password.

Oracle Linux attempts to detect previously installed operating systems and configures GRUB to boot them. However, the Add button in this window allows you to manually configure additional operating systems. Each operating system becomes a bootable entry in GRUB's configuration file.

## Quiz

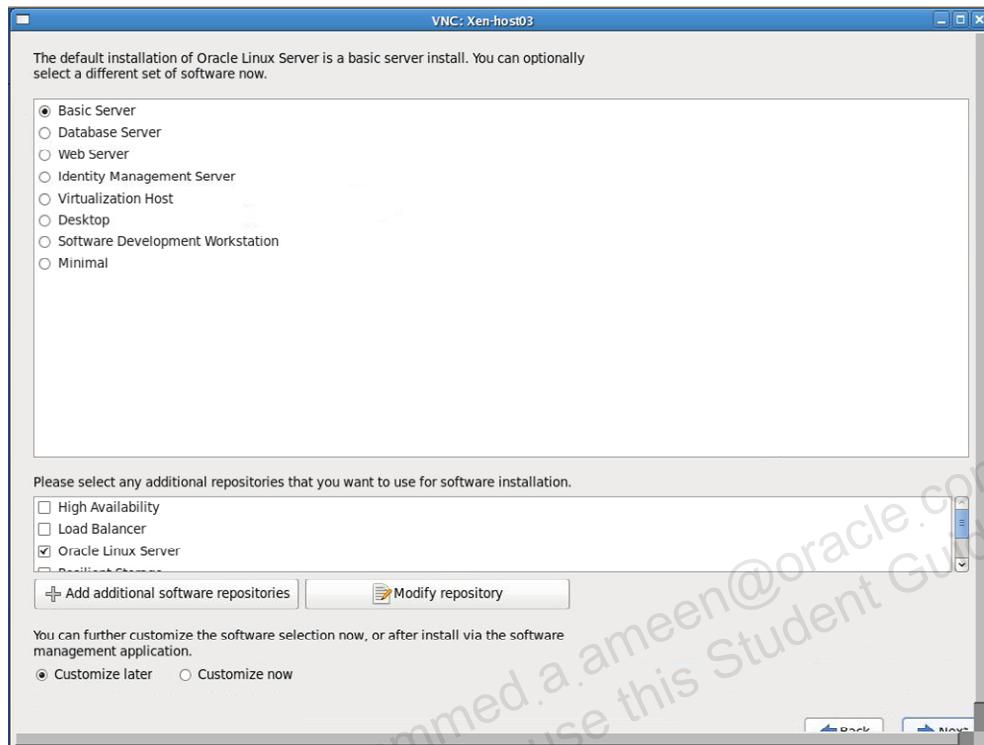
When creating a custom partition layout during the installation of Oracle Linux, which of the following options is available?

- a. Standard Partition
- b. RAID Partition
- c. LVM Physical Volume
- d. All of the above



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Software Package Selection



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The installation process provides the opportunity to select the software packages that you want to install. Categories are provided, each representing different usages of the system. Each category defines the software packages to install. Further customization of the package groups can also be performed. Each of the categories are described as follows:

## **Basic Server**

This option installs software required for a basic server and is selected by default. Basic Server does not include a graphical user environment.

## **Database Server**

This option installs MySQL and PostgreSQL databases.

## **Web Server**

This option installs the Apache web server.

## **Identity Management Server**

This option installs OpenLDAP and the System Security Services Daemon (SSSD).

## **Virtualization Host**

This option installs KVM and Virtual Machine Manager tools.

## **Desktop**

This option installs OpenOffice products, graphical environment and tools, and multimedia applications.

## **Software Development Workstation**

This option installs a compiler and other software development tools.

## **Minimal**

This option installs only the minimum packages necessary to run Oracle Linux.

## **Additional Repositories**

Software repositories provide access to additional software packages. The following repositories are available:

### **High Availability**

This repository includes software packages for clustering.

### **Load Balancer**

This repository includes packages for load-balancing clustering.

### **Oracle Linux Server**

This repository is selected by default. It includes all packages released with the version of Oracle Linux being installed.

### **Resilient Storage**

This repository includes packages for storage clustering.

### **Scalable Filesystem Support**

This repository includes packages to support large file systems.

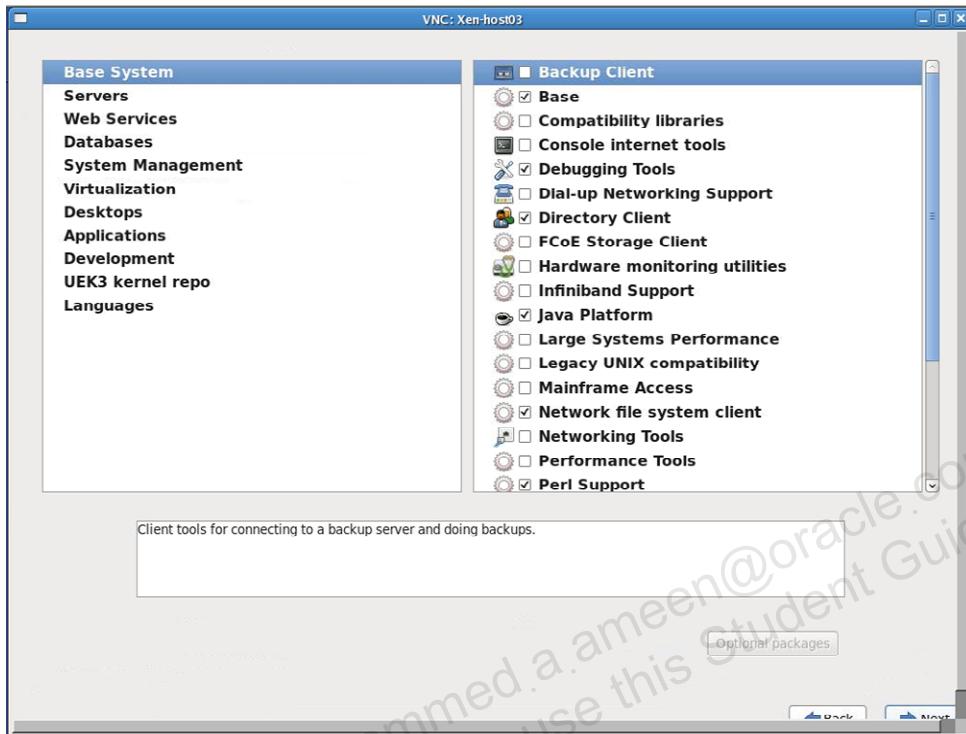
High Availability, Load Balancer, Resilient Storage, and Scalable Filesystem Support are all examples of add-ons for which Red Hat charges extra. See the following for more information: <http://www.redhat.com/products/enterprise-linux-add-ons/>.

These are not supported in Oracle Linux 6 because Oracle does not have access to the source code for the errata stream for each of those features. See the footnote on the following:

<https://linux.oracle.com/supported.html>.

The remaining available repositories can be modified. Additional repositories can be added by providing the repository name and URL.

# Customizing the Package Selection



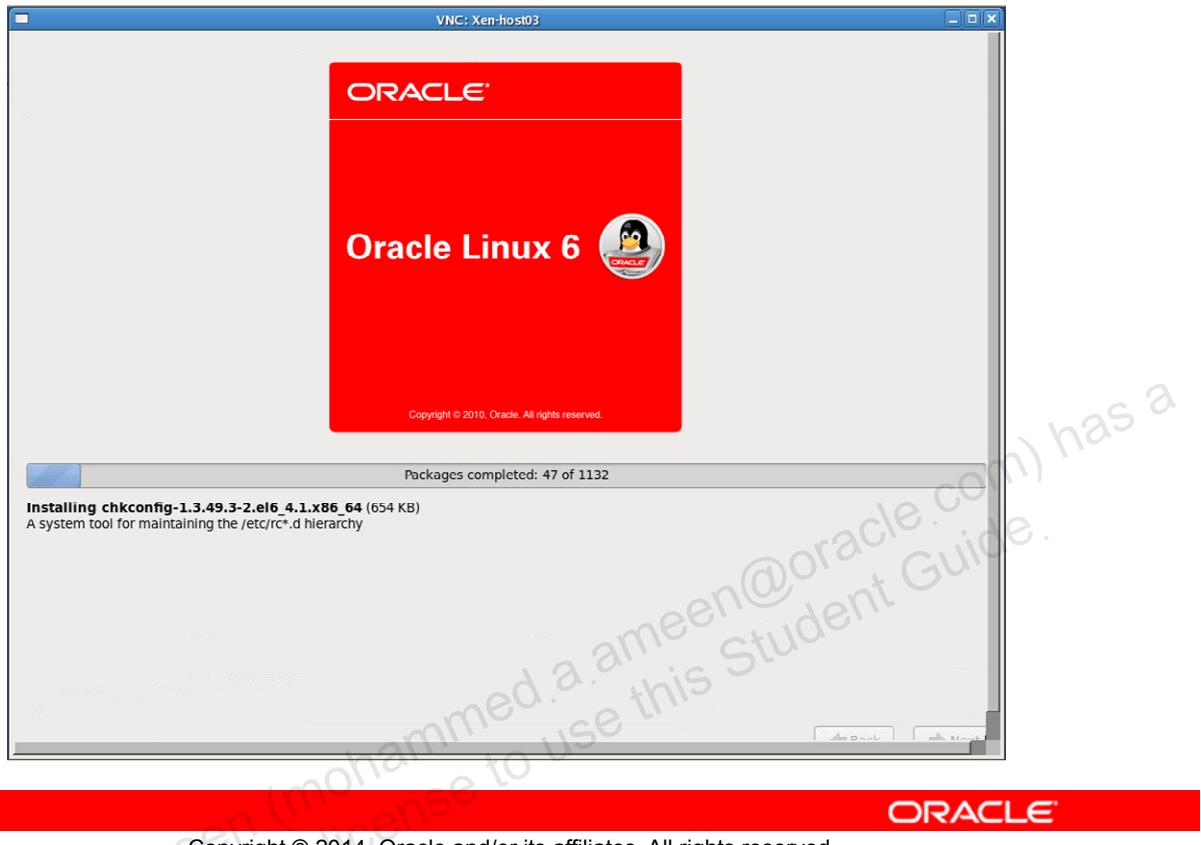
ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

You have the option to customize software provided by the selected package groups and repositories. Click the **Customize Now** button, and then click **Next** to display this window. Categories of packages are listed in the left column. Selecting a category displays the associated package groups in the right column. Select or deselect the individual package groups to indicate whether you want the group to be installed.

Some package groups provide optional packages. These are packages associated with a group, which provide additional functionality but are not required. Click the “Optional packages” button to view and select any optional packages.

# Software Installation



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

ORACLE

This window displays the software packages being installed. A status bar is displayed at the bottom of the window along with the names of the packages being installed.

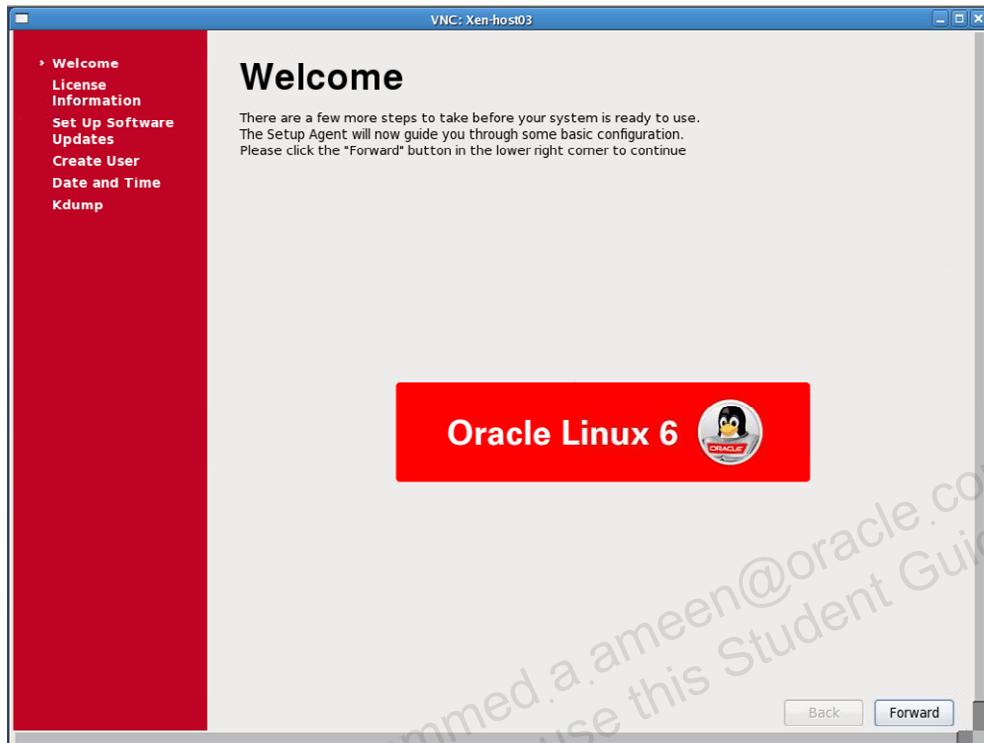
After the packages are installed, a Congratulations window appears along with a button to Reboot the installed system.

## Install Logs

Log files are created during the installation:

- `/root/install.log`: Contains a list of installed software packages
- `/root/install.log.syslog`: Contains syslog messages generated during the installation
- `/var/log`: Contains several Anaconda logs related to the installation

# FirstBoot Tool



ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The FirstBoot window appears if you chose to install the Desktop (graphical) packages. Installing the Desktop system reboots to run level 5 and the FirstBoot tool runs. Installing a Basic Server reboots to run level 3 (no graphics). FirstBoot does not run at run level 3, you are simply presented with a login prompt. The FirstBoot tool runs only after an initial installation and guides you through configuration tasks to perform:

## License Agreement, Software Updates

After the FirstBoot Welcome window, you are asked to agree to the license agreement. You are then prompted to set up software updates. This requires an active network connection to connect to the Unbreakable Linux Network (ULN) and subscribe for software updates.

## Create User, Set Date and Time

You must create a non-administrative user account. Provide username and password at a minimum. You are next prompted to set the current date and time.

## Kdump Setup

Finally, you are given an opportunity to enable the Kdump kernel crash dumping mechanism. If your system crashes, Kdump captures information that assists in determining the cause of the crash.

# Quiz

After installing a Basic Server, the FirstBoot tool runs to perform initial configuration tasks.

- a. True
- b. False

## Summary

In this lesson, you should have learned how to:

- Obtain Oracle Linux operating system software
- Describe the Anaconda installer
- Install Oracle Linux
- Describe the FirstBoot utility

## Practice 3: Overview

The practices cover the following topics:

- Installing Oracle Linux
- Using FirstBoot
- Logging in to Oracle Linux and shutting down
- Re-creating the host03 VM guest

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

## Linux Boot Process



ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe the Linux boot process
- Describe and configure the GRUB bootloader
- Describe and configure kernel boot parameters
- Describe the System V /sbin/init process
- Describe run levels and run level scripts
- Describe the /etc/rc.d directory
- Configure services
- Describe the xinetd superserver
- Describe Upstart

## Linux Boot Process

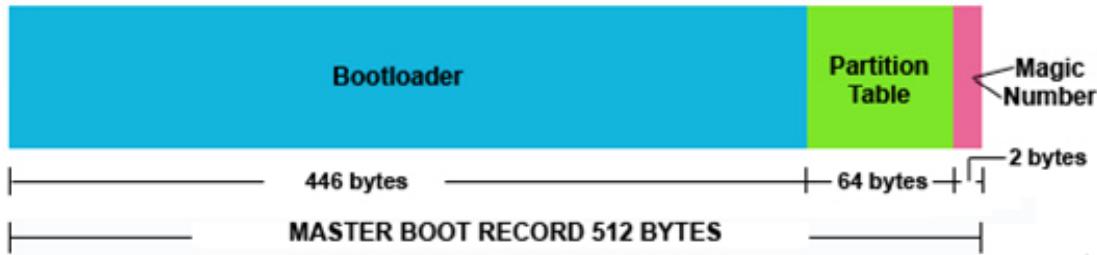
1. The computer's BIOS performs POST.
2. BIOS reads the MBR for the bootloader.
3. GRUB bootloader loads the Linux kernel.
4. Kernel initializes and configures the hardware.
5. Kernel loads modules in the `initramfs` image.
6. Kernel starts the system's first process, `/sbin/init`.
7. `/sbin/init` takes over. It:
  - A. Reads `/etc/inittab`
  - B. Executes `/etc/rc.d/rc.sysinit`
  - C. Boots into run level defined in `/etc/inittab`
  - D. Executes `/etc/rc.d/rc.local`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

It is important to understand the Linux boot process in order to troubleshoot boot problems. These are the high-level steps in the boot process. You also need to be aware of files involved in the boot process, such as `/boot/grub/grub.conf` and `/etc/inittab`, because errors in these files can cause boot problems.

# Master Boot Record (MBR)



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

On a computer with x86 architecture, the Master Boot Record (MBR) is the first 512 bytes of the boot drive that is read into memory by the BIOS. The first 446 bytes of that 512 contain low-level boot code that points to a bootloader somewhere else on the disk, or on another disk. The next 64 bytes contain the partition table for the disk. The last two bytes are the "Magic Number," which is used for error detection.

## Bootloader

The bootloader software runs when a computer starts. It is responsible for loading and transferring control to the kernel. The kernel then initializes the rest of the operating system.

Many bootloaders are available. The most common bootloaders for Linux are LILO (LInux LOader) and GRUB (GRand Unified Bootloader). Oracle Linux uses the GRUB bootloader.

# GRUB Bootloader

- Oracle Linux uses the GRUB bootloader.
  - A subset of the GRUB bootloader code is written to the MBR.
  - The remainder is written to the /boot partition.
- GRUB bootloader is modular and works in stages:
  - Stage1, Stage1\_5, and Stage 2
- Stage 2 is the main GRUB image.
- Stage 2 reads the /boot/grub/grub.conf file.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

GRUB is a product of the GNU project. GRUB conforms to the multiboot specification, which allows it to load many free operating systems directly as well as to chain-load proprietary operating systems. GRUB understands file systems and kernel executable formats, allowing it to load an arbitrary operating system without recording the physical position of the kernel on the disk. The kernel can load just by specifying its file name and the drive and partition where the kernel resides. This information is passed to GRUB through the /boot/grub/grub.conf configuration file, a menu interface, or from the command line.

A subset of the GRUB bootloader code is written to the MBR, and the remainder is written to the /boot partition. In addition, the GRUB bootloader is modular in design and works in stages.

## Stage 1

The stage1 code of GRUB is in the MBR. GRUB stage1 usually points to the next stage of GRUB, stage1\_5 or stage2. GRUB may or may not need to load stage1\_5 depending on the types of file systems present.

```
# file /boot/grub/stage1
stage1: x86 boot sector; GRand Unified Bootloader, stage1 version
0x3, GRUB version 0.94, code offset 0x48
```

## Stage 1\_5

Stage1\_5 deals with specific types of file systems, and is named after them. This code allows the file system to be interpreted appropriately.

```
# ls /boot/grub/stage1/*stage1_5
e2fs_stage1_5    iso9660_stage1_5   reiserfs_stage1_5   xfs_stage1_5
fat_stage1_5     jfs_stage1_5       ufs2_stage1_5      ffs_stage1_5
Minix_stage1_5   vstafs_stage1_5
```

## Stage 2

This is the main GRUB image and resides in the /boot partition at /boot/grub/stage2.

```
# file /boot/grub/stage2
stage2: GRand Unified Bootloader version 3.2, installed partition
65535, identifier 0x0, GRUB version 0.97, configuration file
(hd0,0)/grub/grub.conf
```

It reads the /boot/grub/grub.conf file for configuration information that details how it loads the kernel. You have the option to display a menu of choices generated by grub.conf. You can then:

- Modify the menu
- Choose which operating system to boot
- Take no action, and so allow GRUB to boot the default system

# GRUB Configuration File



The screenshot shows a terminal window titled "root@host03:~". The window displays the contents of the GRUB configuration file, /boot/grub/grub.conf. The file includes directives like default=0, timeout=5, splashimage=(hd0,0)/grub/splash.xpm.gz, and hiddenmenu. It lists two titles: "Oracle Linux Server Unbreakable Enterprise Kernel (3.8.13-16.2.1.el6uek.x86\_64)" and "Oracle Linux Server Red Hat Compatible Kernel (2.6.32-431.el6.x86\_64)". Each title is associated with a root specification, kernel, and initrd.

```
# NOTICE: You have a /boot partition. This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/xvda2
#          initrd /initrd-[generic-]version.img
#boot=/dev/xvda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Oracle Linux Server Unbreakable Enterprise Kernel (3.8.13-16.2.1.el6uek.x86_64)
    root (hd0,0)
    kernel /vmlinuz-3.8.13-16.2.1.el6uek.x86_64 ro root=UUID=4751207b-4fde-4e34-87c8-cb7117848f03 rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
    initrd /initramfs-3.8.13-16.2.1.el6uek.x86_64.img
title Oracle Linux Server Red Hat Compatible Kernel (2.6.32-431.el6.x86_64)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-431.el6.x86_64 ro root=UUID=4751207b-4fde-4e34-87c8-cb7117848f03 rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
    initrd /initramfs-2.6.32-431.el6.x86_64.img
[root@host03 ~]#
```

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The GRUB configuration file, /boot/grub/grub.conf, begins with directive lines: default, timeout, splashimage, and hiddenmenu. The title lines represent installed bootable kernels. Each titled entry specifies associated root, kernel, and initrd directives.

## default

GRUB counts the titled, bootable kernels starting with 0. The line default=0 indicates that GRUB will boot, by default, the first kernel entry.

The default kernel since Oracle Linux 5.6 is the Unbreakable Enterprise Kernel, the first kernel entry. Editing /boot/grub/grub.conf and changing the default=0 directive to default=1 would boot the second kernel entry, the Red Hat Compatible Kernel.

## timeout

The timeout=5 directive means to wait five seconds for keyboard input before booting the default kernel. You can press and hold any alphanumeric key within the timeout duration to display the GRUB menu. timeout=0 means to boot the default kernel immediately. A timeout of -1 (or unset) means to wait indefinitely.

**splashimage**

Oracle Linux and Red Hat Linux distributions hide the boot process by using a boot splash screen. The `splashimage=` directive specifies the splash screen to use. This splash screen can be bypassed, and the output from the boot process displayed, by pressing Esc during the boot process.

**hiddenmenu**

The `hiddenmenu` directive tells GRUB not to display the GRUB menu. This directive has no argument.

**title**

Each title line specifies a bootable kernel. Each title has associated `root`, `kernel`, and `initrd` directives, which should always be indented. The title contains a description of the kernel and the kernel version number in parentheses.

As mentioned in the “Introduction to Oracle Linux” lesson, beginning with Oracle Linux 5.5, customers have a choice when it comes to the kernel: Unbreakable Enterprise Kernel or Red Hat Compatible Kernel. These two installed kernels are indicated by the following titles:

- Oracle Linux Server Unbreakable Enterprise Kernel (3.8.13-16.2.1.el6uek.x86\_64)
- Oracle Linux Server Red Hat Compatible Kernel (2.6.32-431.el6.x86\_64)

The first title references the UEK, the second title references the RHCK.

**root**

The `root` directive specifies the root partition. GRUB starts numbering from 0 for drives and partitions. The first BIOS detected drive is `hd0`, the second is `hd1`. Drives can be locally attached or on a SAN; they are still referenced the same way. GRUB uses drive-comma-partition format. The directive `root (hd0, 0)` specifies the first drive and the first partition on that drive. Mapping between BIOS detected drives and Linux device files is stored in `/boot/grub/device.map`.

```
# cat /boot/grub/device.map
# this device map was generated by anaconda
(hd0) /dev/xvda
```

**kernel**

The `kernel` directive specifies the kernel version number to be booted. A separate `/boot` partition was created; therefore, the path to the kernel (as well as to the `initramfs` image) are relative to `/boot`. Complete path names to the two kernels are:

- `/boot/vmlinuz-3.8.13-16.2.1.el6uek.x86_64`
- `/boot/vmlinuz-2.6.32-431.el6.x86_64`

Each kernel is specified with options. For example, `ro root=UUID=...` specifies the UUID of the `/dev/xvda2` partition and tells GRUB that it is to be mounted read-only and that `/ (root)` is mounted on that partition. The `rhgb` (Red Hat graphical boot) parameter generates a graphical display that tells what is happening as the system boots. The `quiet` option produces less debugging output.

## initrd

The initial RAM disk (`initramfs`) is an initial root file system that is mounted before the real root file system. The job of the initial RAM disk image is to preload the block device modules, such as for IDE, SCSI, or RAID, so that the root file system, on which those modules normally reside, can then be accessed and mounted. After the newly-loaded kernel gets far enough in its initialization sequence (disks probed, memory mapped, and so on) it then switches over to using the real root file system as specified by the `root` directive. This contains, among other things, the `/etc/fstab` file identifying the rest of the file systems to be mounted. The `initramfs` is bound to the kernel and the kernel mounts this `initramfs` as part of this two-stage boot process.

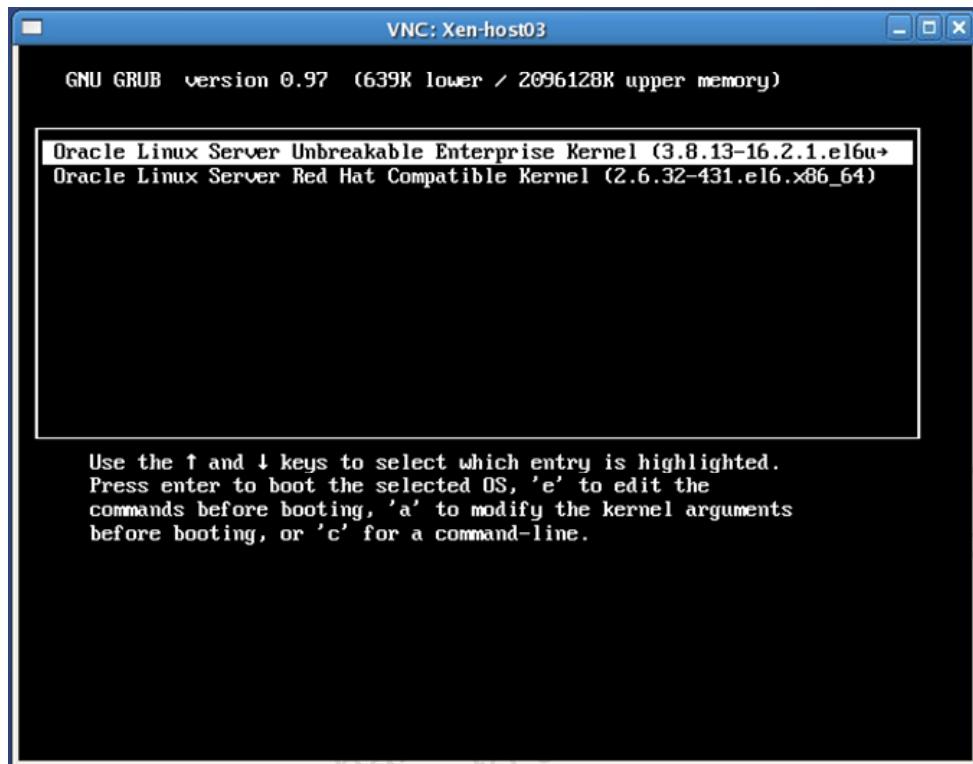
This directive is called `initrd` because the previous tool that created initial RAM disk images, `mkintrd`, created what were known as `initrd` files. In Oracle Linux 6, whenever a new kernel is installed, the `dracut` utility is always called by the installation scripts to create an `initramfs`. The `grub.conf` directive remains `initrd` to maintain compatibility with other tools. You usually do not need to create an `initramfs` manually. This step is automatically performed if the kernel and its associated packages are installed or upgraded from RPM packages distributed by Oracle.

The `initrd` directive must point to the location of the `initramfs` file corresponding to the same kernel version. In other words, the kernel as given on the kernel `/vmlinuz-<kernel_version>` line must match the version number of the `initramfs` image given on the `initrd /initramfs-<kernel_version>.img` line of each stanza.

A separate `/boot` partition was created, therefore the paths to the `initramfs` images are relative to `/boot`. Complete pathnames to the two `initramfs` images are:

- `/boot/initramfs-3.8.13-16.2.1.el6uek.x86_64.img`
- `/boot/initramfs-2.6.32-431.el6.x86_64.img`

## GRUB Menu



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Display the GRUB menu by pressing Esc before the timeout expires. The example in the slide shows that two kernels have been installed:

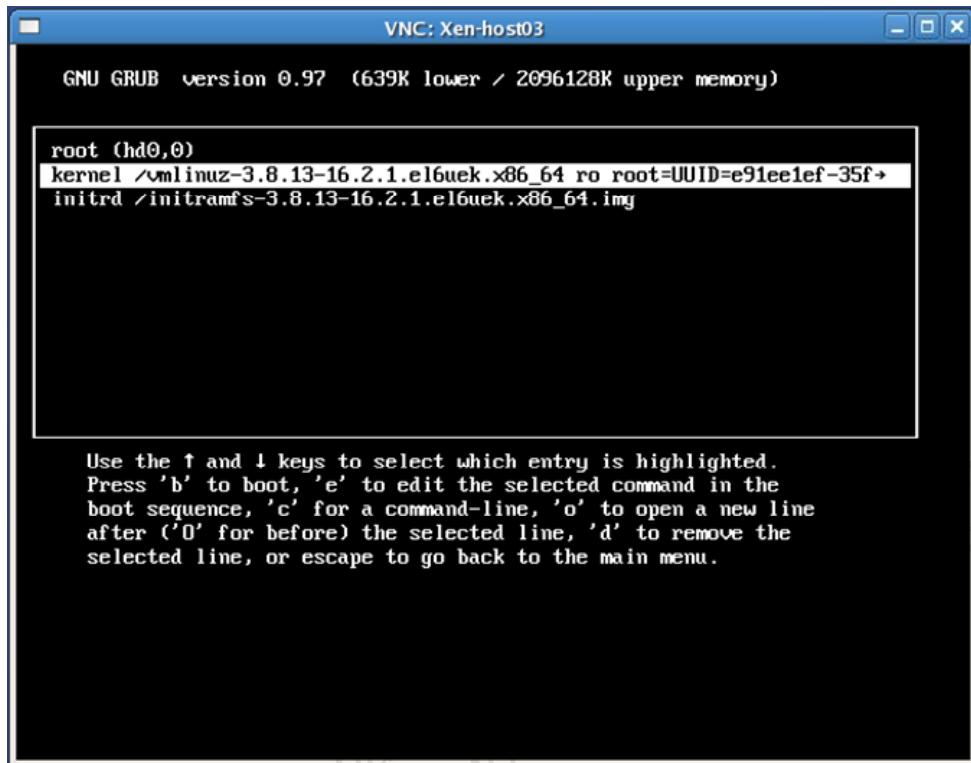
- Oracle Linux Server Unbreakable Enterprise Kernel (3.8.13-16.2.1.el6uek.x86\_64)
- Oracle Linux Server Red Hat Compatible Kernel (2.6.32-431.el6.x86\_64)

These entries correspond to the title line in the /boot/grub/grub.conf file. Editing the title only changes what is displayed on the GRUB menu. The default kernel (Unbreakable Enterprise Kernel) is highlighted.

At the GRUB main menu, a different boot menu item can be selected by using the up arrow and down arrow keys. Kernel parameters can be changed or additional parameters can be passed to the kernel by using the a command.

To modify either the root, kernel, or initrd directives, use the e command. Highlight the kernel you want to modify, and enter e to display an edit menu. From this menu, you can select root, kernel, or initrd for editing.

## Editing a GRUB Menu Option



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Notice that the entries (root, kernel, initrd) in the menu shown in the slide are associated with the title entry in the grub.conf file.

Use the arrow keys to select the particular line that you want to modify, and then enter e again to edit that line. Use the arrow keys to move through the line. The Backspace, Insert, and Delete keys can also be used. Press Enter to save changes, or press Esc to discard changes. Type b to start the boot sequence using the newly-made changes.

All editing changes at boot time are temporary. GRUB does not update the configuration file. For changes to be permanent, edit /boot/grub/grub.conf using a text editor.

Highlighting the kernel and typing e (to edit) displays the kernel entry in GRUB edit mode. The end of the line is displayed initially, but you can use the arrow keys to move the cursor through the kernel line. Moving the cursor to the beginning of the line displays the grub edit> prompt and as much of the kernel line that fits on the screen. Example:

```
grub edit> kernel /vmlinuz-3.8.13-16.2.1.el6uek.x86_64 ro
root=UUID=...->
```

# Kernel Boot Parameters

Kernel Boot Parameters:

- Modify the behavior of the kernel
- Are included in the GRUB configuration file
- Are issued from the GRUB command line
- Are exported to /proc/cmdline



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Linux kernel accepts boot time parameters when it starts to boot the system. Boot parameters serve several purposes including:

- To assist in troubleshooting boot problems
- To provide hardware parameters to the kernel
- To tell the kernel to override the default hardware parameters
- To assist in password and other recovery operations

Parameters can be passed to the kernel from the GRUB prompt, or included with the kernel line entry in /boot/grub/grub.conf. Examples of kernel boot parameters include:

- `ro root=UUID=...`  
Mount the root device read-only on boot and mount root (/) on the specified logical volume.
- `rd_NO_LUKS`  
Disable crypto LUKS detection.
- `rd_NO_LVM`  
Disable LVM detection.

- `rd_NO_DM`  
Disable DM RAID detection.
- `LANG=en_US.UTF-8`  
Specify the system language, written to `/etc/sysconfig/i18n` in the initramfs.
- `SYSFONT=latarcyrheb-sun16`  
Specify the console font, written to `/etc/sysconfig/i18n` in the initramfs.
- `KEYBOARDTYPE=pc`  
Specify the keyboard type, written to `/etc/sysconfig/keyboard` in the initramfs.
- `KEYTABLE=us`  
Specify the keytable file name, written to `/etc/sysconfig/keyboard` in the initramfs.
- `rhgb`  
Enable graphical boot support.
- `quiet`  
Disable most boot messages.

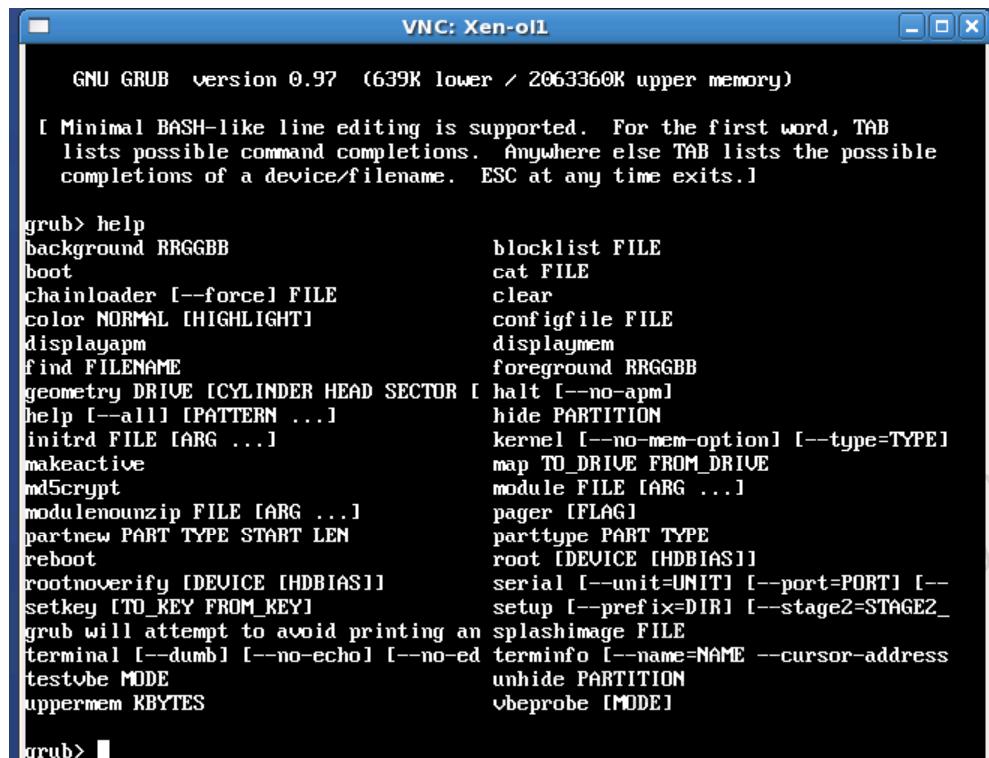
One of the most useful arguments to append to the kernel entry is the word `single`. This instructs `init` to boot the computer in single-user mode, without prompting for the `root` password.

#### **/proc/cmdline File**

Kernel boot parameters are written to the `/proc/cmdline` file and viewable after boot.

```
# cat /proc/cmdline
ro root=UUID=... rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD
SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM
rhgb quiet
```

## GRUB Command Line



```
VNC: Xen-01
GNU GRUB version 0.97 (639K lower / 2063360K upper memory)

[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time exits. ]

grub> help
background RRGGBB          blocklist FILE
boot                           cat FILE
chainloader [--force] FILE   clear
color NORMAL [HIGHLIGHT]    configfile FILE
displayapm                     displaymem
find FILENAME                  foreground RRGGBB
geometry DRIVE [CYLINDER HEAD SECTOR [ halt [--no-apm]
help [--all] [PATTERN ...]   hide PARTITION
initrd FILE [ARG ...]        kernel [--no-mem-option] [--type=TYPE]
makeactive                      map TO_DRIVE FROM_DRIVE
md5crypt                         module FILE [ARG ...]
modulenounzip FILE [ARG ...]  pager [FLAG]
partnew PART TYPE START LEN   parttype PART TYPE
reboot                           root [DEVICE [HDBIAS]]
rootnoverify [DEVICE [HDBIAS]] serial [--unit=UNIT] [--port=PORT] [--setup [--prefix=DIR] [--stage2=STAGE2_
setkey [TO_KEY FROM_KEY]      grub will attempt to avoid printing an splashimage FILE
terminal [--dumb] [--no-echo] [--no-ed terminfo [--name=NAME --cursor-address
testube MODE                   unhide PARTITION
uppermem KBYTES                 vbeprobe [MODE]

grub>
```

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Typing **c** in the GRUB menu brings up the GRUB command line, or shell. You can also access the GRUB command line by executing the **/sbin/grub** command.

The example in the slide shows the output of entering **help** to display available commands. Used by itself, **help** displays a list of the most useful GRUB commands. When used with an argument, it displays a brief synopsis and description of the command specified.

Example:

```
grub> help initrd
initrd: initrd FILE [ARG ...]

Load an initial ramdisk FILE for a Linux format boot image
and set the appropriate parameters in the Linux setup area
in memory.
```

## /sbin/init Process

The /sbin/init process:

- Is the system's first process
- Is the parent of all system processes
- Has a PID of 1
- Reads the /etc/inittab file
- Runs the /etc/rc.d/rc.sysinit script
- Sets the source function library, /etc/rc.d/init.d/functions
- Boots to Linux init run level as described in /etc/inittab

The configuration files for SysV init are in the /etc/rc.d/ directory.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

When the kernel is done with its initialization, it starts the system's first process, /sbin/init, which coordinates the rest of the boot process and is responsible for starting all other processes.

The init process has a process ID (PID) of 1 and is the parent of all system processes. It reads various files and directories before booting into the run level as defined in /etc/inittab.

The init process executes the /etc/rc.d/rc.sysinit script once at boot time. This script sets the host name, initializes the network, mounts the /proc file system, and runs the /etc/rc.d/init.d/functions script, which contains functions used by the scripts in /etc/init.d.

The /etc/rc.d/rc.sysinit script continues by initializing SELinux based on its configuration, prints a text banner, initializes the hardware based on kernel boot arguments, mounts the file systems, cleans up the /var directory, starts swapping, and generally performs the steps required for system initialization.

The init process then executes the scripts in the /etc/rc.d directory to bring up the system to the default run level.

## SysV init Run Levels

- SysV init run levels:
  - 0: Halt
  - 1: Single-user text mode
  - 2: Full multi-user text mode without NFS
  - 3: Full multi-user text mode with NFS
  - 4: Not used (user-definable)
  - 5: Full multi-user graphical mode (X-based login screen)
  - 6: Reboot



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

SysV init run levels provide the ability to use systems for different purposes, and only start the services needed for a specific purpose. For example, a server runs more efficiently at run level 3, because resources are not needed by the X Window System. Diagnostics, backups, and upgrades are best performed at run level 1 when other users cannot log in.

Each run level defines which services are halted and started by `init`. For example, run level 3 starts network services, whereas run level 1 stops these services. Specific services are configured to be started or stopped at a given run level allowing `init` to automatically perform this function. The following run levels are defined by default under Oracle Linux:

- 0 — Halt
- 1 — Single-user text mode
- 2 — Full multi-user text mode without NFS
- 3 — Full multi-user text mode with NFS
- 4 — Not used (user-definable)
- 5 — Full multi-user graphical mode (with an X-based login screen)
- 6 — Reboot

# Working with Run Levels

- To display the current run level:

```
# runlevel  
# who -r
```

- To change the run level:

```
# init 1  
# telinit 3
```

- Use the shutdown command to notify logged-in users of the impending action:

```
# shutdown -r 5  
# shutdown -h now
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Determining the Current Run Level

There are two commands to display the current run level. The first command is `runlevel`, which displays the previous and current run level. If there is no previous run level, the letter N is displayed. The following example indicates that the current run level is 3.

```
# runlevel  
N 3
```

A second command that displays the current run level is `who -r`:

```
# who -r  
run-level 3      2011-10-09 14:58
```

## Changing the Run Level

Either the `init` command or the `telinit` command changes the system run level. The following command stops services needed by the current run level and starts only the services defined by run level 1:

```
# init 1  
output omitted...
```

The `telinit` command works the same way as the `init` command. The following example changes the run level to 5:

```
# telinit 5  
output omitted...
```

## Halt, Reboot, and Shutdown

Run level 0 halts the system. Run level 6 performs a reboot. However, the `shutdown` can also be used to perform both of these functions, and is preferred because this command notifies all logged-in users of the impending action. The following are examples of using the `shutdown` command.

To request that the system be rebooted after it has been brought down:

```
# shutdown -r 5  
The system is going down for reboot in 5 minutes!
```

To request that the system be halted immediately:

```
# shutdown -h now
```

## /etc/inittab File

- The `id:x:initdefault` entry defines the default run level.
  - `x` can be `0,1,2,3,4,5,6` or not specified.
  - If not specified, `id::initdefault`, defaults to run level 3.
- In Oracle Linux 5, entries in `/etc/inittab` control how the `init` process behaves.
  - Oracle Linux 6 uses Upstart
- The format of entries is `id:runlevel:action:process`
- Actions include:
  - `respawn`: Restart the process whenever it terminates.
  - `wait`: Start the process and wait for its termination.
  - `once`: Execute the process once.
  - `boot`: Execute the process during system boot



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The default run level for the system is listed in `/etc/inittab`. To find out the default run level for a system, look for the `initdefault` entry near the top of `/etc/inittab`:

```
id:3:initdefault:
```

The default run level listed in this example is 3, indicated by the number after the first colon. Edit the edit `/etc/inittab` file as `root` to change the default run level.

You can also change the run level at boot time from the GRUB menu. Select and edit the kernel line entry and append the run level at the end of the line. The following example changes the run level of a single boot session to run level 3:

```
<latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us rhgb quiet 3
```

Oracle Linux 6 implements an event-based initialization system named Upstart. However, in Oracle Linux 5, the initialization table, `/etc/inittab`, controls how the `init` process behaves. Each line in this table contains four colon-separated fields:

```
id:runlevel:action:process
```

The entry fields are:

**id (identifier)**

The identifier uniquely defines an entry in `/etc/inittab`.

**runlevel**

The run level is the system run level(s) at which the associated process is executed. The run level consists of zero or more characters chosen from 0123456. For example, if the system is in run level 1, only those entries with a 1 in the `runlevel` field are started. If more than one run level is listed, the associated process is executed at each of the specified run levels. If no run level is specified, `init` executes the process at all run levels.

When `init` is requested to change run levels, all processes without an entry in the `runlevel` field for the target run level receive a warning signal (SIGTERM) to allow them to terminate gracefully. After a 5-second grace period, processes are forcibly terminated by the `kill` signal (SIGKILL).

There are three other values that appear in the `runlevel` field, even though they are not true run levels: a, b, and c. Entries that have these characters in the `runlevel` field are processed only when the `telinit` command requests them to be run (regardless of the current run level of the system). They differ from run levels in that the `init` command can never enter run level a, b, or c. Also, a request for the execution of any of these processes does not change the current run level. Furthermore, a process started by an a, b, or c command is not killed when the `init` command changes run levels. They are killed if the `init` command goes into single-user mode.

**action**

Tells the `init` command how to treat the process specified in the `process` field. The following actions are recognized by the `init` command:

**respawn**

Tells `init` to start the process if it does not exist, but not to wait for it to terminate. If the process does exist, `init` moves on the next entry in `/etc/inittab`. The `init` utility continues to rescan `/etc/inittab`, looking for processes that have died. When a process dies, a respawn entry causes `init` to restart it.

**wait**

When the `init` command enters the run level that matches the entry's run level, start the process and wait for its termination. All subsequent reads of the `/etc/inittab` file while the `init` command is in the same run level cause the `init` command to ignore this entry.

**once**

When the `init` command enters a run level that matches the entry's run level, start the process, and do not wait for its termination. When it dies, do not restart the process. When the system enters a new run level, and the process is still running from a previous run level change, the program is not restarted. All subsequent reads of the `/etc/inittab` file, while the `init` command is in the same run level, cause the `init` command to ignore this entry.

**boot**

Process the entry only during system boot, which is when the `init` command reads the `/etc/inittab` file during system startup. Start the process, do not wait for its termination, and when it dies, do not restart the process. For the instruction to be meaningful, the run level should be the default or it must match the `init` command's run level at boot time. This action is useful for an initialization function following a hardware reboot of the system.

#### **bootwait**

Process the entry the first time that the `init` command goes from single-user to multi-user state after the system is booted. Start the process, wait for its termination, and when it dies, do not restart the process. If the `initdefault` is 2, run the process right after boot.

#### **powerfail**

Execute the process associated with this entry only when the `init` command receives a power fail signal (SIGPWR).

#### **powerwait**

Execute the process associated with this entry only when the `init` command receives a power fail signal (SIGTERM), and wait until it terminates before continuing to process the `/etc/inittab` file.

#### **off**

If the process associated with this entry is currently running, send the warning signal (SIGTERM), and wait 5 seconds before terminating the process with the `kill` signal (SIGKILL). If the process is not running, ignore this entry.

#### **ondemand**

Functionally identical to `respawn`, except this action applies to the `a`, `b`, or `c` values, not to run levels.

#### **initdefault**

An entry with this action is scanned only when the `init` command is initially invoked. The `init` command uses this entry, if it exists, to determine which run level to enter initially. If the `init` command does not find an `initdefault` entry in the `/etc/inittab` file, it requests an initial run level from the user at boot time.

#### **sysinit**

Entries of this type are executed before the `init` command tries to access the console before login. Entries are used only to initialize devices on which the `init` command might try to ask the run level question. These entries are executed and waited for before continuing.

## /etc/rc.d Directory

- The /etc/rc.d/ directory contains configuration files for Linux init.
- The init.d/ directory contains scripts for controlling services.
- rc0.d/, rc1.d/, rc2.d/, rc3.d/, rc4.d/, rc5.d/, and rc6.d/ are run level directories that contain symbolic links to scripts in init.d/.
  - The scripts' names begin with K (kill) and S (start).
- Files in /etc/rc.d:
  - rc.sysinit: Runs once at boot time
  - rc: Starts and stops services when the run level changes
  - rc.local: Is the last script that the init program executes



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The configuration files for SysV init are located in the /etc/rc.d/ directory. Within this directory are the rc, rc.local, rc.sysinit scripts as well as the following directories:

init.d/ rc0.d/ rc1.d/ rc2.d/ rc3.d/ rc4.d/ rc5.d/ rc6.d/

The /etc/rc.d directory is a violation of SysV init. UNIX administrators may be used to /etc/rc#.d directories rather than /etc/rc.d/rc#.d directories. Therefore, there are symbolic links to make Linux look like UNIX, for example:

```
# ls -ld /etc/rc?.d
lrwxrwxrwx  /etc/rc0.d -> rc.d/rc0.d
lrwxrwxrwx  /etc/rc1.d -> rc.d/rc1.d
output omitted...
```

The init.d directory contains the scripts used by the /sbin/init command when controlling services. The scripts are used to start and stop various services.

The rc script runs the scripts that start and stop the services for each run level. The init process calls the rc script with an argument specifying the desired run level. When changing from one run level to another, rc also runs the scripts that stop the service currently running and are not required for the new run level.

The actual scripts that start and stop the services are located in the `init.d` directory. The `rc#.d` directories contain symbolic links to the services in `init.d`. Symbolic links are used in each of the `rc` directories so that the run levels can be reconfigured by creating, modifying, and deleting the symbolic links without affecting the actual scripts they reference.

The `rc#.d` directories are numbered to correspond to the run level they represent. For example, `/etc/rc.d/rc3.d` is the directory for run level 3. When booting (or changing) to run level 3, the `init` programs looks in the `/etc/rc.d/rc3.d` directory to determine which processes to start and stop.

The following is an example listing of the `/etc/rc.d/rc3.d` directory:

```
K01certmonger -> ../init.d/certmonger  
K01smartd -> ../init.d/smard  
K15httpd -> ../init.d/httpd  
K60nfs -> ../init.d/nfs  
S01sysstat -> ../init.d/sysstat  
S08iptables -> ../init.d/iptables  
S55sshd -> ../init.d/sshd
```

As illustrated in this listing, none of the scripts that actually start and stop the services are located in the `/etc/rc.d/rc3.d` directory. Rather, all of the files in `/etc/rc.d/rc3.d/` are symbolic links pointing to scripts located in the `/etc/rc.d/init.d` directory.

The name of each symbolic link begins with either a K or an S. The K links are processes that are killed on that run level, whereas those beginning with an S are started. When entering a new run level, each K (kill) script is executed with an argument to stop, and then each S (start) script is executed with an argument of start.

The `init` command first stops all of the K symbolic links in the directory by issuing the `/etc/rc.d/init.d/<command>` stop command, where `<command>` is the process to be killed. It then starts all of the S symbolic links by issuing `/etc/rc.d/init.d/<command>` start. Each of the symbolic links is numbered to dictate start order. The lower the number, the earlier it is started. Symbolic links with the same number are started alphabetically.

To customize system initialization, add shell scripts to the `/etc/rc.d/init.d` directory and place symbolic links to these files in the `/etc/rc.d/rc#.d` directories, with # corresponding to the specific run level(s) in which the script should be started. For information on writing SysV init scripts, see `/usr/share/doc/initscripts*/sysvinitfiles`.

#### **rc.local**

One of the last things the `init` program executes is the `/etc/rc.d/rc.local` script. This script is useful for system customization. Adding commands to the bottom of this script is an easy way to perform necessary tasks such as starting special services or initialize devices without writing complex initialization scripts in the `/etc/rc.d/init.d` directory and creating symbolic links.

## Stopping and Starting Services

- To stop a service, pass the `stop` argument:

```
# /etc/init.d/<service> stop
```

- To start a service, pass the `start` argument:

```
# /etc/init.d/<service> start
```

- To status a service, pass the `status` argument:

```
# /etc/init.d/<service> status
```

- Can also use the `service` command:

```
# service sshd stop  
# service sshd start  
# service sshd status
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

These commands can be executed as `root` from the command line to start and stop services. For example, to stop the SSH service, pass the `stop` argument:

```
# /etc/init.d/sshd stop  
Stopping sshd: [ OK ]
```

To start the SSH service, pass the `start` argument:

```
# /etc/init.d/sshd start  
Starting sshd: [ OK ]
```

Some scripts also take other arguments, such as `restart`, `reload`, and `status`. Run a script without an argument to display a usage message indicating which arguments it accepts:

```
# /etc/init.d/sshd  
Usage: /etc/init.d/sshd {start|stop|restart|reload|force-  
reload|condrestart|try-restart|status}
```

The `service` program can also be used to run a SysV init script, for example:

```
# service sshd status  
openssh-daemon (pid 1393) is running...
```

# Configuring Services

chkconfig:

- Maintains the /etc/rc.d directory hierarchy
- Changes only the configuration
- Does not change the current state of service



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `chkconfig` command provides a way for you to maintain the /etc/rc.d directory hierarchy. It relieves the task of directly manipulating the symbolic links in those directories. This command can add, remove, list startup information, and check the state of system services. It changes the configuration only; it does not change the state of any service.

To see a list of all services, use the following command:

```
# chkconfig --list
abrt 0:off 1:off 2:off 3:on 4:off 5:on 6:off
acpid 0:off 1:off 2:on 3:on 4:on 5:on 6:off
atd 0:off 1:off 2:off 3:on 4:on 5:on 6:off
output omitted...
```

All services that run their own daemons are listed, followed by their configured state for each run level. For example, the `abrt` service is configured to run at run level 3, and not to run at the remaining run levels. Include the daemon name to only check how it is configured, for example:

```
# chkconfig --list sshd
sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

The `chkconfig` command can also be used to enable or disable services for specific run levels. The following command enables the `httpd` service for run levels 2, 3, 4, and 5:

```
# chkconfig httpd on
```

To enable the service for specific run levels, add the `--level` option followed by the string of numbers from 0 to 6 representing each run level in which you want the service to run. For example, to enable the `abrt` service for run levels 3 and 5, enter:

```
# chkconfig --level 35 abrt on
```

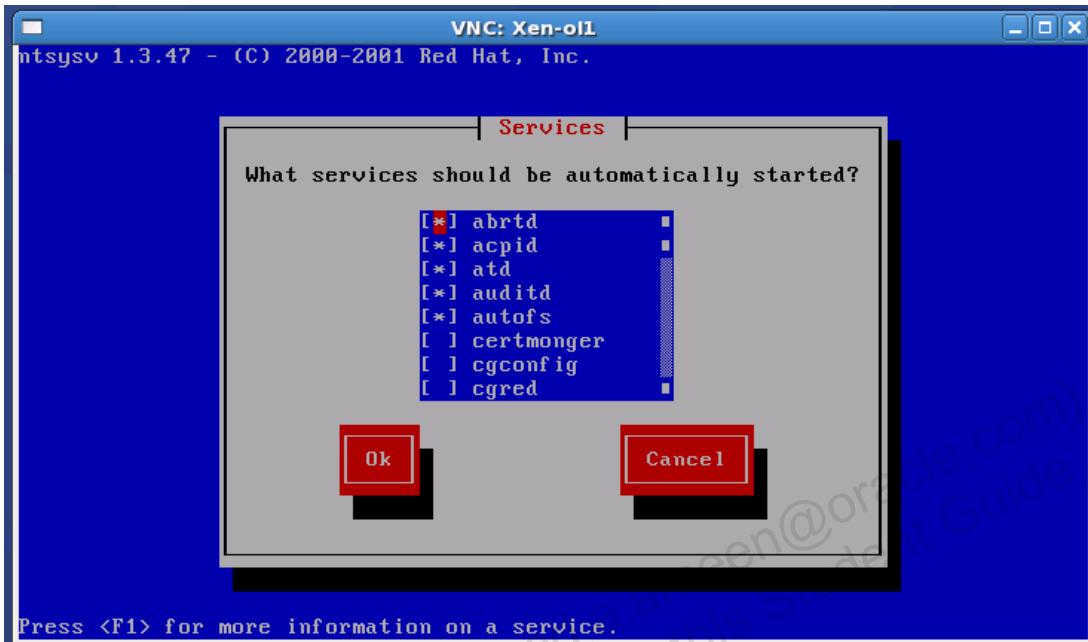
Use the `off` argument to disable a service. For example, the following command disables the `sshd` service for run levels 2, 3, 4, and 5:

```
# chkconfig sshd off
```

As mentioned, the `chkconfig` command changes only the configuration. It does not change the current state of any service. The previous command changed the configuration for the `sshd` service to be off; however, if the service was running, it will still be running. When you reboot the system, `sshd` does not start. To stop it without rebooting, use the `service` command:

```
# service sshd stop
```

## ntsysv Utility



ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `ntsysv` utility is a command-line application with a simple text user interface to configure which services are to be started in selected run levels. Enter `ntsysv` from the command line to display the user interface. The utility displays a list of available services from the `/etc/rc.d/init.d/` directory along with their current status.

- [\*] means the service is enabled.
- [ ] means the service is disabled.

Use the up and down arrow keys to select a service. Use the Spacebar to enable or disable the selected service. Press F1 to display a description of the selected service.

To save any changes, use the Tab key to navigate to the OK button and press Enter. To discard changes, navigate to the Cancel button and press Enter.

By default, the `ntsysv` utility affects the current run level. To configure services for different run levels, use the `--level` option followed by the run level(s). The following example allows configuration of services for run levels 3 and 5:

```
# ntsysv --level 35
```

The `ntsysv` utility does not start or stop the service, it only configures it to start or stop (similar to the `chkconfig` utility). Use the `service` command to start or stop the service.

## Xinetd Service

- Xinetd is a “superserver” that starts programs that provide Internet services.
- Xinetd monitors all configured ports and starts services on demand.
  - Better security because daemons do not run all the time
  - Lower system resource (memory and CPU) usage
- Global configuration file is /etc/xinetd.conf.
- Configuration directory is /etc/xinetd.d, which contains configuration files for each service managed by xinetd.
- Use the chkconfig --list command to display status of services under xinetd control.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Xinetd stands for eXtended InterNet Daemon. It is a “superserver” that starts programs that provide Internet services. Instead of having a service start at system initialization time and always running, xinetd monitors all configured ports and starts services on demand. When a request for a service is made, xinetd starts the appropriate server. The benefits of using xinetd over running each service individually is better security and lower system resource (memory and CPU) usage because the services are not running all the time. Another benefit is that xinetd is not limited to services listed in the /etc/services file.

The main configuration file is /etc/xinetd.conf. This file contains general configuration settings that affect every service under xinetd control. See the manual page for xinetd.conf for a description of the configuration settings.

There is also the /etc/xinetd.d directory that contains configuration files for each service managed by xinetd. The names of the files in this directory correlate to the service. The format of files in the /etc/xinetd.d directory use the same conventions as /etc/xinetd.conf. As with xinetd.conf, this directory is read when the xinetd service is started. For any changes to take effect, you must restart the xinetd service. The primary reason the configuration for each service is stored in a separate file is to make customization easier and less likely to affect other services.

With xinetd installed, the chkconfig --list command displays the status of services under xinetd control.

# Upstart

- Oracle Linux 6 implements an event-based initialization system named Upstart.
- Services are started and stopped by events.
- At startup, `init` processing is controlled by the run level event emitted by a call to Upstart.
- Upstart is in a transition stage; therefore, Oracle Linux 6 maintains compatibility with SysV `init`.
- The `/etc/inittab` file is deprecated and is used for setting up only the default run level.
- Configuration is accomplished by the use of Upstart jobs that are defined in the `/etc/init` directory.
- Use the `initctl` command to control Upstart jobs.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle Linux 6 implements an event-based initialization system named Upstart. With Upstart, tasks and services are started and stopped by events. Events are used to trigger tasks or services, collectively known as jobs. Events may be triggered by hardware changes, by starting or stopping tasks, or by other processes on the system. At startup, `init` processing is controlled by the run level event emitted by a call to Upstart.

Upstart is in a transition stage in Oracle Linux 6. A few Upstart jobs are defined but Oracle Linux 6 maintains compatibility with the SysV `init` run level approach of managing services. With Upstart, the `/etc/inittab` file is deprecated and is used for setting up only the default run level. All lines in the `/etc/inittab` file are comments with the exception of the `initdefault` line. Other configuration is accomplished by the use of Upstart jobs that are defined in the `/etc/init` directory. Upstart reads its job configuration from this directory. Following is a partial list of the current Upstart jobs:

```
# ls /etc/init
control-alt-delete.conf      init-system-dbus.conf
kexec-disable.conf           plymouth-shutdown.conf
prefdm.conf                  quit-plymouth.conf
rc.conf                      rcS.conf
```

Files in the /etc/init directory must end in .conf to be processed by Upstart. Each file defines a single service or task. These files are plain text and not executable.

The main process in an Upstart .conf file is defined by using either the exec or script stanzas. These stanzas specify the executable or shell script that will be run when the job is considered to be running. For example, the /etc/init/rcS.conf job contains the following exec entry:

```
exec /etc/rc.d/rc.sysinit
```

The /etc/init/start-ttys.conf job contains the following script entry. Each script stanza is terminated by an end script stanza.

```
script
    . /etc/sysconfig/init
    for tty in $(echo $ACTIVE_CONSOLES) ; do
        [ "$RUNLEVEL" = "5" -a "$tty" = "$X_TTY" ] &&
    continue
        initctl start tty TTY=$tty
    done
end script
```

Use the start on <event> stanza to cause the job to be automatically started based on an <event>. Use the stop on <event> stanza to cause the job to be automatically stopped.

Upstart uses the initctl utility to control Upstart jobs. Several initctl commands are available. Run the following command to list all the Upstart jobs and their associated status. Non-root users can run initctl for read-only commands.

```
# initctl list
```

Enter either of the following commands to start a new instance of an Upstart job. The <job> argument is the name of a .conf file in the /etc/init directory.

```
# initctl start <job>
# start <job>
```

Enter either of the following commands to stop an Upstart job:

```
# initctl stop <job>
# stop <job>
```

Enter either of the following commands to view the status of an Upstart job:

```
# initctl status <job>
# status <job>
```

Enter the following command to view the Upstart logging priority:

```
# initctl log-priority
```

The default logging priority is “message”. By default, only the stopping of Upstart services is logged. To set the highest logging priority, “debug”, enter the following command:

```
# initctl log-priority debug
```

The following command requests that the <event> be emitted, causing other jobs to be started or stopped depending on the use of “start on” or “start off” in their configuration.

```
# initctl emit <event>
```

Event names can be user-defined. The most well known event used by the default Upstart configuration is the “runlevel” event.

## Summary

In this lesson, you should have learned:

- The purpose of the GRUB bootloader
- Configuration of the GRUB bootloader
- Configuration of kernel boot parameters
- The purpose of `/sbin/init` in the Linux boot process
- Differences in Linux init run levels
- The purpose of `/etc/inittab`
- The contents of the `/etc/rc.d` directories
- Configuration of services at different run levels
- The use of the `chkconfig` and `ntsysv` utilities
- The use of the `xinetd` service
- The Upstart event-based initialization system



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Quiz

Which of the following directives is needed for a bootable kernel entry in the GRUB configuration file to be valid?

- a. title
- b. root
- c. kernel
- d. initrd

# Quiz

Which of the following files is the default run level defined in?

- a. /boot/grub/grub.conf
- b. /etc/inittab
- c. /etc/rc.d/rc.sysinit
- d. /etc/rc.local

## Quiz

Using either the `chkconfig` or `ntsysv` utilities to configure a service to start in selected run levels also changes the current state of the service.

- a. True
- b. False



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 4: Overview

The practices for this lesson cover the following topics:

- Exploring the GRUB bootloader
- Booting different kernels
- Using the GRUB menu
- Changing the default run level
- Exploring and configuring init services
- Exploring and configuring Upstart jobs

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

## System Configuration

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to describe:

- System date and time configuration
- /etc/sysconfig directory
- proc file system
- sysfs file system
- sysctl utility

# Configuring System Time

- The `system-config-date` command provides a GUI from which you can:
  - Change the system date and time
  - Configure the time zone used by the system
  - Set up the NTP daemon to synchronize the system clock with a time server
- Run the `date` command to display or set the system date and time.
  - Can also be used to display dates in the past or future
- Run the `hwclock` command to display or set the hardware clock date and time.
  - Can sync hardware clock with system time
  - Can sync system time with hardware clock



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

You were presented with the Time Zone Selection window during installation, which allowed you to select the appropriate time zone for your location. From the Time Zone Selection window, you also had the option to specify (or not) that the system clock uses UTC.

You were presented with the Date and Time window during Firstboot. From this window, you could either “Synchronize date and time over the network” with a remote time server using NTP or “Manually set the date and time of your system”.

Use the `system-config-date` command to display the Date/Time Properties Tool. This tool has a GUI from which you can change the system date and time, configure the time zone used by the system, and set up the Network Time Protocol daemon to synchronize the system clock with a time server.

Command-line utilities also exist, allowing you to configure system time. The `date` command allows you to display or set the system date and time. Run the `date` command with no arguments to display the current date and time:

```
# date  
Wed Jul 10 13:10:04 MDT 2013
```

The `date` command provides a variety of output formatting options. For example, “`date +%A`” only displays the day of the week. Run the `date --help` command to view the usage.

You can display the time and date in the future, or in the past. For example, to display the date one year from now:

```
# date -d "1 year"  
Thu Jul 10 13:10:12 MDT 2014
```

To display the date one month in the past:

```
# date -d "1 month ago"  
Mon Jun 10 13:10:22 MDT 2014
```

Use the following syntax to change the current date. Replace *YYYY* with a four-digit year, *MM* with a two-digit month, and *DD* with a two-digit day of the month.

```
date +%D -s YYYY-MM-DD
```

To change the date to (for example) July 11, 2013:

```
# date +%D -s 2013-07-11
```

Use the following syntax to change the current time. Replace *HH* with a two-digit hour, *MM* with a two-digit minute, and *SS* with a two-digit second. Include the *-u* option if your system clock is set to use UTC.

```
date +%T -s HH-MM-SS -u
```

Use the `hwclock` command to query and set the hardware clock, also known as the RTC (real-time clock). This clock runs independently of any control program running in the CPU and even when the machine is powered off. The `hwclock` command allows you to:

- Display the current time
- Set the hardware clock to a specified time
- Set the hardware clock from the system time (`hwclock -w`)
- Set the system time from the hardware clock (`hwclock -s`)

The system time is the time kept by a clock inside the Linux kernel and driven by a timer interrupt. The system time is the time that matters. The hardware clock's basic purpose in a Linux system is to keep time when Linux is not running. The system time is initialized to the time from the hardware clock when Linux starts up, and then the hardware clock is not used again.

The following example displays the system time (using the `date` command), syncs the system time to the hardware clock (using the `hwclock -s` command), displays the hardware clock (using the `hwclock` command), and displays the new system time (using the `date` command):

```
# date  
Thu Jul 11 11:41:38 MDT 2013  
# hwclock -s  
# hwclock  
Thu Jul 11 2013 05:38:57 AM MDT -0.052243 seconds  
# date  
Thu Jul 11 05:38:59 MDT 2013
```

From this example, you can see that the `hwclock -s` command set the system time to be equal to the hardware clock.

## Using Network Time Protocol

- Network Time Protocol (NTP) provides a method of verifying and correcting your computer's time by synchronizing it with another system.
- Configure using the `system-config-date` command:
  - Select "Synchronize date and time over the network".
  - Select from a list of predefined NTP servers, or add server(s).
- Use the `ntpdate` command to perform a one-time synchronization of the system clock:
  - `# ntpdate 0.rhel.pool.ntp.org`
- Configure the `ntpd` daemon to synchronize the system time automatically at boot time:
  - Specify NTP servers in `/etc/ntp.conf`.
  - Start the `ntpd` service.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Network Time Protocol (NTP) provides a method of verifying and correcting your computer's time by synchronizing it with another system. Use the `system-config-date` command to display the Date/Time Properties Tool GUI, then select "Synchronize date and time over the network" to use NTP. You can then select from a list of predefined NTP servers:

- `0.rhel.pool.ntp.org`
- `1.rhel.pool.ntp.org`
- `2.rhel.pool.ntp.org`

Instead of using a predefined server, you can edit a predefined server, or add a new server. From the GUI, you also have the option to select "Synchronize the system clock before starting the service" and "Use a Local Time source." Click OK to save the configuration and automatically start the NTP daemon, `ntpd`.

You can use the `ntpdate` command to perform a one-time synchronization of the system clock with a remote NTP server. Run `ntpdate` followed by one or more server addresses. Example:

```
# ntpdate 0.rhel.pool.ntp.org 1.rhel.pool.ntp.org
```

If no errors are displayed, the system time is now synchronized with the remote NTP server. The `ntpd -q` command provides the same functionality as `ntpdate`. The `ntpdate` command will no longer be supported at some time in the future.

You can also configure the `ntpd` daemon to synchronize the system time automatically at boot time. Edit the `/etc/ntp.conf` file and add or edit the list of public NTP servers. The following list of servers is included by default but you can change or add as needed:

```
# cat /etc/ntp.conf
...
server 0.rhel.pool.ntp.org
server 1.rhel.pool.ntp.org
server 2.rhel.pool.ntp.org
...
```

After editing the `/etc/ntp.conf` file, use the `service` command to start the NTP daemon:

```
# service ntpd start
```

Use the `chkconfig` command to ensure the NTP daemon starts at boot time:

```
# chkconfig ntpd on
```

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

## /etc/sysconfig Directory

- The /etc/sysconfig directory contains a hierarchy of system configuration files.
- For complete information on these files, see /usr/share/doc/initscripts\*/sysconfig.txt.
- The contents of the directory vary depending on programs installed.
- Files in the directory describe:
  - Configuration parameters
  - Arguments for respective daemons
  - Custom options for commands
  - System defaults
  - Kernel information
  - Much more



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The /etc/sysconfig directory contains files that control the system configuration. See /usr/share/doc/initscripts\*/sysconfig.txt for complete information about these files. The actual content of your /etc/sysconfig directory depends on the programs that you have installed on your machine. Some of the files found in the /etc/sysconfig directory are described as follows:

- **atd**: This file is used to specify additional command line arguments for atd daemon.
- **authconfig**: This file sets the authorization to be used on the host. For example, USEMKHOMEDIR=no disables creating a home directory for a user on the first login.
- **autofs**: This file defines custom options for the automatic mounting of devices. It controls the operation of the automount daemons.
- **crond**: This file is used to pass arguments to the crond daemon at boot time.
- **i18n**: i18n stands for Internationalization (i plus 18 letters plus n). This configuration file defines the default language, any supported languages, and the default system font. These entries are kernel boot parameters in /boot/grub/grub.conf.

LANG=en\_US.UTF-8

SYSFONT=latarcyrheb-sun16

- **init**: This file controls how the system appears and functions during the boot process.

- **iptables-config**: This file stores information used by the kernel to set up packet filtering services at boot time or when the service is started.
- **iptables**: This file stores the actual firewall configuration rules.
- **keyboard**: This file controls the behavior of the keyboard. The KEYBOARDTYPE=pc and KEYTABLE=us entries are kernel boot parameters in /boot/grub/grub.conf.
- **modules**: This directory specifies kernel modules to be loaded at boot time.
- **named**: This file is used to pass arguments to the named daemon at boot time. The named daemon is a Domain Name System (DNS) server that implements the Berkeley Internet Name Domain (BIND) distribution. This server maintains a table that associates host names with IP addresses on the network.
- **network**: This file is used to specify information about the network configuration.
- **nfs**: NFS requires ports for RPC (remote procedure call) services. This causes problems for configuring firewall rules. To overcome this problem, use the /etc/sysconfig/nfs file to control which ports the required RPC services run on.
- **ntpd**: This file is used to pass arguments to the ntpd daemon at boot time. The ntpd daemon sets and maintains the system clock to synchronize with an Internet standard time server.
- **samba**: This file is used to pass arguments to the smbd and the nmbd daemons at boot time. The smbd daemon offers file sharing connectivity for Windows clients on the network. The nmbd daemon offers NetBIOS over IP naming services.

These files and other /etc/sysconfig files are described further in applicable lessons.

# proc File System

- The proc file system is a hierarchy of special files that represent:
  - The current state of the kernel
  - Details of system hardware
  - Running processes
  - System configuration information and interfaces
- It is a virtual file system containing virtual files.
- Use cat, more, less to view most files.
- Utilities exist to view other files, for example:
  - lspci, free, top
- Some files can be modified to adjust kernel settings:
  - echo value > /proc/file



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The proc file system contains a hierarchy of special files that represent the current state of the kernel. It is named after its original purpose, which is an interface to the structures within running processes to support debugging tools. Linux adopted this from Solaris but also added the interface to the kernel. The proc file system has become quite messy over the years and so Linux created the sysfs file system to clean it up.

Files in the /proc directory contain information about your hardware and current processes running on your system. Files that have write permission can be modified to change the configuration of the kernel.

## Virtual File System

Files in the /proc directory are virtual files; therefore, proc is referred to as a virtual file system. Most virtual files are listed as zero bytes in size but contain a large amount of information when viewed. Most of the time and date stamps on virtual files reflect the current time and date; however, these files are constantly updated.

Virtual files such as /proc/interrupts, /proc/meminfo, /proc/mounts, and /proc/partitions provide a view of the system's hardware. Others, like the /proc/filesystems file and the /proc/sys directory provide system configuration information and interfaces.

Files containing information about similar topics are grouped into virtual directories. For example, process directories contain information about each running process on the system.

## Viewing Virtual Files

Most virtual files within `/proc` can be viewed by using commands such as `cat`, `more`, and `less`. For example, to view information about the system's CPU, enter:

```
# cat /proc/cpuinfo
processor      : 0
vendor_id     : GenuineIntel
cpu family   : 6
model name   : Intel® Core™ i5-2520M CPU @ 2.5GHz
output omitted...
```

Certain files can only be accessed with `root` privileges. And some files in `/proc` contain information that is not human-readable. Use utilities such as `lspci`, `free`, and `top` to view these files. For example, use the `lspci` command to list all PCI devices:

```
# lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE
output omitted...
```

## Changing Virtual Files

Most virtual files within the `/proc` directory are read-only. However, some are writable and can be used to adjust settings in the kernel. This is especially true for files in the `/proc/sys` directory. To change the value of a virtual file, use the following syntax:

```
echo value > /proc/file
```

For example, to change the host name, enter:

```
# echo www.example.com > /proc/sys/kernel/hostname
```

Other files act as binary or Boolean switches. Viewing the file returns either a 0 (off or false) or a 1 (on or true). Example:

```
# cat /proc/sys/net/ipv4/ip_forward
0
```

The 0 indicates the kernel is not forwarding network packets. To turn packet forwarding on:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# cat /proc/sys/net/ipv4/ip_forward
1
```

## Top-Level Files Within /proc

```
[root@host03 ~]# ls /proc
1   15   1809  2103  2167  26   995      kallsyms    self
10  1500  1812  2109  2171  27   996      kcore        slabinfo
1044 1542  1824  2112  2176  28   997      keys         softirqs
11  1548  1826  2115  2178  29   acpi       key-users   stat
12  1568  1841  2117  2187  3    buddyinfo  kms         swaps
1275 1599  19    2119  22    30   bus        kpagemap   sys
13  16    1911  2128  2276  307  cgroups   kpagemflags sysrq-trigger
1300 1679  1917  2129  2277  308  cmdline   latency_stats sysvipc
1331 1686  1959  2131  2278  36   cpufreq   loadavg   timer_list
1346 1687  1968  2133  2279  37   crypto    locks     timer_stats
1357 17    1973  2135  23    38   devices   mdstat   tty
1361 1703  1983  2139  2348  39   diskstats meminfo   uptime
1372 1711  1992  2142  2368  394  dma       misc     version
1374 1739  2     2148  2394  4    driver    modules   vmallocinfo
1377 1771  20    2149  2395  5    execdomains mounts   vmstat
14  1782  2000  2152  2396  6    fb       mtrr     xen
1414 1795  2001  2155  24    7    filesystems net      zoneinfo
1457 18    2075  2156  2410  8    fs       pagetypeinfo
1461 1800  2080  2157  2418  9    interrupts partitions
1471 1802  2088  2160  242   945  iomem    sched debug
1490 1804  2090  2162  244   967  ioports  schedstat
1499 1805  21    2166  2488  994  irq     scsi
```

[root@host03 ~]#

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Some of the more useful virtual files in the top-level of the /proc directory are described here. This is not meant to be an all-inclusive list, but to give examples of some of the files and their purpose. Many of these files are described further in applicable lessons.

- **/proc/buddyinfo:** This file is used primarily for diagnosing memory fragmentation issues.
- **/proc/cmdline:** This file shows the parameters passed to the kernel at the time it is started.
- **/proc/cpuinfo:** This virtual file identifies the type of processor used by your system.
- **/proc/crypto:** This file lists all installed cryptographic ciphers used by the Linux kernel, including additional details for each.
- **/proc/devices:** This file displays the various character and block devices currently configured (not including devices whose modules are not loaded).

- **/proc/dma:** This file contains a list of the registered ISA DMA channels in use.
- **/proc/execdomains:** This file lists the execution domains currently supported by the Linux kernel, along with the range of personalities they support.
- **/proc/filesystems:** This file displays a list of the file system types currently supported by the kernel. The first column signifies whether the file system is mounted on a block device. Those beginning with nodev are not mounted on a device. The second column lists the names of the file systems supported. The `mount` command cycles through the file systems listed here when one is not specified as an argument.
- **/proc/interrupts:** This file records the number of interrupts per IRQ on the x86 architecture.
- **/proc/iomem:** This file shows you the current map of the system's memory for each physical device.
- **/proc/ioports:** This file provides a list of currently registered port regions used for input or output communication with a device.
- **/proc/kcore:** This file represents the physical memory of the system and is stored in the core file format. The contents of this file are designed to be examined by a debugger, such as `gdb`, and is not human readable.
- **/proc/kmsg:** This file is used to hold messages generated by the kernel. These messages are then picked up by other programs, such as `/bin/dmesg`.
- **/proc/loadavg:** This file provides a look at the load average in regard to both the CPU and I/O over time, as well as additional data used by `uptime` and other commands.
- **/proc/locks:** This file displays the files currently locked by the kernel. The contents of this file contain internal kernel debugging data and can vary tremendously, depending on the use of the system.
- **/proc/mdstat:** This file contains the current information for multiple-disk, RAID configurations.
- **/proc/meminfo:** This file reports a large amount of valuable information about the system's RAM usage.
- **/proc/modules:** This file displays a list of all modules loaded into the kernel. Most of this information can also be viewed by using the `/sbin/lsmod` command.

## Process Directories in /proc

- Process directories are named after a program's process ID (PID).
- They contain information specific to process:
  - **cmdline**: The command issued
  - **cwd**: The current working directory
  - **environ**: Environment variables
  - **fd**: File descriptors
  - **maps**: Memory maps to executables and library files
  - **root**: A link to the root directory of the process
  - **status**: The status of the process



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The /proc directory contains directories with numerical names. These directories are named after a program's process ID and contain information about that process. The owner and group of each process directory are set to the user running the process. Each process directory contains several files including:

- **cmdline**: The command issued when starting the process
- **cwd**: A symbolic link to the current working directory for the process
- **environ**: A list of the environment variables for the process
- **exe**: A symbolic link to the executable of this process
- **fd**: The directory containing all of the file descriptors for a particular process
- **maps**: A list of memory maps to executables and library files associated with process
- **mem**: The memory held by the process (the file cannot be read by the user)
- **root**: A link to the root directory of the process
- **stat**: The status of the process including run state and memory usage
- **statm**: The status of the memory in use by the process
- **status**: The status of the process in a more readable form than stat or statm

## Other Directories in /proc

- Directories group information concerning the kernel.
- Examples include:
  - /proc/bus: Available buses
  - /proc/driver: Drivers in use by the kernel
  - /proc/fs: Exported file systems
  - /proc/net: Network parameters and statistics
  - /proc/scsi: SCSI devices
- /proc/sys contains files used to enable or disable kernel features.
  - Files with write permission may be used to configure the kernel.
  - Use `echo value > filename` to make changes.
  - Changes are not persistent across reboot.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Other directories within the /proc directory group similar information by topic. The following is a partial list of these directories:

- **/proc/bus:** This directory contains information about the various buses available on the system. The subdirectories and files available within /proc/bus vary depending on the devices connected to the system.
- **/proc/bus/pci, /proc/bus/usb:** You can get a list of all PCI and USB devices present on the system by using the `cat` command on the `devices` file within these directories, but the output is difficult to read and interpret. For a human-readable list of devices, run the `lspci` and `lsusb` commands.
- **/proc/driver:** This directory contains information for specific drivers in use by the kernel.
- **/proc/fs:** This directory shows which file systems are exported. If running an NFS server, typing `cat /proc/fs/nfsd/exports` displays the file systems being shared and the permissions.
- **/proc/irq:** This directory is used to set IRQ to CPU affinity, which allows the system to connect a particular IRQ to only one CPU. Alternatively, it can exclude a CPU from handling any IRQs.

- **/proc/self/net:** This directory provides a comprehensive look at various networking parameters and statistics. Each directory and virtual file within this directory describes aspects of the system's network configuration. The /proc/net file is a symbolic link to this directory.
- **/proc/scsi:** The primary file in this directory is /proc/scsi/scsi, which contains a list of every recognized SCSI device. From this listing, the type of device, as well as the model name, vendor, SCSI channel, and ID data is available.
- **/proc/sys:** This directory is different from others in /proc, because it not only provides information about the system but also allows you to immediately enable and disable kernel features. If a file has write permissions, it may be used to configure the kernel. Changing a value within a /proc/sys/ file is done by echoing the new value into the file. For example, to change the host name to www.example.com:

```
# echo www.example.com > /proc/sys/kernel/hostname
```

Other files act as binary or Boolean switches. A value of 0 represents off or false. A value of 1 represents on or true. For example, to turn packet forwarding on:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Changes made by using the echo command are not persistent and disappear when the system is restarted. To make configuration changes take effect after the system is rebooted, add them to the /etc/sysctl.conf file.

- **/proc/sys/dev:** This directory provides parameters for particular devices on the system.
- **/proc/sys/fs:** This directory contains options and information concerning various aspects of the file system, including quota, file handle, and inode information.
- **/proc/sys/kernel:** This directory contains a variety of different configuration files that directly affect the operation of the kernel.
- **/proc/sys/net:** This directory contains subdirectories concerning various networking topics. You can alter the files within these directories to adjust the network configuration on a running system.
- **/proc/sysvipc:** This directory contains information about System V Interprocess Communication (IPC) resources. The files in this directory relate to System V IPC calls for messages (msg), semaphores (sem), and shared memory (shm).
- **/proc/tty:** This directory contains information about the available and currently used tty devices on the system. The drivers file is a list of the current tty devices in use.

## sysfs File System

- Beginning with version 2.6, the kernel also exports information to another virtual file system called sysfs.
- sysfs is mounted on /sys.

```
[root@host03 ~]# ls -l /sys
total 0
drwxr-xr-x. 2 root root 0 Jan 11 06:46 block
drwxr-xr-x. 15 root root 0 Jan 11 06:46 bus
drwxr-xr-x. 37 root root 0 Jan 11 06:46 class
drwxr-xr-x. 4 root root 0 Jan 11 06:46 dev
drwxr-xr-x. 16 root root 0 Jan 11 06:46 devices
drwxr-xr-x. 4 root root 0 Jan 12 08:44 firmware
drwxr-xr-x. 4 root root 0 Jan 12 08:44 fs
drwxr-xr-x. 5 root root 0 Jan 12 08:44 hypervisor
drwxr-xr-x. 5 root root 0 Jan 11 06:46 kernel
drwxr-xr-x. 71 root root 0 Jan 11 06:47 module
drwxr-xr-x. 2 root root 0 Jan 11 06:47 power
[root@host03 ~]#
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

In addition to /proc, the kernel also exports information to another virtual file system called sysfs. sysfs is used by programs such as udev to access device and device driver information. The creation of sysfs helped clean up the proc file system because much of the hardware information has been moved from proc to sysfs.

The sysfs file system is mounted on /sys. The top-level directories are shown. Following is a brief description of some of these directories:

- /sys/block: This directory contains entries for each block device in the system. Symbolic links point to the physical device that the device maps to in the physical device tree. For example, attributes for the xvda virtual disks reside in the following directory:

```
# ls /sys/block/xvda
/sys/block/xvda:
alignment_offset  device    power      ro       subsystem xvda2
bdi              ext_range queue     size      trace     xvda3
capability      holders   range      slaves   uevent    xvda4
dev              inflight removable stat     xvda1    xvda5
```

- **/sys/bus**: This directory contains subdirectories for each physical bus type supported in the kernel. Each bus type has two subdirectories: devices and drivers. The devices directory lists devices discovered on that type of bus. The drivers directory contains directories for each device driver registered with the bus type. Driver parameters can be viewed and manipulated. For example, to list the drivers for the virtual devices, enter:

```
# ls -lR /sys/bus/xen/drivers
...
/sys/bus/xen/drivers/vbd
lrwxrwxrwx. module -> ../../../../../../module/xen_blkfront
...
/sys/bus/xen/drivers/vif
lrwxrwxrwx. module -> ../../../../../../module/xen_netfront
...
```

- **/sys/class**: This directory contains every device class registered with the kernel. Device classes describe a functional type of device. Examples include input devices, network devices and block devices.
- **/sys/devices**: This directory contains the global device hierarchy of all devices on the system. This directory also contains a platform directory and a system directory. The platform directory contains peripheral devices specific to a particular platform such as device controllers. The system directory contains non-peripheral devices such as CPUs and APICs.
- **/sys/firmware**: This directory contains subdirectories with firmware objects and attributes.
- **/sys/module**: This directory contains subdirectories for each module that is loaded into the kernel, for example:

```
# ls /sys/module/xen*
/sys/module/xen_blkfront
  drivers  holders  initstate  notes  refcnt  sections  srcversion
/sys/module/xen_netfront
  drivers  holders  initstate  notes  refcnt  sections  srcversion
```

- **/sys/power**: The system power state can be controlled from this directory. The disk attribute controls the method by which the system will suspend to disk. The state attribute allows a process to enter a low power state.

## sysctl Utility

- The sysctl utility is used to assign values to writable files in /proc/sys.
- To view kernel settings, enter:
  - sysctl -a
- To assign values, the syntax is:
  - sysctl -w *kernel parameter=value*
- Example:
  - sysctl -w net.ipv4.ip\_forward=1
- Changes are lost when the system is rebooted.
- To preserve settings, add them to the config file:  
/etc/sysctl.conf



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The /sbin/sysctl utility can also be used to view or modify values to writable files in the /proc/sys directory. To view the current kernel settings, enter:

```
# sysctl -a
kernel.sched_child_runs_first = 0
kernel.sched_min_granularity_ns = 1000000
output omitted...
```

This is the same information seen if each of the files were viewed individually.

The echo command can be used to assign values to writable files in /proc/sys:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

The equivalent sysctl command follows, displaying the result of the change immediately:

```
# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

Changes made using both echo and sysctl are lost when the system is rebooted. To preserve custom settings, add them to the /etc/sysctl.conf file. Values added to this file take effect each time the system boots.

Changes made in the /etc/sysctl.conf file take effect immediately when issuing the following command:

```
# sysctl -p
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
output omitted...
kernel.msgmax = 65536
kernel.shmmmax = 68719476736
kernel.shmall = 4294967296
```

Notice the entries in /etc/sysctl.conf correspond to the directory hierarchy in /proc/sys. For example, to view the value of the net.ipv4.ip\_forward kernel parameter:

```
# cat /proc/sys/net/ipv4/ip_forward
0
# grep net.ipv4.ip_forward /etc/sysctl.conf
net.ipv4.ip_forward = 0
```

To view the value of the kernel.msgmax kernel parameter:

```
# cat /proc/sys/kernel/msgmax
65536
# grep kernel.msgmax /etc/sysctl.conf
kernel.msgmax = 65536
```

# Quiz

Which of the following directories contains files that you can edit to pass arguments to daemons?

- a. /etc/sysconfig
- b. /proc/sys
- c. /var/spool
- d. /var/log



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Quiz

Which of the following directories contains files that allow you to immediately enable and disable kernel features?

- a. /etc/sysconfig
- b. /proc/sys
- c. /var/spool
- d. /var/log

## Quiz

Which of the following contains process directories named after a program's process ID, each containing information specific to that process?

- a. /etc
- b. /proc/sys
- c. /proc/pid
- d. /proc

## Summary

In this lesson, you should have learned about:

- Configuring system time
- System configuration files in `/etc/sysconfig` directory
- Kernel state in the `/proc` directory
- Hardware devices and drivers in `sysfs`
- Usage of the `sysctl` utility to change kernel state



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 5: Overview

The practices for this lesson cover the following tasks:

- Exploring the `/etc/sysconfig` directory
- Exploring the `proc` file system
- Exploring the `sysfs` file system
- Using the `sysctl` utility



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Package Management

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

- Describe Oracle Linux package management
- Use the `rpm` utility
- Describe the Oracle public `yum` server
- Describe and configure `yum` repositories
- Use the `yum` utility
- Describe the Unbreakable Linux Network (ULN)
- Describe the steps to switch from RHN to ULN

# Introduction to Package Management

- Oracle Linux uses Red Hat Package Manager (RPM).
- Oracle Linux also provides the `yum` utility, which:
  - Resolves RPM dependencies
  - Connects to repositories to download software
- Oracle public `yum` server:
  - Offers a free way to install packages
  - Does have free errata
- Unbreakable Linux Network (ULN):
  - Is a comprehensive resource for support subscribers
  - Offers access to software patches, updates, and fixes



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

All software on a Linux system is divided into packages that can be installed, uninstalled, upgraded, queried, and verified. Oracle Linux uses the Red Hat Package Manager (RPM) to facilitate the installation, upgrade and removal of software packages.

Oracle Linux also provides the `yum` utility, which works with RPM packages. When `yum` installs or upgrades a software package, it also installs or upgrades any package dependencies. The `yum` utility downloads package headers and packages from repositories. Repositories are storage locations from which software packages can be retrieved and installed.

## Oracle Public `yum` Server

The Oracle public `yum` server offers a free and convenient way to install packages from the Oracle Linux installation media via a `yum` client. Errata (bug fixes, security fixes, and enhancement) are also available from this public `yum` server for free.

## Unbreakable Linux Network (ULN)

The Unbreakable Linux Network (ULN) is a comprehensive resource for Oracle Unbreakable Linux support subscribers. ULN offers access to Linux software patches, updates, and fixes. Extra packages not included in the original distribution can also be downloaded from ULN.

# **rpm Utility**

- Query options:

```
# rpm -qa  
# rpm -qi package_name  
# rpm -ql package_name  
# rpm -qf filename  
# rpm -qc package_name
```

- Installing and updating packages:

```
# rpm -Uvh package_name
```

- Installing a new kernel:

```
# rpm -ivh kernel_package_name
```

- Removing packages:

```
# rpm -e package_name
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `rpm` utility provides many useful options for querying and verifying packages, as well as installing, upgrading, and removing packages. The following provides examples of these options.

## **Query Packages**

To list all installed packages, use the following command:

```
# rpm -qa  
orca-2.28.2-1.el6.x86_64  
avahi-0.6.25-11.el6.x86_64  
iso-codes-3.16-2.el6.noarch  
GConf2-gtk-2.28.0-6.el6.x86_64  
...
```

The format of `rpm` package names is `name-version-release.architecture`. The example shows packages for version 6 of Oracle Linux (el6) with architectures of either:

- `x86_64`: Any AMD64 or Intel 64 CPUs
- `noarch`: Any CPU architecture

To display detailed package information (of the `filesystem` package, for example), enter:

```
# rpm -qi filesystem
```

To list the files in a package (the `bash` package, for example), enter:

```
# rpm -ql bash
```

To perform a reverse search, that is to determine what package a specific file (`/etc/hosts`, for example) belongs to, enter:

```
# rpm -qf /etc/hosts
```

To list configuration files associated with a package (the `bash` package, for example), enter:

```
# rpm -qc bash
/etc/skel/.bash_logout
/etc/skel/.bash_profile
/etc/skel/.bashrc
```

## Installing and Updating Packages

Using the `rpm -U package_name` command upgrades installed packages, as well as installs new packages. For example, to install or upgrade the `zsh` package:

```
# rpm -Uvh zsh-4.3.10-4.1.el6.x86_64.rpm
```

The `-v` (verbose) option displays more information and the `-h` (hash) option displays progress.

## Installing a New Kernel

When installing a new kernel, use the `-i` option so as not to upgrade the current kernel, for example:

```
# rpm -ivh kernel-uek-2.6.39-100.5.1.el6uek.x86_64.rpm
```

## Removing Packages

To remove a package (the `zsh` package, for example), enter:

```
# rpm -e zsh-4.3.10-4.1.el6.x86_64
```

## Oracle Public yum Server

- Oracle offers free packages from Oracle Linux installation media.
- Errata are also available from the Oracle public yum server.
  - Subscribe to Oracle errata mailing list from this site.
- Packages can be accessed at:
  - <http://public-yum.oracle.com/>
- Use the wget utility to download the repo file.
- The wget utility updates the /etc/yum.repos.d directory.
- After the download, enable the appropriate yum repository.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Oracle public yum server offers a free and convenient way to install packages from the Oracle Linux installation media via a yum client. Errata (bug fixes, security fixes, and enhancement) are also available from this public yum server. You can also subscribe to the Oracle Linux errata mailing list from this site. The public yum server is offered without support of any kind and can be accessed at <http://public-yum.oracle.com/>.

Use the wget utility to download the repo file into the /etc/yum.repos.d directory. The wget utility is a noninteractive command-line utility that can retrieve files by using HTTP, HTTPS, or FTP.

For Oracle Linux 6, download the yum repo configuration file by running the following commands as root:

```
# cd /etc/yum.repos.d  
# wget http://public-yum.oracle.com/public-yum-ol6.repo
```

After the file has downloaded, the /etc/yum.repos.d directory is updated as follows:

```
# ls /etc/yum.repos.d  
public-yum-ol6.repo
```

## Contents of public-yum-ol6.repo:

```
# cat public-yum-ol6.repo
[public_ol6_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=http://public-
yum.oracle.com/repo/OracleLinux/OL6/latest/$basearch
gpgkey=http://public-yum.oracle.com/RPM-GPG-KEY-oracle-ol6
gpgcheck=1
enabled=1

[public_ol6_ga_base]
name=Oracle Linux $releasever GA installation media copy
($basearch)
baseurl=http://public-
yum.oracle.com/repo/OracleLinux/OL6/0/base/$basearch
gpgkey=http://public-yum.oracle.com/RPM-GPG-KEY-oracle-ol6
gpgcheck=1
enabled=0

[public_ol6_u1_base]
...
[public_ol6_u2_base]
...
[public_ol6_u3_base]
...
[public_ol6_UEK_latest]
...
[public_ol6_UEK_base]
...
```

Enable the appropriate repository by editing the yum repo configuration file. Locate the section in the file for the repository that you plan to update from—for example, [public\_ol6\_latest]. Set enabled to 1 (enable) or 0 (disable).

After the repository is enabled, you can begin using yum, for example:

```
# yum list
Installed Packages
389-ds-base-libs.i686      1.2.9.14-1.el6      @ol6_u2_base
ConsoleKit.i686            0.4.1-3.el6       @anaconda-
OracleLinux...
...
```

Oracle Public Yum no longer supports HTTPS. It is now using Akamai, so when accessing Oracle Public Yum on an Oracle Network (in an office or while on VPN) you will need to do the following (please modify these settings for proper proxy URLs in geographies other than the US):

For wget, set the "http\_proxy" environment variable:

```
# export http_proxy=http://www-proxy.us.oracle.com:80/
```

When using yum internally, add the following to the /etc/yum.conf file:

```
proxy=http://www-proxy.us.oracle.com:80/
```

You could use ftp://gdsuln/YUM\_local/OracleLinux or  
ftp://obiftp/YUM\_local/OracleLinux. Oracle hosts a copy of ol6\_latest and  
ol5\_latest on the utility servers worldwide. Sample configuration:

```
[ol6_latest_local]
name=OL6 updates from local utility server
baseurl=ftp://obiftp/YUM_local/OracleLinux/OL6/latest/$basearch
gpgcheck=1
enabled=1
proxy=_none_

[ol6_UEK_latest_local]
name=Latest Unbreakable Enterprise Kernel for Oracle Linux
$releasever ($basearch)
baseurl=ftp://obiftp/YUM_local/OracleLinux/OL6/UEK/latest/$basearch
gpgcheck=1
enabled=1
proxy=_none_
```

## yum Configuration

- /etc/yum.conf:
  - Is the primary configuration file
  - Holds global settings
- /etc/yum.repos.d:
  - Is the directory that defines repositories
  - Contains repo files
- repo files define which repositories to use.
- Each repo file includes specifications for related repositories.
- The baseurl directive indicates the location of the main repository.
- The enabled directive (set to 1) designates the repository to use.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The main configuration file for yum is /etc/yum.conf. Configuration files that define repositories are in the /etc/yum.repos.d directory. An example of /etc/yum.conf follows here:

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit=3
```

Global configurations are defined in the [main] section:

- **cachedir**: The directory to store downloaded packages
- **keepcache**: Set to 0 to indicate to remove packages after installing them.
- **debuglevel**: The amount of information logged, from 0 to 10
- **logfile**: The yum log file
- **exactarch**: When set to 1, yum updates packages only with packages of the same architecture.
- **obsoletes**: When set to 1, yum replaces obsolete packages during an update.
- **gpgcheck**: When set to 1, yum checks the GPG signatures to verify authenticity of the packages. The gpgkey directive specifies the location of the GPG key.
- **plugins**: When set to 1, enables yum plugins that extend functionality.
- **installonly\_limit**: The maximum number of versions that can be installed simultaneously for any single package

## yum Repositories

Oracle Linux stores information about each repository in a separate file in the /etc/yum.repos.d directory. The following is an example:

```
# ls /etc/yum.repos.d  
public-yum-ol6.repo  ULN-Base.repo  ULN-Base.repo.uln-int
```

The repo files define which repositories to use. Each repo file includes specifications for several related repositories. For example, the ULN-Base.repo file holds [ol6\_latest], [ol6\_ga\_base], [ol6\_ga\_patch], and [ol6\_ul\_base]:

```
# cat ULN-Base.repo  
[ol6_latest]  
name=Oracle Linux 6Server i386  
baseurl=http://uln-  
internal.oracle.com/yum/OracleLinux/OL6/latest/$basearch  
gpgcheck=1  
enabled=1  
...  
...
```

The directives in the repo files include:

- **name**: Describes the repository
- **baseurl**: Is the location of the main repository (`http://`, `ftp://`, or `file://`)
- **enabled**: When set to 1, yum uses the repository. The repository is disabled if set to 0.

yum repositories can also be locally accessible, not just over the Internet. Local yum repositories are created by using the `createrepo` command and then setting `baseurl` to the local directory.

# yum Utility

- List all packages:

```
# yum list
```

- List all installed packages:

```
# yum list installed
```

- List packages available to be installed:

```
# yum list available
```

- Check for updates to installed packages:

```
# yum check-update
```

- Update, install, remove packages:

```
# yum update package_name
```

```
# yum install package_name
```

```
# yum remove package_name
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `yum` utility is often the fastest way to perform package management tasks. It provides capabilities beyond those provided by `rpm` and by graphical package management tools. There are many `yum` commands, but the following provides examples of common tasks.

## Listing Packages

There are several `yum` commands to list packages in any repository enabled on your system or installed. You can list specific types of packages as well as refine your list with a package specification of any package's name, architecture, version, or release.

To list all packages in all the repositories and all the packages installed on your system, use the following command, `yum list all` gives the same output:

```
# yum list
```

To list all the packages installed on the system, use the following command:

```
# yum list installed
```

To list all the packages available to be installed in any enabled repository on your system, use the following command:

```
# yum list available
```

The following example finds the name of the package that a file (for example, /etc/sysconfig/atd) belongs to:

```
# yum provides /etc/sysconfig/atd  
at-3.1.10-43.el6.x86_64 : Job spooling tools
```

## Checking for Updates

To see which installed packages on your system have updates available, use the following command:

```
# yum check-update
```

The package name plus architecture, the version of the updated package, and the repository (or ULN channel) are displayed. Entering `yum list update` returns the same output.

## Updating Packages

You can choose to update a single package, multiple packages, or all packages at once. If any dependencies of the package (or packages) have updates available, they are updated also.

### Updating a Single Package

To update a single package, use the following command syntax:

```
yum update package_name
```

For example, to update the bind-libs package, enter:

```
# yum update bind-libs
```

`yum` checks dependencies, displays dependencies resolved and a transaction summary, prompts “Is this ok [y/N]”, waits for your response, and then downloads and installs the package and any dependent packages needed. Use `yum -y` to bypass the prompt.

### Updating All Packages

To update all packages and their dependencies, enter `yum update` (without any arguments):

```
# yum update
```

## Installing Packages

To install a new package together with any package dependencies, use the following syntax:

```
yum install package_name
```

For example, to install the `zsh` package, enter:

```
# yum install zsh
```

## Updating and Installing Kernels

You do not need to worry about the distinction between installing and upgrading a kernel package when you use `yum`. `yum` always installs a new kernel regardless of whether you are using `yum update` or `yum install`.

## Removing Packages

To remove a package, use the following syntax:

```
yum remove package_name
```

For example, to remove the `zsh` package, enter:

```
# yum remove zsh
```

## yum Groups

- yum groups are collections of software packages referred to by a single “group” name.
- yum supports the following group commands:

```
# yum grouplist  
# yum groupinfo groupname  
# yum groupinstall groupname  
# yum groupupdate groupname  
# yum groupremove groupname
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Software packages that interoperate with others can be packaged together as a group. The yum command includes several subcommands for managing these groups.

To list installed and available groups in two separate lists, use the following command:

```
# yum grouplist
```

To get detailed information about a specific group, use the following command:

```
# yum groupinfo groupname
```

To install/update packages in a group, use the following command:

```
# yum groupinstall groupname  
# yum groupupdate groupname
```

To remove all packages in a group, use the following command:

```
# yum groupremove groupname
```

# Unbreakable Linux Network (ULN)

ULN is accessed at <https://linux.oracle.com/>

The screenshot shows the Oracle ULN homepage. At the top, there's a navigation bar with links for "Getting Started With Unbreakable Linux Network", "About Oracle Linux Support", "About Oracle's Unbreakable Enterprise Kernel", "NEW: UNL Documentation", "NEW: Errata and CVE Information", "About Oracle VM", and "Reporting An Issue with Oracle Linux". The main content area contains several sections: "Getting Access" (with links to register, buy support subscriptions, and switch from RHN), "About ULN" (with links to data sheet, white paper, FAQ, and local YUM setup), and "Create New ULN User". There are also sections for "Oracle Linux Support" (including links to more information, features, and documentation), "Oracle's Unbreakable Enterprise Kernel" (with links to getting started and release notes), and "NEW: UNL Documentation" (with a link to the CSI Administration feature). The footer of the page includes a "Sign On" and "Register - FAQ" button, and a copyright notice: "Copyright © 2014, Oracle and/or its affiliates. All rights reserved."

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Unbreakable Linux Network (ULN) is a comprehensive resource for Oracle Unbreakable Linux support subscribers. ULN offers Linux software patches, updates, and fixes, along with information on the `yum` program and support policies. Support subscribers can also download useful extra packages not included in the original distribution. ULN can be accessed at <https://linux.oracle.com/>. Also included on the ULN site are instructions to register with ULN, to create local `yum` repositories, and to switch from Red Hat Network (RHN) to ULN.

## The Update Agent (`yum`)

ULN subscribers have the option of using `yum` to manage their systems. To use ULN and `yum`, users must register their systems with ULN and subscribe to a ULN channel. It is also possible to subscribe to multiple channels at once. There are several ULN channels available, and one containing the latest version is automatically chosen after registration, depending on the architecture and OS revision of the system to be managed.

After it is started, `yum` connects to the central ULN server repository and downloads the latest software packaged in RPM format. It then installs RPMs on the registered machine, maintaining a log. `yum` lets you choose which packages to update, because it is not necessary to install all the newly available packages. You can get a list of all the available packages and then choose which ones to download.

# ULN Channels

Register your systems and subscribe to a ULN channel.

Name	Label	Description	Packages
Oracle Linux 6 Latest (x86_64)	ol6_x86_64_latest	All packages released for Oracle Linux 6 (x86_64) including the latest errata packages. (x86_64)	6768
Latest Unbreakable Enterprise Kernel for Oracle Linux 6 (x86_64)	ol6_x86_64_UEK_latest	Latest Unbreakable Enterprise Kernel packages for Oracle Linux 6 (x86_64)	15
Dtrace for Oracle Linux 6 (x86_64) - Latest	ol6_x86_64_Dtrace_latest	Latest packages required for Dtrace on Oracle Linux 6 (x86_64)	11
Unbreakable Enterprise Kernel Release 3 for Oracle Linux 6 (x86_64) - Latest	ol6_x86_64_UERK3_latest	Latest packages for Unbreakable Enterprise Kernel Release 3 for Oracle Linux 6 (x86_64)	22
Oracle Linux 6 GA (x86_64)	ol6_ga_x86_64_base	All packages released for Oracle Linux 6 GA (x86_64). No errata included.	6010
Oracle Linux 6 GA Patches (x86_64)	ol6_ga_x86_64_patch	Updated packages published after release of Oracle Linux 6 (x86_64)	724
Oracle Linux 6 Dtrace Userspace Tools (x86_64) - Latest	ol6_x86_64_Dtrace_userspace_latest	The latest Dtrace userspace tools for Oracle Linux 6 (x86_64).	2
Oracle Linux 6 Update 1 Installation media copy (x86_64)	ol6_u1_x86_64_base	All packages released on the Oracle Linux 6 Update 1 (x86_64) installation media. This channel does not contain updates.	6148
Oracle Linux 6 Update 1 Patch (x86_64)	ol6_u1_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 1 (x86_64).	631
Oracle Linux 6 Update 2 Installation media copy (x86_64)	ol6_u2_x86_64_base	All packages released on the Oracle Linux 6 Update 2 (x86_64) installation media. This channel does not contain updates.	6279
Oracle Linux 6 Update 2 Patch (x86_64)	ol6_u2_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 2 (x86_64).	287
Oracle Linux 6 Update 3 Installation media copy (x86_64)	ol6_u3_x86_64_base	All packages released on the Oracle Linux 6 Update 3 (x86_64) installation media. This channel does not contain updates.	6324
Oracle Linux 6 Update 3 Patch (x86_64)	ol6_u3_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 3 (x86_64)	743
Oracle Linux 6 Update 4 Installation media copy (x86_64)	ol6_u4_x86_64_base	All packages released on the Oracle Linux 6 Update 4 (x86_64) installation media. This channel does not contain updates.	6245
Oracle Linux 6 Update 4 Patch (x86_64)	ol6_u4_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 4 (x86_64)	916
Oracle Linux 6 Update 5 Installation media copy (x86_64)	ol6_u5_x86_64_base	All packages released on the Oracle Linux 6 Update 5 (x86_64) installation media. This channel does not contain updates.	6421
Oracle Linux 6 Update 5 Patch (x86_64)	ol6_u5_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 5 (x86_64)	115
HA Utilities for MySQL and Oracle Linux 6 (x86_64)	ol6_x86_64_mysql-ha-util	Management Utilities for MySQL HA with Oracle Linux 6	2

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

There are more than 200 unique channels supported by ULN. The architectures currently supported are i386, x86\_64, and ia64 (starting with Oracle Linux 4 update 6 and Oracle Linux 5 update 4). You are automatically subscribed to the channel containing the latest software for the architecture and OS revision of your system.

You may also choose a specific OS revision that you would like your system to remain at. You should subscribe to the appropriate channel corresponding to the architecture of your system and the update level desired. Specific revisions of Oracle Linux have patches and errata issued, but you are not forced to upgrade from a given revision level to the next to get these fixes. Channels also exist for Oracle VM, OCFS2, RDS, and productivity applications. The installer determines which architecture to run.

## \_base Channels

The \_base channels (also known as installation media channels) provide RPMs for each major version and subsequent minor updates of Oracle Linux as released on their respective installation media (DVD or ISO). For example, there is a \_base channel for Oracle Linux 6 Update 3 as well as Oracle Linux 6. Security errata and bug fixes are not published to these channels.

### **\_patch Channels**

The \_patch channels provide only the packages that have changed since the initial release of a particular version (whether a minor update or a major version). If multiple releases are created for the same package, due to multiple vulnerabilities found at different times, these channels always provide the most recent version of such a package.

### **\_latest Channels**

The \_latest channels provide RPMs for all the packages in the distribution, including those errata also provided in the \_patch channels (that is, the version of any RPM downloadable on the \_latest channels is always the most recent available). For some RPMs, this corresponds to the same version distributed initially with the original distribution (if no vulnerabilities have been found to date). For others, the version is the same as was provided in the \_patch channel for the highest update level. For example, the ol6\_<arch>\_latest channel for Oracle Linux 6 contains the combination of the ol6\_u3\_<arch>\_base and ol6\_u3\_<arch>\_patch channels.

### **\_addons Channels**

The \_addons channels provide RPMs not included in the base distribution, such as RPMs to be used in creating a yum repository for Oracle Linux 6.

### **\_oracle Channels**

The \_oracle channels provide distribution for Oracle freely downloadable software (in RPM format) that runs on Linux (for instance, Oracle Instant Client and asmlib).

## **New Channels**

As new major releases and new minor updates of Oracle Linux become available, new channels are created by Oracle, to distribute the new RPMs. That is, the current ol6\_u1\_<arch>\_base and ol6\_u1\_<arch>\_patch channels remain available and do not include the new updates, making it therefore possible for ULN subscribers to remain on a specific release level of Oracle Linux and selectively apply errata on top of that. Every time a new minor update is released, two new channels (\_base and \_patch) are created for each architecture. The ol6\_<arch>\_latest channels continue to distribute the highest possible version of any package, and therefore follow the “head” of the development tree, independent of the update level. A similar philosophy is followed with the channels for major versions of Oracle Linux.

If you prefer to remain at a certain update level, but are currently subscribed, for example, to the ol6\_<arch>\_latest channel, you must subscribe to the ol6\_u<number>\_<arch>\_patch and ol6\_u<number>\_<arch>\_base channels for the desired update level and architecture and then unsubscribe from the \_latest channel. This can be done through the web interface.

## UEK R3 Kernel Image and User Space Packages on ULN

- The UEK R3 kernel image and user space packages are available on the following ULN channels:
  - ol6\_latest
  - ol6\_UEK\_latest
  - ol6\_x86\_64\_UEK\_R3\_latest
  - ol6\_x86\_64\_Dtrace\_userspace\_latest
  - ol6\_x86\_64\_ofed\_UEK
  - ol6\_x86\_64\_mysql-ha-utils
- You need to subscribe to ol6\_x86\_64\_UEK\_R3\_latest as well as ol6\_UEK\_latest and ol6\_latest.
- The other channels are needed for DTrace, OFED, and DRBD.

**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The UEK R3 kernel image and user space packages are available on the following ULN channels:

- ol6\_latest (latest user space packages for Oracle Linux 6 other than DTrace, OFED, and DRBD packages)
- ol6\_UEK\_latest (latest user space packages for UEK other than DTrace, OFED, and DRBD packages)
- ol6\_x86\_64\_UEK\_R3\_latest (kernel-uek\*, dtrace-modules-\*, libdtrace-\*, and uname26)
- ol6\_x86\_64\_Dtrace\_userspace\_latest (dtrace-utils\*)
- ol6\_x86\_64\_ofed\_UEK (latest OFED tools packages)
- ol6\_x86\_64\_mysql-ha-utils (drbd84-utils)

You need to subscribe to ol6\_x86\_64\_UEK\_R3\_latest as well as ol6\_UEK\_latest and ol6\_latest. The other channels are needed for DTrace, OFED, and DRBD.

DTrace stands for Dynamic Tracing. OFED stands for OpenFabrics Enterprise Distribution. DRBD stands for Distributed Replicated Block Device.

## Switching from RHN to ULN

- RHEL uses the `rhn_register` program to register a system with Red Hat Network (RHN).
- Oracle has a `uln_register` program to switch to ULN if you are running RHEL6.
- Do the following:
  1. Download the following files from <https://linux-update.oracle.com/rpms>:
    - `uln_register.tgz`
    - `uln_register-gnome.tgz`
  2. Extract the packages.
  3. Install the packages.
  4. Create a ULN account at <https://linux.oracle.com/register>.
  5. Register your system with ULN by running `uln_register`.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Red Hat Enterprise Linux has an application called `rhn_register` to register a system with Red Hat Network (RHN). This application normally runs as part of the FirstBoot program on a new installation of RHEL. A system can be registered with RHN at a later time by running `rhn_register`.

Oracle makes it easy to switch from RHN to Unbreakable Linux Network (ULN). Oracle's `uln_register` program allows you to switch to ULN if you are running Red Hat Enterprise Linux 6. Details are available at <https://linux.oracle.com/switch.html>.

### 1. Download the Registration Packages

The packages that are required to register your system to ULN for i386 and x86\_64 architectures are available from <http://linux-update.oracle.com/rpms>. The file names to download are:

- `uln_register.tgz`
- `uln_register-gnome.tgz`

### 2. Extract the Registration Packages

Extract the `.tgz`, files on RHEL6 by using the following commands:

```
# tar -xzf uln_register.tgz  
# tar -xzf uln_register-gnome.tgz (only if rhn-setup-gnome is  
already installed)
```

### 3. Install the Registration Packages

After extracting the files, change to the `uln_migrate` directory and install the registration packages as follows:

```
# cd uln_migrate  
# rpm -Uvh *.rpm
```

### 4. Create a ULN Account

Before you can register a server, you must first create a ULN account. You can create a ULN account at <http://linux.oracle.com/register>.

### 5. Register Your System with ULN

To register your system, run the following command as the `root` user in a terminal window or on the command line:

```
# uln_register
```

Follow the instructions on the screen and provide the requested information. `uln_register` also collects machine information and uploads it to Oracle's server.

# Quiz

yum repositories can be both local and remote.

- a. True
- b. False

# Quiz

Errata are available only by subscribing to ULN.

- a. True
- b. False

## Summary

In this lesson, you should have learned how to:

- Describe Oracle Linux package management
- Use the `rpm` utility
- Describe the Oracle public `yum` server
- Describe and configure `yum` repositories
- Use the `yum` utility
- Describe the Unbreakable Linux Network (ULN)
- Describe the steps to switch from RHN to ULN

## Practice 6: Overview

The practices for this lesson cover the following topics:

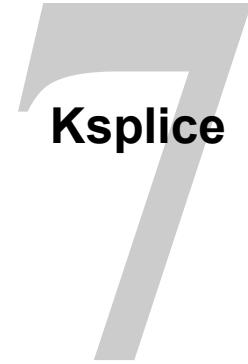
- Using the `rpm` utility
- Creating a local `yum` repository
- Using the `yum` utility
- Using the Unbreakable Linux Network



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe the purpose of Ksplice
- Describe how Ksplice works
- Describe Ksplice implementation steps
- View Ksplice packages on ULN
- Use Ksplice Uptrack commands
- Use the Ksplice web interface
- Use Ksplice Offline Client



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Introduction to Ksplice

Ksplice:

- Updates the kernel on a running system
- Applies the latest kernel security errata (CVEs)
  - Patches are effective immediately without rebooting.
- Does not halt the system
- Does not restart applications
- Applies updates in the background
- Requires Oracle Premier support subscription
- Works with both Unbreakable Enterprise Kernel and Red Hat Compatible Kernel
- Easy-to-use website



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

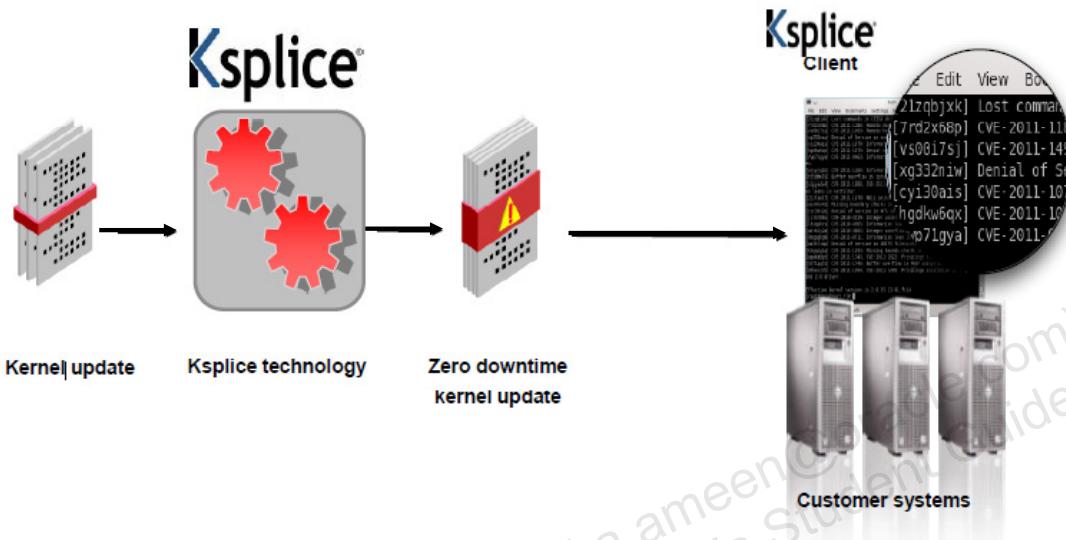
Oracle Linux customers with a Premier support subscription have access to Oracle Ksplice technology. Ksplice updates are available for systems running Oracle Linux 5 or later with either the Unbreakable Enterprise Kernel or the Red Hat Compatible Kernel.

Oracle Ksplice updates are kernel updates that can be applied on a running system. The Ksplice patches are applied to the running Linux kernel and are effective immediately. Oracle Ksplice patches only the running kernel. On subsequent reboot, these patches are applied at boot time.

This technology allows you to apply the latest kernel security errata (CVEs: Common Vulnerabilities and Exposures) without rebooting. It does not halt the system or restart applications; the updates are just applied in the background with a negligible impact, usually only a millisecond pause.

The Ksplice Uptrack website has an easy-to-use interface that lets you view registered systems, patches installed, available patches, and statuses of systems. It also lets you create access control groups. For each server, the website shows available updates that have not yet been applied. Each update lists a one-line description of the update. Further down the website is a list of the installed updates on your running kernel.

## How Ksplice Works



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

A kernel update comes from either Oracle or from the kernel community. The Ksplice team takes the update and works it into a binary patch that is inserted into a running kernel. You apply the patch by using the Ksplice tools, and the patch is up and running immediately.

Because you do not need to reboot or bring your system down, you can apply security updates as they become available without having to wait for your users to tell you that it is okay to take down the system. With Ksplice, you can keep your systems secure without jeopardizing high availability.

# Ksplice Implementation

A summary of actions to get started with Ksplice:

1. Register your system(s) with Unbreakable Linux Network (ULN).
2. Subscribe to the appropriate Ksplice channel.
3. Use the `yum` command to install the `uptrack` package.
4. Perform any required configuration.
5. View status from the System Status page of the Ksplice web interface at <https://uptrack.ksplice.com>.

After you register your system with ULN and install the `uptrack` package, you receive an email containing instructions for logging on to <https://uptrack.ksplice.com>.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The following is a summary of actions to get started with Ksplice:

1. Register your system(s) with Unbreakable Linux Network (ULN).
2. Subscribe to the appropriate Ksplice channel.

ULN channels that are available for Ksplice on Oracle Linux:

- `o15_i386_ksplice` – Oracle Ksplice clients, updates, and dependencies for Oracle Linux 5 on i386 systems
  - `o15_x86_64_ksplice` – Oracle Ksplice clients, updates, and dependencies for Oracle Linux 5 on x86\_64 systems
  - `o16_i386_ksplice` – Oracle Ksplice clients, updates, and dependencies for Oracle Linux 6 on i386 systems
  - `o16_x86_64_ksplice` – Oracle Ksplice clients, updates, and dependencies for Oracle Linux 6 on x86\_64 systems
3. Use the `yum` command to install the `uptrack` package.
  4. Perform any required configuration.
  5. View status from the System Status page of the Ksplice web interface at <https://uptrack.ksplice.com>.

# Ksplice Packages on ULN

The screenshot shows the Oracle ULN interface with the title "Unbreakable Linux Network". The navigation bar includes Home, Channels, Systems, Errata, CVE, and CSI Administration. The current location is Home > Channels > Channel: Ksplice for Oracle Linux 6 (x86\_64) > Packages. A sub-menu bar at the top of the page area shows "Channel Detail" and "Channel Packages". Below this is a search bar with a "Go" button. The main content is a table titled "Package" with two columns: "Package" and "Description". The table lists numerous packages related to uptrack and python-uptrack, such as "uptrack-1.2.14-0.el6.noarch" and "uptrack-updates-2.6.32-100.28.11.el6.x86\_64-20131014-0.noarch". The descriptions provide details about each package's function, such as being a client for the Ksplice Uptrack rebootless kernel update service or being a rebootless update for the Ksplice Uptrack rebootless kernel update service.

Package	Description
<a href="#">python-ksplice-uptrack-0.2.1-1.el6.noarch</a>	-
<a href="#">uptrack-1.2.14-0.el6.noarch</a>	Client for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-libyaml-0.1.3-1.el6.x86_64</a>	-
<a href="#">uptrack-offline-1.2.15.offline-0.el6.noarch</a>	Oracle Linux Support tool - offline client for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-python-cjson-1.0.5.el6.x86_64</a>	-
<a href="#">uptrack-PyYAML-3.08-4.el6.x86_64</a>	-
<a href="#">uptrack-updates-2.6.32-100.28.11.el6.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-100.28.15.el6.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-100.28.17.el6.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-100.28.9.el6.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-100.34.1.el6uek.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-100.35.1.el6uek.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-100.36.1.el6uek.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-100.37.1.el6uek.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-131.0.15.el6.x86_64-20131018-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-131.12.1.el6.x86_64-20131018-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-131.17.1.el6.x86_64-20131018-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-131.21.1.el6.x86_64-20131018-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-131.4.1.el6.x86_64-20131018-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-131.6.1.el6.x86_64-20131018-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-200.16.1.el6uek.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-200.19.1.el6uek.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
<a href="#">uptrack-updates-2.6.32-200.20.1.el6uek.x86_64-20131014-0.noarch</a>	Rebootless updates for the Ksplice Uptrack rebootless kernel update service

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This slide shows packages available from the “Ksplice for Oracle Linux 6 (x86\_64)” channel on ULN. Use the `yum` command to install the `uptrack` package:

```
# yum install uptrack
```

After the installation is complete, the tool automatically registers your system with the Uptrack service and checks for any available Oracle Ksplice updates for your running kernel. If new versions are available, the Uptrack tools provide you with the list of updates.

Your server must have access to the Internet. If a proxy is used, set the proxy in your shell before running the Uptrack commands. The commands to set the proxy are:

```
# export http_proxy=http://proxy.company.com port
# export https_proxy=http://proxy.company.com:port
```

The Uptrack configuration file location is `/etc/uptrack/uptrack.conf`. Modify this file to configure a proxy server, automatically install updates at boot time, or automatically check for new updates and apply them at the same time.

Oracle Ksplice patches are stored locally in `/var/cache/uptrack` and, by default, are automatically re-applied after a reboot (very early in the boot process). It is recommended that you also install the regular kernel RPM packages for released errata. This enables you to boot into a newer kernel version when you have a restart of the operating system. At that point, the Oracle Ksplice patches are applied starting from this new kernel as a baseline.

# Using Ksplice Uptrack

The screenshot shows a web browser window with the Oracle Ksplice logo at the top. The main title is "Take the Tour". Below it, a sub-section titled "Quick links" lists five items: 1. Quick links, 2. Uptrack command line tools, 3. Graphical interface, 4. Web interface, and 5. Monitoring. Under "Quick links", there is a bulleted list: • For a quick overview of Ksplice Uptrack's features, see our [features page](#). • For answers to commonly-asked questions, see our [FAQ](#). • Want to know more about how Ksplice works? Read our [whitepaper](#). • Ready to try it out? [Sign up](#) and [install Uptrack](#). Below this, there is a section titled "Uptrack command line tools" with a sub-section titled "uptrack-upgrade". It contains a code snippet and some explanatory text:

```
# uptrack-upgrade -y
The following steps will be taken:
Install [89g1kof] CVE-2018-4243: Denial of service due to wrong execve memory accounting.
Install [dfvnbz08] CVE-2018-4158: Kernel information leak in socket filters.
```

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This screen shows the “Take the Tour” page at <https://ksplice.oracle.com/uptrack/using>. It lists all the Ksplice Uptrack commands and provides examples and sample output. It describes how to configure your systems for automatic updates. It also describes access policies, how to configure email notifications, how to use the Uptrack API to monitor and control your Ksplice Uptrack account, and other features of Ksplice.

There is also a Ksplice User’s Guide available at [http://docs.oracle.com/cd/E37670\\_01/E39380/html/](http://docs.oracle.com/cd/E37670_01/E39380/html/).

## Ksplice Uptrack Command Summary

- **uptrack-upgrade:** Download and apply new updates.
- **uptrack-show:** List the active updates in your running kernel.
- **uptrack-remove:** Remove applied updates from the running system and return to the original kernel version.
- **uptrack-uname:** Display the effective kernel version based on active Oracle Ksplice updates.
- **uptrack-install:** This lets you install a specific update.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The following provides a summary of the Ksplice Uptrack Commands:

- **uptrack-upgrade:** Connect to the Uptrack update server and download new updates when available. These updates can be applied immediately as well.
- **uptrack-show:** List the active Oracle Ksplice updates in your running kernel.
- **uptrack-remove:** Remove applied updates from the running system and return to the original kernel version and state.
- **uptrack-uname:** The modified version of `uname` that knows how to read the effective kernel version based on active Oracle Ksplice updates
- **uptrack-install:** Install a specific update (if you don't want them all). This command is the opposite of `uptrack-remove`. The `uptrack-remove [id]` command takes an ID of the update to remove, and removes it. The `uptrack-install [id]` command takes the ID of an update to install, and installs it.

To remove the `uptrack` package, enter:

```
# yum remove uptrack
```

You can find the uninstall instructions on the Uptrack website as well.

# System Status

The screenshot shows the Ksplice System Status page. At the top, there are tabs for System Status, Group Management, Allow/Deny Policies, Settings, and Feedback and Support. Below the tabs, it says "Active Installations | Inactive Machines". The main area is titled "Oracle Internal Account - Overview". It displays an "Access key: [REDACTED]" with status information: 0 active machines are up to date, 1 active machine is out of date, and 3 machines have stopped using the Uptrack service. The "Active Installations" section lists one machine: dbhost.example.com (10.0.2.15) with 0 installed, 4 more updates available, and Auto Install set to No. The "Inactive Machines" section is empty. A note at the bottom says "To install Ksplice Uptrack on more systems, please see the [Installation instructions](#). To remove Ksplice Uptrack from a system, please see the [Removal instructions](#)".

**ORACLE®**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This screen shows the System Status page of the Ksplice web interface at <https://uptrack.ksplice.com>. All of your Ksplice-enabled systems are listed on this page. Systems listed under Active Installations are those systems that are connected to the Ksplice servers. Inactive Machines listed at the bottom of the page are those that have stopped using Ksplice Uptrack services.

The Machine column lists the individual systems. Click the machine name to display detailed information on available updates, installed updates, and additional information.

The Status in the example on the slide shows there are four updates available. Run the `uptrack-upgrade` command to apply the updates.

Auto Install is set to No. This is a configuration parameter, `autoinstall`, in the `/etc/uptrack/uptrack.conf` file. Setting this directive to yes automatically downloads and installs Ksplice updates by using cron.

The Original Kernel and Effective Kernel are the same because no updates have been installed. The last column gives the Ksplice Uptrack version.

# System Updated

The screenshot shows the Oracle Ksplice web interface. At the top, there's a navigation bar with links for System Status, Group Management, Allow/Deny Policies, Settings, and Feedback and Support. Below that is a sub-navigation bar with Active Installations and Inactive Machines. The main content area is titled "Oracle Internal Account - Overview". It displays an "Access key: [REDACTED]" and a summary: 1 active machine is up to date, 0 active machines are out of date, and 3 machines have stopped using the Uptack service. A section titled "Active Installations" lists one machine: dbhost example.com (10.0.2.15) is Up to date! (4 installed), Auto Install is No, Kernel product is Oracle Unbreakable Enterprise Kernel 2, Original Kernel is 2.6.39-100.5.1.el6uek, Effective Kernel is 2.6.39-100.7.1.el6uek, and Uptack version is 1.2.2. Below this, there are links for Installation instructions and Removal instructions. A section for "Inactive Machines" notes that three systems have stopped connecting to the Uptack service.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This screen shows the System Status page of the Ksplice web interface after running the `uptrack-upgrade` command. The Effective Kernel is now at a different level than the Original Kernel. The Effective kernel is shown using the `uptrack-uname` command. The original kernel version is shown using the `uname` command.

If the `install_on_reboot` parameter in the `/etc/uptrack/uptrack.conf` file is set to yes, the updates that were installed using Ksplice are re-installed after reboot by the `uptrack` service.

The Original and Effective Kernel only become the same if you installed the new kernel on disk by using `yum` and rebooted. If you reboot without updating the kernel on disk by using `yum`, you boot back into the old kernel and Ksplice applies its updates automatically at boot-time, bringing you back to the same Original and Effective Kernel versions you had before you rebooted (in this example, `2.6.39-100.5.1.el6uek` and `2.6.39-100.7.1.el6uek`).

## Ksplice Offline Client

- Systems running the Ksplice Offline Client do not need a network connection to the Oracle Uptrack server.
- Ksplice updates for each supported kernel version are bundled into an RPM and made available from ULN.
  - Ksplice updates for Oracle Linux 6 x86\_64 systems are available on the `ol6_x86_64_kssplice` ULN channel.
- Download the RPM from ULN to a memory stick and then use the `rpm` command to install the update package.
- Alternatively, create a Local Yum Server that acts as a Ksplice mirror.
  - Local Yum Server requires network connection to ULN.
  - Ksplice Offline Clients require access only to the Local Yum Server.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Ksplice Offline Client removes the requirement for a server to have a direct network connection to the Oracle Uptrack server. All available Ksplice updates for each supported kernel version are bundled into an RPM and made available from ULN. For example, the Ksplice updates for Oracle Linux 6 x86\_64 systems are available on the `ol6_x86_64_kssplice` ULN channel. The RPM for each kernel version at this channel is updated when a new Ksplice patch becomes available. You can download the RPM from ULN to a memory stick, for example, and then use the `rpm` command to install the update package.

You can also create a Local Yum Server that acts as a mirror of Ksplice for Oracle Linux channels on ULN. At regular intervals, download the latest Ksplice update packages from ULN to this Local Yum Server. Only the Local Yum Server requires access to the Oracle Uptrack server. After installing Ksplice Offline Client on your other systems, these systems need only to be able to connect to the Local Yum Server.

Systems that are running Ksplice Offline Client are not registered with <https://uptrack.kssplice.com>; therefore, you cannot use the Ksplice web interface or the Ksplice Uptrack API on these unregistered systems.

## Modifying a Local Yum Server to Act as a Ksplice Mirror

- Local Yum Server must be registered with ULN.
- Perform the configuration steps as follows:
  1. Log in to ULN at <http://linux.oracle.com>
  2. From the ULN Systems tab, click the link for your system in the list of registered machines.
  3. On the System Details page, click Edit.
  4. On the Edit System Properties page, verify that the Yum Server check box is selected and click Apply Changes.
  5. On the System Details page, click Manage Subscriptions.
  6. On the System Summary page, select needed channels.
  7. When you have finished selecting channels, click Save Subscriptions and log out of ULN.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

If you have a system that is registered with ULN that is also configured as a Local Yum Server, you can modify the system to act as a Ksplice mirror. Perform the following steps from the Local Yum Server:

1. Log in to ULN at <http://linux.oracle.com> with the ULN username and password that you used to register the system.
2. From the ULN Systems tab, click the link for your system in the list of registered machines.
3. On the System Details page, click Edit.
4. On the Edit System Properties page, verify that the Yum Server check box is selected and click Apply Changes.
5. On the System Details page, click Manage Subscriptions.
6. On the System Summary page, select channels from the list of available or subscribed channels and click the arrows to move the channels between the lists.
7. When you have finished selecting channels, click Save Subscriptions and log out of ULN.

For more information on available release channels, see

<http://www.oracle.com/technetwork/articles/servers-storage-admin/yum-repo-setup-1659167.html>.

## Updating a Local Yum Server with Ksplice Channels

- To update the Yum repositories for the registered channels, perform the following steps:
  1. Download the 167283.sh script from  
<http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/167283.sh>.
  2. Edit the 167283.sh script and set the value of the REP\_BASE variable to the base directory for the repository.
  3. Run the 167283.sh script to create the Yum repositories for the registered channels.
- It is recommended that you set up a cron job to run the 167283.sh script on a regular basis.
- The cron facility is discussed in the lesson titled “Automating Tasks.”



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

After subscribing to the Ksplice channel(s), perform the following steps to update the Yum repositories for the registered channels:

1. Download the 167283.sh script from  
<http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/167283.sh>.
2. Edit the 167283.sh script and set the value of the REP\_BASE variable to the base directory for the repository, for example:  
REP\_BASE=/var/yum
3. Run the 167283.sh script to create the Yum repositories for the registered channels.
  - a. To download only binary RPMs, enter:  
sh 167283.sh
  - b. To download both binary and source RPMs, enter:  
sh 167283.sh src

It is recommended that you set up a cron job to perform this task. For example, the following crontab entry for root runs the script twice per day at 6:00 AM and 6:00 PM:

```
0 6,18 * * * sh /var/downloads/yum/167283.sh
```

This example assumes that the 167283.sh script is located in the /var/downloads directory. The cron facility is discussed in the lesson titled “Automating Tasks.”

## Configuring Ksplice Offline Clients to Use the Local Ksplice Mirror

- Perform the following steps to configure a system as a Ksplice offline client:
  1. In the `/etc/yum.repos.d` directory, edit the existing repository file and disable all entries by setting `enabled=0`.
  2. In the `/etc/yum.repos.d` directory, create the `local-yum.repo` file.
  3. Install the Ksplice offline client with the following command:  
`# yum install uptrack-offline`
  4. Install the Ksplice updates that are available for the kernel:  
`# yum install uptrack-updates-`uname -r``
- The command given in step 4 installs the Ksplice updates and applies them. It is recommended that you set up a daily `cron` job to perform this task.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

When you have set up a Local Yum Server that can act as a Ksplice mirror, you can configure your other systems to receive Ksplice updates. Perform the following steps to configure a system as a Ksplice offline client:

1. In the `/etc/yum.repos.d` directory, edit the existing repository file such as `public-yumol6.repo` and disable all entries by setting `enabled=0`.
2. In the `/etc/yum.repos.d` directory, create the `local-yum.repo` file, which contains entries such as the following for an Oracle Linux 6 client:

```
[ol6_x86_64_ksplice]
name=Ksplice for $releasever - $basearch
baseurl=http://<IP_address_of_local_yum_server>/yum/OracleLinux
/OL6/ksplice/$basearch/
enabled=1
```

3. Install the Ksplice offline client:  
`# yum install uptrack-offline`
4. Install the Ksplice updates that are available for the kernel:  
`# yum install uptrack-updates-`uname -r``

# Quiz

Ksplice kernel patches are effective immediately without requiring a reboot.

- a. True
- b. False



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Summary

In this lesson, you should have learned how to:

- Describe the purpose of Ksplice
- Describe how Ksplice works
- Describe Ksplice implementation steps
- View Ksplice packages on ULN
- Use Ksplice Uptrack commands
- Use the Ksplice web interface
- Use Ksplice Offline Client



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 7: Overview

The practices for this lesson cover the following:

- Viewing the Ksplice Offline Client packages from the Unbreakable Linux Network (ULN)
- Using `sftp` to upload the Ksplice packages
- Installing the Ksplice Offline Client and kernel updates



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# 8

## Automating Tasks

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe available automated tasks utilities
- Configure cron jobs
- Describe cron directories and files
- Use the crontab utility
- Configure anacron jobs
- Use at and batch utilities

# Automating System Tasks

- Oracle Linux can run tasks automatically, and comes with automated tasks utilities: cron, anacron, at, batch.
- cron jobs can run as often as every minute.
  - A scheduled cron job is skipped if the system is down.
- anacron can run a job only once a day.
  - Scheduled jobs are remembered and run the next time that the system is up.
- crond searches multiple files and directories for scheduled jobs:
  - /var/spool/cron/
  - /etc/anacrontab
  - /etc/cron.d



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle Linux can be configured to automatically run tasks (or jobs) within a specified period of time, on a specified date, or when the system load average is below a specified number. You can use automated tasks to perform periodic backups, monitor the system, run custom scripts, and more. Oracle Linux comes with several automated tasks utilities.

## cron and anacron

Both cron and anacron are used to schedule the execution of recurring tasks according to a combination of the time, day of the month, month, day of the week, and week. cron allows jobs to be run as often as every minute. However, if the system is down when a cron job is scheduled, the job is not executed. anacron can run a job only once a day, but anacron remembers scheduled jobs and runs them the next time the system is up. The main purpose of anacron is to run cron jobs that did not run because the system was down. It is most useful on laptop computers to run the daily or weekly cron jobs after the system is booted with a particular delay to avoid overloading the system.

## crond Daemon

The crond daemon executes scheduled tasks. It searches /var/spool/cron for crontab files for individual users, /etc/anacrontab, and the files in the /etc/cron.d directory. It checks each command to see whether it should be run in the current minute. When a task is scheduled for execution, crond executes it as the user who owns the file describing the task.

# Configuring cron Jobs

- cron jobs are defined in /etc/crontab.
- Jobs are defined by:
  - minute: from 0 to 59
  - hour: from 0 to 23
  - day: from 1 to 31
  - month: from 1 to 12, or short name of month
  - day-of-week: from 0 to 7, or short name of day
  - user: user under which the jobs are run
  - command: shell command or script to execute
- Special characters can be used:
  - Asterisk, hyphen, comma, and forward slash



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

cron jobs are defined in the /etc/crontab configuration file. This is the systemwide crontab. Users can also have cron jobs. cron jobs are specified in the following format:

minute    hour    day    month    day-of-week    user    command

- **minute:** From 0 to 59
- **hour:** From 0 to 23
- **day:** From 1 to 31 (must be a valid day if a month is specified)
- **month:** From 1 to 12 (or the short name of the month such as Jan or Feb)
- **day-of-week:** From 0 to 7, where 0 or 7 represents Sunday (or the short name of the week such as Sun or Mon)
- **user:** The user under which the jobs are run
- **command:** The command to execute; can be a shell command or script

Other special characters may be used:

- An asterisk (\*) can be used to specify all valid values.
- A hyphen (-) between integers specifies a range of integers.
- A list of values separated by commas (,) specifies a list.
- A forward slash (/) can be used to specify step values.

## cron Examples

The following provides examples for specifying a cron job to run by minutes:

- \*/5 \* \* \* \* command: run command every five minutes
- \*/10 \* \* \* \* command: run command every 10 minutes
- \*/15 \* \* \* \* command: run command every 15 minutes
- 0-59/2 \* \* \* \* command: run command every other minute

The following provides examples for specifying a cron job to run by hours:

- 0 \*/2 \* \* \* command: run command every two hours
- 0 \*/3 \* \* \* command: run command every three hours
- 0 \*/4 \* \* \* command: run command every four hours
- 0 \*/5 \* \* \* command: run command every five hours

The following provides examples for specifying a cron job to run by days:

- 0 0 \* \* 5 command: run command at every Friday at midnight
- 0 0 \* \* 6 command: run command at every Saturday at midnight

The following provides examples for specifying a cron job to run by months:

- 0 0 1 5,10 \* command: run command first of May at midnight and first of October at midnight
- 0 0 1 \*/3 \* command: run command every third month at midnight

## Other cron Directories and Files

- /etc/cron.d
  - Contains files with same syntax as the /etc/crontab
  - root privileges only
- Other cron directories in /etc:
  - cron.hourly
  - cron.daily
  - cron.weekly
  - cron.monthly
- Scripts in these directories run hourly, daily, weekly, or monthly, depending on the name of the directory.
- /etc/cron.allow and /etc/cron.deny files restrict user access to cron.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The /etc/cron.d directory contains files that have the same syntax as the /etc/crontab file. Only root has permission to create and modify files in this directory.

```
# ls /etc/cron.d
0hourly  raid-check  sysstat
# cat /etc/cron.d/0hourly
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/
01 * * * * root run-parts /etc/cron.hourly
```

Also in /etc are cron directories named:

- cron.hourly
- cron.daily
- cron.weekly
- cron.monthly

Scripts in these directories run hourly, daily, weekly or monthly, depending on the name of the directory. Create entries in the /etc/anacrontab file to schedule the execution of these scripts. Example:

```
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45

# the jobs are started during the following hours only
START_HOURS_RANGE=3-22

#period in days    delay in minutes    job-identifier    command
1          5        cron.daily         nice run-parts /etc/cron.daily
7          25       cron.weekly        nice run-parts /etc/cron.weekly
@monthly  45       cron.monthly      nice run-parts /etc/cron.monthly
```

### Controlling Access to cron

The /etc/cron.allow and /etc/cron.deny files are used to restrict access to cron. The access control files are checked each time a user tries to add or delete a cron job. If the file cron.allow exists, only users listed in it are allowed to use cron, and the cron.deny file is ignored. If cron.allow does not exist, users listed in cron.deny are not allowed to use cron. If neither file exists, only root can use cron. The format of both access control files is one username on each line.

## crontab Utility

- The crontab utility allows users other than root to configure cron tasks.
- User-defined crontabs are stored in:
  - /var/spool/cron/<user>
- To create or edit a crontab:
  - Enter the command: crontab -e.
  - Use the same format as /etc/crontab without specifying a user.
- To list the contents of a user-defined crontab, enter:
  - crontab -l



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Users other than root configure cron tasks by using the crontab utility. All user-defined crontabs are stored in the /var/spool/cron/ directory and are executed by using the usernames of the users that created them.

To create or edit a crontab as a user, log in as that user and enter the command crontab -e. The file uses the same format as /etc/crontab with one exception, do not specify a user. When the changes to the crontab are saved, the crontab is stored according to username and written to the file /var/spool/cron/<username>. To list the contents of your own personal crontab file, use the crontab -l command.

The following set of commands illustrates the usage of the crontab utility (as root):

```
# crontab -e  
30 08 10 06 * /full-backup  
# ls /var/spool/cron  
root  
# crontab -l  
30 08 10 06 * /full-backup
```

## Configuring anacron Jobs

- anacron jobs are defined in /etc/anacrontab.
- Jobs are defined by:
  - Period in days: The frequency of execution in days
  - Delay in minutes: The minutes to wait before executing the job
  - Job-identifier: A unique name used in log files
  - Command: A shell command or script to execute



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

anacron jobs are defined in the /etc/anacrontab configuration file. The following is an example of the /etc/anacrontab file:

```
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=30
# the jobs are started during the following hours only
START_HOURS_RANGE=16-20
#period in days    delay in minutes    job-identifier    command
1          20      dailyjob            nice run-parts /etc/cron.daily
7          25      weeklyjob           /etc/weeklyjob.bash
@monthly  45      monthlyjob         ls /proc >> /tmp/proc
```

The first five lines are variables used to configure the environment in which the `anacron` tasks are run:

- **SHELL**: The shell environment to use
- **PATH**: The path used to execute commands
- **MAILTO**: The username to email output of the `anacron` jobs to
- **RANDOM\_DELAY**: The maximum number of minutes to be added to the delay in the `minutes` variable specified for each job (the minimum delay defaults to 6 minutes)
- **START\_HOURS\_RANGE**: An interval when scheduled jobs can be run

The remaining lines in the `/etc/anacrontab` file represent scheduled jobs:

- **period in days**: The frequency of execution of a job in days. This can be a macro. (@daily = 1, @weekly = 7, @monthly = once a month)
- **delay in minutes**: The number of minutes `anacron` waits, if necessary, before executing a job (0 = no delay)
- **job-identifier**: A unique name of a job used in the log files
- **command**: A command to execute (can be a shell command or a script)

Jobs defined in this `anacrontab` file are randomly delayed by 6–30 minutes and can be executed between 16:00 and 20:00.

The first job runs anywhere between 16:26 and 16:50 every day. The command executes all programs in the `/etc/cron.daily` directory (using the `run-parts` script, which takes a directory as a command-line argument and sequentially executes every program within that directory).

The second job runs once a week and executes the `/etc/weeklyjob.bash` script.

The third job is executed once a month and runs the `ls /proc >> /tmp/proc` command.

## at and batch

- at and batch are utilities for scheduling one-time tasks.
- The at command executes a task at a specific time.
- The batch command executes a task when system load average drops below 0.8.
- The atd service must be running to use at or batch.
- at command syntax:
  - at *time*
  - The *time* argument is the time to execute the command.
  - The *time* argument accepts multiple formats.
- batch command syntax:
  - batch (the at> prompt is displayed)



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The at command is used to schedule a one-time task at a specific time. The batch command is used to schedule a one-time task to be executed when the system load average drops below 0.8. The atd service must be running to use at or batch.

### Using at

To schedule a one-time job at a specific time, enter the command at *time*, where *time* is the time to execute the command. Use one of the following *time* arguments:

- HH:MM
- midnight: at 12:00 AM
- noon: at 12:00 PM
- teatime: at 4:00 PM
- month-name day year
- MMDDYY, MM/DD/YY, or MM.DD.YY
- now + time: time is in minutes, hours, days, or weeks (for example, now + 5 days )

After entering the at command with the time argument, the at> prompt is displayed. Type the command to execute, either a shell command or script, and press Enter. Multiple commands can be specified by typing each command followed by pressing Enter.

After typing all the commands, press Enter to go to a blank line and press Ctrl + D. You are emailed standard output and standard error from commands. Use the `atq` command to view pending jobs. Example:

```
# at now + 2 hours
at> /full-backup
at> <EOT>
Job 1 at 2011-10-12 16:52
# atq
1           2011-10-12 16:52 a root
```

### **Using batch**

`batch` is similar to `at` except commands or scripts are not executed until the load average is below 0.8. Type `batch` and the `at>` prompt is displayed. Enter multiple commands or scripts, pressing Enter after each entry. Press Ctrl + D on a blank line to end.

### **Controlling Access to at and batch**

The `/etc/at.allow` and `/etc/at.deny` files are used to restrict access to `at` and `batch`. The access control files are checked each time a user tries to use either command. Usage of these files is similar to the usage of the `cron.allow` and `cron.deny` files. The `root` user can always execute `at` and `batch` commands, regardless of the access control files. If the file `at.allow` exists, only users listed in it are allowed to use the commands and the `at.deny` file is ignored. If `at.allow` does not exist, users listed in `at.deny` cannot use the commands.

# Quiz

Both at and batch are used to schedule the execution of recurring tasks.

- a. True
- b. False



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Summary

In this lesson, you should have learned how to:

- Describe available automated tasks utilities
- Configure cron jobs
- Describe cron directories and files
- Use the crontab utility
- Configure anacron jobs
- Use at and batch utilities



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 8: Overview

The practices for this lesson cover the following:

- Automating tasks using the crontab and the at commands



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Kernel Module Configuration

9

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

- Describe loadable kernel modules
- Dynamically load and unload kernel modules
- Configure kernel module parameters

# Loadable Kernel Modules (LKM)

- LKMs extend the functionality of a running kernel.
- Kernel modules are dynamically loaded and unloaded.
- To list currently loaded kernel modules:
  - `lsmod`
- To view details about a specific kernel module:
  - `modinfo <module_name>`
- Kernel modules often have dependencies.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Linux kernel is loaded into memory by the boot loader. New code can be added to the kernel by including the source files in the kernel source tree and recompiling the kernel. But Linux also supports loadable kernel modules (LKM) that allow you to add code to a running kernel.

Kernel modules are dynamically loaded and unloaded on demand. They provide device drivers to allow the kernel to access new hardware, support for different file system types, and generally extend the functionality of the kernel.

## **Listing the Loaded Kernel Modules**

To list which kernel modules are currently loaded into the kernel, use the `lsmod` command. This command produces output by reading the `/proc/modules` file. Example:

```
# lsmod
Module           Size  Used by
autofs4          22739   3
ip_tables         15845   1 iptable_filter
parport          33812   2 ppdev,parport_pc
...
...
```

The output lists the name of the kernel module and the amount of memory the module uses. The “Used by” column gives the total number of processes that are using the module and the other modules that it depends on, followed by a list of those dependent modules. For example, the parport module depends on the ppdev and parport\_pc modules, which would be loaded before loading parport. These three modules are currently used by two processes.

### **module-init-tools Package**

The `lsmod` command and other kernel module files and utilities such as `modinfo`, `modprobe`, `depmod`, `insmod`, and `rmmod` are provided by the `module-init-tools` package:

```
# rpm -qf /sbin/lsmod
module-init-tools-3.9-21.0.1.el6.x86_64
```

To list all files provided by the `module-init-tools` package, enter:

```
# rpm -ql module-init-tools
/etc/depmod.d
/etc/depmod.d/dist.conf
...
...
```

### **Listing Module Details**

The `modinfo` command displays detailed information about a specific kernel module. For example, to display information about the `ipv6` kernel module, enter:

```
# modinfo ipv6
filename:          /lib/modules/3.8.13-
16.2.1.el6uek.x86_64/kernel/net/ipv6/ipv6.ko
alias:            net-pf-10
description:      IPv6 protocol stack for Linux
depends:
parm:             disable: Disable module such that it is non-functional
parm:             disable_ipv6: Disable IPv6 on all interfaces (int)
parm:             autoconf: Enable IPv6 address autoconfiguration
...
...
```

Description of the output includes:

- **filename:** The absolute path of the kernel object file
- **description:** The short description of the module
- **alias:** The internal alias names for the module, if any
- **depends:** A comma-separated list of modules that this module depends on, if any
- **parm:** The parameter name and a short description

Modules are loaded from the `/lib/modules/<kernel_version>/kernel` directory. For example, to display the absolute path of the `nfs` kernel object file, enter:

```
# modinfo -n nfs
/lib/modules/3.8.13-16.2.1.el6uek.x86_64/kernel/fs/nfs/nfs.ko
```

# Loading and Unloading Kernel Modules

- Use the modprobe command:
  - To load kernel modules:
    - modprobe <module\_name>
  - To unload kernel modules:
    - modprobe -r <module\_name>
- Kernel module dependencies are listed in /lib/modules/<kernel\_version>/modules.dep.
  - The file is created by depmod when kernel modules are installed.
- The Oracle-supplied kernel modules used in Grid Infrastructure are:
  - oracleacfs, oracleadv, oracleoks
- The directory to specify modules to load at boot time is:
  - /etc/sysconfig/modules



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Loading Kernel Modules

Kernel modules are loaded by using the modprobe command. The device manager for the Linux kernel, udev, uses modprobe to load drivers for automatically detected hardware.

For example, to load the nfs kernel module, enter:

```
# modprobe nfs
```

To list the newly-loaded module, enter:

```
# lsmod | grep nfs
```

nfs	266190	0	
lockd	66514	1	nfs
fscache	41704	1	nfs
nfs_acl	2477	1	nfs
auth_rpcgss	38928	1	nfs
sunrpc	203671	5	nfs, lockd, nfs_acl, auth_rpcgss

The nfs kernel module and many other kernel modules have dependencies. Dependent modules are loaded first. Some of these modules may, in turn, be dependent on other modules. You can use `modprobe -v` (verbose) to view the dependency resolution when loading a kernel module, as in the following example:

```
# modprobe -v nfs
...
insmod /lib/modules/3.8.13-
16.2.1.el6uek.x86_64/kernel/net/sunrpc/sunrpc.ko
insmod /lib/modules/3.8.13-
16.2.1.el6uek.x86_64/kernel/fs/lockd/lockd.ko
insmod /lib/modules/3.8.13-
16.2.1.el6uek.x86_64/kernel/fs/fscache/fscache.ko
insmod /lib/modules/3.8.13-
16.2.1.el6uek.x86_64/kernel/fs/nfs/nfs.ko
```

Note that `modprobe` uses the `insmod` command to load the modules into the kernel. Do not use `insmod`, however, because this command does not resolve dependencies.

Kernel module dependencies are listed in

`/lib/modules/<kernel_version>/modules.dep`. The `modprobe` command queries this file to determine dependencies. The `modules.dep` file is created by `depmod` when kernel modules are installed on your system.

## Unloading Kernel Modules

Unload kernel modules by using the `modprobe -r` command. You can also use the verbose option. For example, to unload the nfs kernel module, enter:

```
# modprobe -rv nfs
rmmmod /lib/modules/3.8.13-
16.2.1.el6uek.x86_64/kernel/fs/nfs/nfs.ko
rmmmod /lib/modules/3.8.13-
16.2.1.el6uek.x86_64/kernel/fs/fscache/fscache.ko
rmmmod /lib/modules/3.8.13-
16.2.1.el6uek.x86_64/kernel/fs/lockd/lockd.ko
...
rmmmod /lib/modules/3.8.13-
16.2.1.el6uek.x86_64/kernel/net/sunrpc/sunrpc.ko
```

Modules are unloaded in the reverse order, with the `nfs.ko` kernel module being unloaded first followed by the modules it was dependent on. Modules being used by a process or modules needed by other loaded modules are not unloaded.

The `modprobe -r` command uses `rmmmod` to unload the modules. But similar to `insmod`, it is not recommended to use `rmmmod` directly to unload kernel modules.

## ACFS and ADVM Drivers

The installation of the Oracle Grid Infrastructure (GI) stack also installs ACFS (ASM Cluster File System), and ADVM (ASM Dynamic Volume Manager) drivers and utilities. ASM (Automatic Storage Management) is a feature within the Oracle Database to simplify the management of database files. There are three drivers to support ACFS and ADVM. These drivers are dynamically loaded (in top-down order) by the OHASD (Oracle High Availability Service Daemon) process during Oracle Clusterware startup.

- **oracleooks.ko**: This is the kernel services driver, providing memory management, lock, and cluster synchronization primitives.
- **oracleadvm.ko**: The ADVM driver maps I/O requests against an ADVM Volume Device to blocks in a corresponding on-disk ASM file location. This ADVM driver provides volume management driver capabilities that directly interface with the file system.
- **oracleacfs.ko**: This is the ACFS driver that supports all ACFS file system file operations.

Use the `lsmod` command to list these kernel modules. Example:

```
# lsmod | grep oracle
Module           Size    Used by
oracleacfs      781476   5
oracleadvm      212736   9
oracleooks       224864   2 oracleacfs,oracleadvm
```

## Using the /etc/sysconfig/modules Directory

You can specify modules to be loaded at boot time by creating a file in the `/etc/sysconfig/modules` directory. Files in this directory must be executable shell scripts and the file name must end with `.modules`. These scripts are called from `/etc/rc.d/rc.sysinit`, which is executed at boot time. The lines in `rc.sysinit` that run scripts from this directory are:

```
# Load other user-defined modules
for file in /etc/sysconfig/modules/*.modules ; do
    [ -x $file ] && $file
done
```

The `/etc/sysconfig/modules/* .modules` could be a simple call to `modprobe`. Example:

```
#!/bin/sh
modprobe abc
```

Or the file could be more elaborate, as in the following example:

```
#!/bin/sh
if [ ! -c /dev/input/uinput ] ; then
    exec /sbin/modprobe uinput >/dev/null 2>&1
fi
```

# Kernel Module Parameters

- Pass parameters to a kernel module:
  - `modprobe <module_name> [parameter=value]`
- Configuration directory for modprobe:
  - `/etc/modprobe.d`
- Create `*.conf` files in `/etc/modprobe.d` to:
  - Specify options
  - Create aliases
  - Override normal modprobe behavior
  - Blacklist kernel modules
- You can also specify parameters in `/etc/modprobe.conf`.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Just as the kernel can accept boot time parameters to modify the behavior of the kernel, kernel modules can also accept parameters to modify their behavior. The syntax for passing parameters to a kernel module with modprobe is:

```
modprobe <module_name> [parameter=value]
```

Multiple `parameter=value` pairs can be passed by separating the pairs with spaces. Ensure that the module is not previously loaded, because `modprobe` does not reload the module.

## Configuration Directory for modprobe

The configuration directory for modprobe is `/etc/modprobe.d`. Files in this directory end with `.conf` and are used for the following purposes:

- To specify options used with kernel modules
- To create aliases or alternative names for a module
- To override the normal modprobe behavior for modules with special requirements

Earlier versions of modprobe had a configuration file named `/etc/modprobe.conf`, rather than a configuration directory. This file can be used if it exists. Use the same format for entries in both `/etc/modprobe.conf` and `/etc/modprobe.d/*.conf` files.

The format of these .conf files is one command per line. Valid commands to use in these files include the following:

```
alias, options, install, remove, blacklist
```

### **Alias**

Use the syntax `alias alias_name module_name` to create alternative names for kernel modules. You can also use shell wildcards in alias names. Example:

```
alias usbdevfs usbcore
```

### **Options**

Use the syntax `options module_name option(s)` to add options to `module_name`.

Example:

```
options b43 nohcrypt=1 qos=0
```

### **Install**

Use the syntax `install module_name command(s)` to tell `modprobe` to run shell commands rather than inserting the module in the kernel. Example:

```
install net-pf-6 /bin/true
```

### **Remove**

This is similar to the `install` command, except it is invoked when `modprobe -r` is run. Use the syntax `remove module_name command(s)` to tell `modprobe -r` to run shell commands rather than unloading the module from the kernel.

### **Blacklist**

Use the syntax `blacklist module_name` to tell `modprobe` to ignore a module's internal aliases. Internal aliases are those seen when using the `modinfo <module_name>` command. The `blacklist` keyword is typically used when the associated hardware is not needed, or when two or more modules both support the same devices, or a module invalidly claims to support a device.

# Quiz

Which of the following commands displays details about a kernel module?

- a. lsmod <module\_name>
- b. modinfo <module\_name>
- c. modprobe <module\_name>
- d. depmod <module\_name>

## Summary

In this lesson, you should have learned how to:

- Describe loadable kernel modules
- Dynamically load and unload kernel modules
- Configure kernel module parameters

## Practice 9: Overview

The practices for this lesson cover the following:

- Using loadable kernel modules



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# 10

## User and Group Administration

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe user and group implementation
- Describe user and group configuration files
- Configure users and groups by using command-line utilities
- Implement user private groups (UPG)
- Configure password aging and the hashing algorithm
- Use the User Manager GUI tool
- Manage the use of `su` and `sudo` commands
- Describe user and group implementation in the enterprise



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Introduction to Users and Groups

- User account information is stored in /etc/passwd.
- Group information:
  - Group information is stored in /etc/group.
  - Each user has a private group (UPG).
  - Users can belong to more than one group.
- Oracle Linux uses shadow passwords.
  - /etc/shadow: Hashed user passwords
  - /etc/gshadow: Hashed group passwords
  - /etc/login.defs: Security policies



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Each user in Linux has a unique user ID (UID), which is an ordinary integer number, and an associated username. Users log in using their usernames, but the system uses the associated UIDs. Each user account also has a home directory and a login shell. When users log in, they are placed in their home directory and their login shell executes. All of this user account information is stored in the /etc/passwd file.

Each user also belongs to one or more groups. Different users can be assigned to the same group. Access can be given to a group and all members of the group are granted the same access privileges. Each group account in Linux has a unique group ID (GID) and an associated group name. Group information is stored in the /etc/group file.

Oracle Linux uses a user private group (UPG) scheme. When a new user account is added, a new user private group is also created. The user private group has the same name as the user, and the new user is the only member of this group.

Both users and groups use shadow passwords. Passwords are hashed and stored in different files, /etc/shadow for users and /etc/gshadow for groups. Security improves by storing hashed passwords in “shadow” files, because these files are readable only by the root user. The use of shadow passwords also provides password aging parameters and allows security policies to be enforced, using the /etc/login.defs file.

Only the root user can add, modify, or delete user and group accounts.

# User and Group Configuration Files

- Contents of /etc/passwd:
  - username: placeholder: UID: GID: GECOS: home dir: shell
- Contents of /etc/shadow:
  - username: hashed password: password aging information
- Contents of /etc/group:
  - groupname: placeholder: GID: comma-separated members
- Contents of /etc/gshadow:
  - groupname: hashed password: GID: comma-separated administrators: comma-separated members
  - Group passwords are rarely used.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## /etc/passwd

When a new user is added, the information is stored as a single, colon-separated line in /etc/passwd. Here is an example of an entry in this file:

```
# tail -1 /etc/passwd
oracle:x:500:500:Oracle Linux Student:/home/oracle:/bin/bash
```

The following describes this entry:

- **oracle**: Username
- **x**: Indicates that shadow passwords are used
- **500**: UID, these begin with 500 and increment by 1 for each newly added user. UIDs below 500 are reserved for system use.
- **500**: GID of the user's primary group. These begin with 500 and increment by 1 for each new group. Users can belong to more than one group.
- **Oracle Linux Student**: GECOS (General Electric Comprehensive Operating System) information, used only for informational purposes such as full name
- **/home/oracle**: Home directory for this user
- **/bin/bash**: Default shell for this user

## /etc/shadow

With shadow passwords, a new entry is automatically added to /etc/shadow when a new user is created. This file can be viewed only by root. Here is an example of an entry in this file:

```
# tail -1 /etc/shadow
oracle:$6$LKIWOKV...:15283:0:99999:7:::
```

The following describes this entry:

- **oracle:** Username
- **\$6\$LKIWOKV...:** Hashed password value (partial value shown). The plain text password itself is not stored on the disk. An algorithm creates a unique string from a password.
- **15283:** Number of days since password has changed (counted in days since Jan 1, 1970).
- **0:** Number of days that need to pass before the password must be changed by the user.
- **99999:** Maximum number of days since the password changed that the password can be used. After this amount of days, the password must be changed by the user.
- **7:** Number of days before expire date that the user is warned about the pending password change policy. If the password is not changed after this number of days, the user account is locked.

The next field is empty but is used to store the last date when the account is locked (counted in days since Jan 1, 1970). The last field is also empty but is not used.

## /etc/group

Because Oracle Linux uses a UPG scheme, a new entry is automatically created in /etc/group when a new user is added. The group name is the same as the username. Here is an example of an entry in this file:

```
# tail -1 /etc/group
oracle:x:500:
```

The following describes this entry:

- **oracle:** Group name
- **x:** Indicates that shadow passwords are used
- **500:** GID

The last field is empty in this example, but it would list group members who do not have the group as a primary group. As mentioned, users can belong to more than one group. The GID stored in the user's entry in /etc/passwd is the user's primary group. Use this last field in the /etc/group file to indicate secondary group membership.

## /etc/gshadow

Hashed group passwords are stored in this file. However, group passwords are rarely used. Here is an example of an entry in this file:

```
# tail -1 /etc/gshadow
oracle:!:!:!
```

The following describes this entry:

- **oracle:** Group name
- **!:!:!**: Hashed password. The !! Indicates that the account is locked.

The last two fields are used to designate administrators and members.

## Adding a User Account

- Use the `useradd` command to add a user:
  - `useradd [options] user_name`
- Use the `passwd` command to create a password:
  - `passwd [options] user_name`
- User default settings are stored in:
  - `/etc/default/useradd`
- Use the `-D` option to display or modify defaults:
  - `useradd -D [options]`
- A new user's home directory is populated with files from:
  - `/etc/skel` directory
- To create a nologin user:
  - `useradd -s /sbin/nologin user_name`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

### `useradd`

Use the `useradd` command to add a user account. The syntax is:

```
useradd [options] user_name
```

When creating a new user without any options, the default settings are applied. Example:

```
# useradd jim  
# tail -1 /etc/passwd  
jim:x:501:501::/home/jim:/bin/bash
```

Also by default, `useradd` creates a locked user account. To unlock the account and assign a password, run the `passwd user_name` command as root. Example:

```
# passwd jim
```

The `passwd user_name` command prompts you for a new password. Depending on the complexity of the password, you may be notified the password is bad (too short or too simple). Re-enter the same password to continue and unlock the user account.

The same `passwd` command is used to change a password. The root user can always change a user's password. Users are prompted to enter the current password first.

## Default Settings

The default settings for a new user can be viewed and modified using the `-D` option. Example:

```
# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

The `INACTIVE` directive sets the number of days after a password expires until the account is locked. A value of `0` locks the account as soon as the password expires. A value of `-1` disables the feature. Contents of the `SKEL` (`/etc/skel` by default) are copied to a new user's home directory when the user account is created. Default settings are stored in `/etc/default/useradd`. The following options, used with `-D`, change the `useradd` command defaults:

- `-b default_home`: The initial path prefix for a new user's home directory
- `-e default_expire_date`: The date on which the user account is disabled
- `-f default_inactive`: The number of days after a password has expired before the account is locked
- `-g default_group`: The group name or ID for a new user's initial group
- `-s default_shell`: The new user's login shell

For example, to change a new user's login shell to the Bourne shell, enter the following:

```
# useradd -D -s /bin/sh
```

## useradd Options

Several options are available to the `useradd` command to override default settings: The following are some of the more commonly used options:

- `-c comment`: The new user's GECOS information, such as full name
- `-d home_dir`: The initial path prefix for a new user's home directory
- `-e expire_date`: The date (format `YYYY-MM-DD`) when the user account is disabled
- `-g initial_group`: The group name or number of the user's initial login group. The group name must exist. A group number must refer to an already existing group.
- `-G group`: A list of secondary groups that the user is also a member of. Each group is separated from the next by a comma, with no intervening whitespace.
- `-p passwd`: Set the new user's password.
- `-s shell`: The name of the user's login shell

For example, to create a new username of "mary", and include the user's name, and change the login shell to the C shell, enter the following:

```
# useradd -c "Mary Smith" -s /bin/csh mary
```

## **nologin Shell**

When you add a new user account, the user is granted shell access by default. You can create a user account with **nologin shell** for purposes of running a service such as SMTP, FTP, or running a web server, for example. A user without a login shell cannot log in to a system and, therefore, cannot run any commands interactively on the system. Processes can run as that user, however.

Logging in as a user with a **nologin shell** is politely refused and a message is displayed that the account is not available. If the file `/etc/nologin.txt` exists, **nologin** displays the file's contents rather than the default message.

To create a **nologin** user, first ensure that **nologin** exists in the `/etc/shells` file:

```
# cat /etc/shells  
/bin/sh  
/bin/bash  
/sbin/nologin  
/bin/tcsh  
/bin/csh
```

To add a new user called `test` with no shell access:

```
# useradd -s /sbin/nologin test
```

Attempting to log in as user `test` displays:

```
# su - test  
This account is currently not available.
```

# Modifying or Deleting User Accounts

- Use the `usermod` command to modify a user:
  - `usermod [options] user_name`
- Example: To add a user to a secondary group (GID=517):
  - `usermod -aG 517 user_name`
- Use the `userdel` command to delete a user:
  - `userdel [options] user_name`
- Options to `userdel` include:
  - `-f`: Force removal even if user is logged in
  - `-r`: Remove the user's home directory

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## `usermod`

Use the `usermod` command to modify an existing user account. The syntax is:

```
usermod [options] user_name
```

One of the most common uses of the `usermod` command is to add a user to another (secondary) group. Use the `-a` and `-G` options followed by a comma-separated list of the secondary groups to add the user to. The following example lists the contents of `/etc/group` before and after modifying a user and adding them to a secondary group:

```
# grep 517 /etc/group
students:x:517:
# usermod -aG 517 mary
# grep 517 /etc/group
students:x:517:mary
```

## `userdel`

Use the `userdel` command to delete a user account. Example:

```
# userdel mary
```

# Group Account Administration

- Use the groupadd command to add a group account:
  - groupadd [options] *group\_name*
- Use the groupmod command to modify a group account:
  - groupmod [options] *group\_name*
- Use the groupdel command to delete a group account:
  - groupdel *group\_name*
- Use the gpasswd command to administer group accounts:
  - gpasswd [options] *group\_name*
- Example: To add a user (jim) to a group (students):
  - gpasswd -a jim students
- The groups command prints the groups to which a user belongs.
- The newgrp command changes the real group identification.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## groupadd

Use the groupadd command to add a group account. The syntax is:

```
groupadd [options] group_name
```

## groupmod

Use the groupmod command to modify a group account. The syntax is:

```
groupmod [options] group_name
```

## groupdel

Use the groupdel command to delete a group account. The syntax is:

```
groupdel group_name
```

You can remove groups even if there are members in the group. You cannot remove the primary group of any existing user. You must remove the user before removing the group.

## gpasswd

Use the gpasswd command to administer /etc/group and /etc/gshadow. Every group can have administrators, members and a password. The syntax is:

```
gpasswd [options] group_name
```

## groups

The `groups` command displays the groups that a user belongs to. The following example illustrates that user `oracle` belongs to two groups, `oracle` (primary group) and `students` (secondary group):

```
$ grep oracle /etc/passwd
oracle:x:500:500:Oracle Student:/home/oracle/bin/bash
$ grep oracle /etc/group
oracle:x:500:
students:x:556:student1,student2,oracle
```

The `groups` command (logged on as `oracle`) verifies these group memberships.

```
$ whoami
oracle
$ groups
oracle students
```

## newgrp

The `newgrp` command executes a new shell and changes a user's real group identification. The following example illustrates the group ID before and after running the command. It also illustrates that a new shell is executed.

```
$ id
uid=500(oracle) gid=500(oracle) groups=500(oracle),566(students)...
```

Note that the gid equals 500 (oracle).

```
$ ps
 PID TTY      TIME CMD
20279 pts/0 00:00:00 bash
20411 pts/0 00:00:00 ps
$ newgrp students
$ id
uid=500(oracle) gid=566(students) groups=500(oracle),566(students)...
```

Note that the gid now equals 566 (students).

Also note that a new shell was executed:

```
$ ps
 PID TTY      TIME CMD
20279 pts/0 00:00:00 bash
20464 pts/0 00:00:00 bash
20486 pts/0 00:00:00 ps
```

The `newgrp` command does not recognize group ID numbers and you can only change your real group name to a group that you are a member of. Running the command without an argument sets the real group identification to the user's primary group.

## User Private Groups

- Each user belongs to a unique group.
  - Eliminates the need for `umask=0022`, which makes groups read-only
  - Allows `umask=0002`, which gives write permission to group
- Additional steps to implement:
  1. Create a directory to share.
  2. Create a new group.
  3. Add users to this new group.
  4. Change the group ownership for the directory.
  5. Set the `setgid` bit on the directory.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

As previously mentioned, Oracle Linux uses a user private group (UPG) scheme. Whenever new user accounts are added, they have a unique group. The purpose of this UPG scheme is to make Linux groups easier to use.

Traditionally on Linux systems, the `umask` was `0022`, which prevents other users and other members of a user's primary group from modifying a file. The following example illustrates the permissions on newly created files and directories when `umask` is set to `0022`:

```
$ umask  
0022  
$ mkdir project  
$ ls -ld project  
drwxr-xr-x. project  
$ touch project/testfile  
$ ls -l project/testfile  
-rw-r--r--. project/testfile
```

As you can see, the permissions for the group are read-only when `umask` is set to `0022`.

With UPG, all users have their own private group, so the group protection provided by setting umask to 0022 is not needed. With UPG, the umask is set to 0002 in /etc/profile and /etc/bashrc. Permissions on newly created files and directories are:

```
$ umask  
0002  
$ mkdir project  
$ ls -ld project  
drwxrwxr-x. project  
$ touch project/testfile  
$ ls -l project/testfile  
-rw-rw-r--. Project/testfile
```

Now, the group does have write permission on the newly created file and directory.

To allow multiple users write access to files within the same directory, create a new group, add the users to this new group, change the group ownership on this directory to the new group, and set the setgid bit on the directory. Files created in this directory will have the group permission set to the directory's group, rather than the primary group ID of the user who created the file.

For example, if you created a *project* group, added users to this *project* group, created a *project* directory, changed the group ownership for this *project* directory to the *project* group, set the setgid bit for the *project* directory, all *project* users will be able to edit the *project* files and create new files in the *project* directory. Any files these users create will retain their *project* group status. Other *project* users will always be able to edit these files.

## Password Configuration

- Password aging requires users to change their password.
- Use the chage command to configure password aging:
  - chage [options] user\_name
- Current values are displayed and changed interactively:
  - Minimum Password Age [0] :
  - Maximum Password Age [99999] :
  - Last Password Change [2011-11-06] :
  - Password Expiration Warning [7] :
  - Password Inactive [-1] :
  - Account Expiration Date [1969-12-31] :
- Use the authconfig command to configure the password hashing algorithm:
  - authconfig --passalgo=<algorithm> --update



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Password aging requires users to change their password periodically. Use the chage command to configure password expiration. The syntax is:

```
chage [options] user_name
```

Enter the chage command, followed by a username, to display existing password aging values and make modifications. For example, to display and change values for user frank, type (as user root):

```
# chage frank
Changing the ageing information for frank
Enter the new value, or press ENTER for the default
      Minimum Password Age [0] :
      Maximum Password Age [99999] :
      Last Password Change (YYYY-MM-DD) [2011-11-06] :
      Password Expiration Warning [7] :
      Password Inactive [-1] :
      Account Expiration Date (YYYY-MM-DD) [1969-12-31] :
```

Password aging information is stored in the /etc/shadow file. To view the user frank's entry before making any changes:

```
# grep frank /etc/shadow
frank:$6$XB1Um6w...:15284:0:99999:7:::
```

Changing the minimum password age value to 14 and maximum password age value to 30 means that in 14 days the user has 30 days to change their password. The new entry appears as:

```
# grep frank /etc/shadow
frank:$6$XB1Um6w...:15284:14:30:7:::
```

Based on this information, the user is warned to change his password seven days before the date the password expires.

The INACTIVE directive is used to set the number of days of inactivity after a password has expired before the user account is locked. Setting INACTIVE to -1 disables this feature.

### chage Options

A number of options are available for the chage command.

To list aging information:

```
# chage -l frank
Last password change : Nov 06, 2011
Password expires      : Dec 06, 2011
Password inactive     : never
Account expires        : never
Minimum number of days between password change : 14
Maximum number of days between password change   : 30
Number of days or warning before password expires: 7
```

To force a user to set a new password immediately (force immediate expiration), set the last password change value to 0. Example:

```
# chage -d 0 frank
```

After login, the user is prompted to change his password.

### authconfig

The Linux user password hashing algorithm is also configurable. Use the authconfig command to determine the current algorithm being used, or to set it to something different. To determine the current algorithm:

```
# authconfig --test | grep hashing
password hashing algorithm is sha512
```

To change the algorithm, use the --passalgo option with one of the following as a parameter: descrypt, bigcrypt, md5, sha256, or sha512, followed by the --update option. For example, to change the algorithm to MD5:

```
# authconfig --passalgo=md5 --update
```

## /etc/login.defs File

- The /etc/login.defs file provides default user account settings.
- Default values include:
  - Location of user mailboxes
  - Password aging controls
  - Values for automatic UID selection
  - Values for automatic GID selection
  - User home directory creation options
  - umask value
  - Encryption method used to encrypt passwords



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

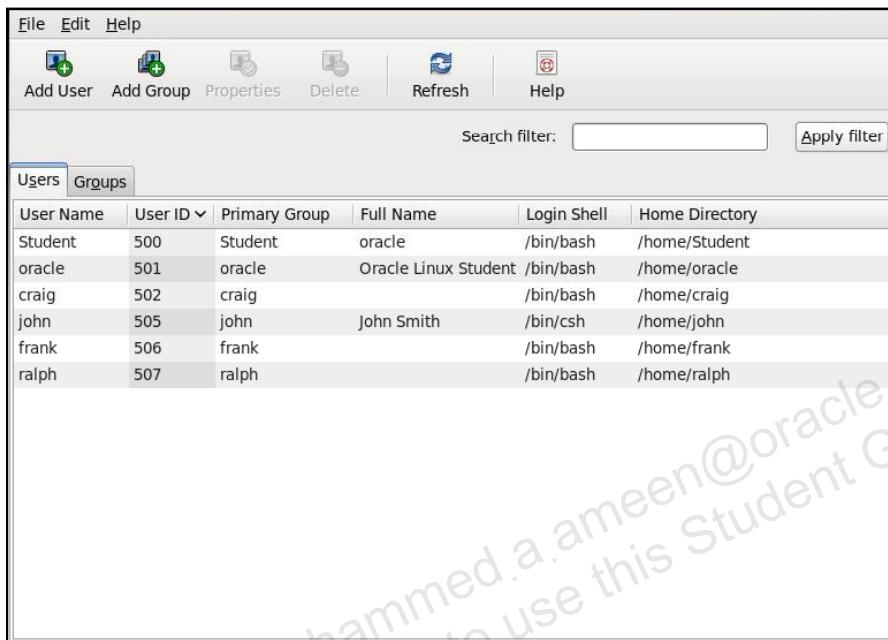
The /etc/login.defs file provides default configuration information for several user account parameters. The useradd, usermod, userdel, and groupadd commands, and other user and group utilities take default values from this file. Each line consists of a directive name and associated value. The following is a partial list of /etc/login.defs directives:

- Location of user mailboxes
- Password aging controls
- Minimum and maximum values for automatic UID selection (500 to 60000)
- Minimum and maximum values for automatic GID selection (500 to 60000)
- Whether home directories should be created when adding a new user
- Default umask
- Encryption method used to encrypt passwords

If the USERGROUPS\_ENAB directive in /etc/login.defs is set to YES, a group is created for the user with the same name as the username. If the directive is set to NO, the useradd command will set the primary group of the new user to the value specified by the GROUP directive in the /etc/default/useradd file, or 100 by default.

# User Manager Tool

The `system-config-users` command starts User Manager.



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The slide shows the graphical tool used to perform user and group account administration. If you have the X Window System installed (run level 5), the User Manager application provides a GUI to view, modify, add, and delete local users and groups. Use the `system-config-users` command to start the tool.

Two tabs are available, a Users tab for user administration, and a Groups tab for group administration. To add a user or group, click the Add User or Add Group button. To modify an existing user or group, select the entry from the list and click the Properties button. Select an entry from the list and click the Delete button to delete a user or group account.

A Search filter is available to find a specific user or group. Enter the first few letters of the name in the “Search filter” field and click the “Apply filter” button. You can also sort on any column by clicking the column header.

## Restricting Use of the su Command

- You can limit access to the `su` command to only those users who are members of the `wheel` group.
- To limit access to the `su` command to the `oracle` user, add the `oracle` user to the `wheel` group as follows:
  - `usermod -aG wheel oracle`
- Add the following line to the `/etc/pam.d/su` file to only permit root access to members of the `wheel` group:
  - `auth required pam_wheel.so use_uid`
- Pluggable Authentication Modules (PAM) is discussed in Lesson 16.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

You can limit access to the `su` command to only those users who are members of the `wheel` group. Two steps are needed to limit user access to the `su` command:

1. Add user(s) to the `wheel` group.
2. Add an entry to the `/etc/pam.d/su` file.

For example, to limit access to the `su` command to the `oracle` user, add the `oracle` user to the `wheel` group as follows:

```
# usermod -aG wheel oracle
```

Pluggable Authentication Modules (PAM) is discussed further in the lesson titled “Pluggable Authentication Modules (PAM)” but PAM is an authentication mechanism that allows you to configure how applications use authentication to verify the identity of a user. Add the following line to the `/etc/pam.d/su` file to permit only root access to members of the `wheel` group:

```
auth required pam_wheel.so use_uid
```

The preceding entry is in the `/etc/pam.d/su` file by default, but is commented out. Users using the `su` command still need to know the root password. The following line allows members of the `wheel` group to `su` to root without knowing the password:

```
auth sufficient pam_wheel.so trust use_uid
```

## Allowing Use of the sudo Command

- sudo privileges are configured in the /etc/sudoers file.
- The following entry is present in the /etc/sudoers file by default:
  - root ALL=(ALL) ALL
- The following entry in /etc/sudoers allows the oracle user to use sudo to run administrative commands:
  - oracle ALL=(ALL) ALL
- The oracle user can now run administrative commands by preceding the command with sudo, for example:

```
$ sudo useradd new_user  
[sudo] password for oracle:
```
- You are prompted for the oracle user password, not the root user password.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

You can also allow a regular user to run a command as root by preceding the administration command with sudo. The following example uses sudo to allow a regular user to add a new user account. You are prompted for your regular user password, not the root password.

```
$ sudo useradd new_user
```

The sudo privileges are configured in the /etc/sudoers file. The following entry is present in the /etc/sudoers file by default:

```
root ALL=(ALL) ALL
```

The “ALL=(ALL)” argument gives the root user permission to run from any terminal, acting as any user. The ending “ALL” argument indicates that the root user can execute any command.

Use the visudo command to edit the /etc/sudoers file. This command locks the /etc/sudoers file against simultaneous edits. To authorize the oracle user to run any command with root privileges, create the following entry in the /etc/sudoers file:

```
# visudo  
oracle ALL=ALL
```

The oracle user can now run administration commands by preceding them with sudo.

# User/Group Administration in the Enterprise

- User and group account information is often centralized.
- Centralized information can be retrieved using:
  - Lightweight Directory Access Protocol (LDAP)
  - Network Information Service (NIS)
- User home directories can also be centralized and accessed remotely.
  - Remote file systems can be auto-mounted.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

In enterprise environments with possibly hundreds of servers and thousands of users, user and group account information can be stored in a central repository rather than in files on several servers. You can configure user and group information on a central server and retrieve this information by using services such as Lightweight Directory Access Protocol (LDAP) and Network Information Service (NIS). You can also create users' home directories on a central server and automatically mount, or access, these remote file systems.

## LDAP (Lightweight Directory Access Protocol)

LDAP is a set of open protocols used to access information stored remotely over a network. LDAP is commonly used for centrally managed users and groups, user authentication, or system configuration.

## Network Information Service (NIS)

NIS is a directory service that provides a centralized location for usernames, group names, and host names. NIS simplifies the maintenance of these common administrative files by keeping them in a central database. As with LDAP, other systems on the network contact the central server to retrieve information.

# Quiz

Oracle Linux implements shadow passwords and user private groups.

- a. True
- b. False



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Quiz

Which of the following statements are true?

- a. User account information is stored in /etc/passwd.
- b. Group account information is stored in /etc/group.
- c. Password aging information is stored in /etc/passwd.
- d. Default settings for a new user are stored in /etc/default/useradd and /etc/login.defs.

## Summary

In this lesson, you should have learned how to:

- Describe user and group implementation
- Describe user and group configuration files
- Configure users and groups by using command-line utilities
- Implement user private groups (UPG)
- Configure password aging and the hashing algorithm
- Use the User Manager GUI tool
- Manage which users can use the `su` and `sudo` commands
- Describe user and group implementation in the enterprise



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 10: Overview

The practices for this lesson cover the following:

- User account administration
- Group account administration
- User private groups
- Password aging
- Using the User Manager GUI
- Restricting use of the `su` command
- Allowing use of the `sudo` command

# 11

## File Systems

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe disk partitioning
- Use disk partitioning utilities
- Describe supported file system types
- Describe file system creation, mounting, and maintenance
- Describe and maintain swap space

# Disk Partitions

- Partitioning divides a disk drive into logical disks.
  - Each partition is treated as a separate disk.
  - A partition table defines the partitions.
- Minimum recommended partitions (file systems):
  - / (root)
  - /boot
  - swap
- Create additional partitions to simplify administration.
- Extended partitions allow the creation of more than four primary partitions.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Partitioning divides a disk drive into one or more logical disks. Each partition is treated as a separate disk with its own file system. Partition information is stored in a partition table.

Oracle Linux needs at least one partition for its root file system. It is also recommended to create a second partition dedicated as a swap partition. On Intel-compatible hardware, the BIOS that boots the system can often only access the first 1024 cylinders of the disk. For this reason, a third partition is often created as a boot partition to store the kernel image and a few other files needed at boot time.

Additional partitions can be created to simplify administration and backups, to increase system security, and to accommodate other needs, such as testing. For example, data that frequently changes, such as user home directories, databases, and log file directories, is typically created on separate partitions to facilitate backups.

## Primary and Extended Partitions

The original partitioning scheme for PC hard disks allowed only four partitions, called primary partitions. To create more than four partitions, one of these four partitions can be divided into many smaller partitions, called logical partitions. When a primary partition is subdivided in this way, it is known as an extended partition. The partitioning tools presented in this lesson allow you to create primary or extended partitions.

During installation in the practices for the “Installing Oracle Linux” lesson, three primary partitions were created:

- xvda1, mounted on /boot
- xvda2, mounted on /
- xvda3, mounted on /home

The fourth partition (xvda4) was created as an extended partition, allowing it to be subdivided into logical partitions. One logical partition (xvda5) was created on the extended partition that was designated as a swap partition.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Partition Table Manipulation Utilities

- Three partition utilities are presented in this lesson:
  - fdisk
  - cfdisk
  - parted
- Do not partition a device while it is in use.
- Ensure that file systems are unmounted:
  - Use the `umount` command.
- Ensure that swap space is disabled:
  - Use the `swapoff` command.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Various utilities are available to display and manipulate the partition table. Three of these utilities are presented in this lesson:

- fdisk
- cfdisk
- parted

If the file system is greater than 2 TB, you cannot use `fdisk` but you must use `parted`.

When removing or resizing a partition by using any partition table manipulator command, the device on which that partition resides must not be in use. Also *not* recommended is creating a new partition on a device that is in use. Unmount the file system partitions and disable the swap partition before making any changes to the partition table.

Use the `umount` command to unmount file systems, and use the `swapoff` command to disable the swap space. These commands are discussed in more detail later in this lesson.

# **fdisk Utility**

- The **fdisk** utility is a partition table manipulator for Linux.
- Use the **fdisk -l** option to list the partition table.
  - Device: Lists the partitions
  - Boot: \* indicates that the partition contains boot files
  - Start and End: The starting and ending cylinders
  - Blocks: The number of blocks allocated to the partition
  - Id and System: The partition type
- Partition naming (example: /dev/sda1)
  - /dev: The directory containing device files
  - sd: SCSI disk; hd: IDE disk; xvd: Virtual disk
  - a: First disk; b: Second disk; c: Third disk
  - 1: First partition; 2: Second partition; 3: Third partition

**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The **fdisk** utility is a common partition table manipulator for Linux. Use **fdisk -l** to list the partition table. Output varies depending on the number of attached disks and partitions. To display the partition for a specific device, include the device name as an argument. For example:

```
# fdisk -l /dev/xvda
Disk /dev/xvda: 12.9 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000238a2

      Device  Boot   Start     End   Blocks   Id  System
  /dev/xvda1    *       1      13    102400   83  Linux
  ...

```

Without specifying a device as an argument, partitions in **/proc/partitions** are listed.

The first five lines of output from the `fdisk -l /dev/xvda` command are summary information about the device itself, `/dev/xvda`. The example output shows a 12.9 GB virtual disk (`xvda`) with 63 sectors per track. With zone density recording (ZDR), a disk surface is divided into sixteen circumferential zones. The number of sectors for each track is different in each zone, with the outermost zone containing the most sectors and the innermost zone containing the smallest number of sectors. Therefore, if a partition spans zones, the number of sectors per track would not be the same.

The partition table is displayed after the summary information. Seven columns of information are listed in the partition table. The Device column shows five partitions: `/dev/xvda1`, `/dev/xvda2`, `/dev/xvda3`, `/dev/xvda4`, and `/dev/xvda5`. The Boot column shows that the first partition, `/dev/xvda1`, has an asterisk (\*) indicating that this partition contains the files required by the boot loader to boot the system. The Start and End columns list the starting and ending cylinders of each partition. The Blocks column lists the number of blocks allocated to the partition. The Id and System columns identify the partition type.

### Partition Types

The partition types can be displayed and changed by using the `fdisk` utility. A partial list of partition types are:

- 83: Linux
- 82: Linux swap
- 5: Extended
- 8e: Linux LVM

### Partition Naming

The Linux partition naming scheme is in the `/dev/xxYN` form. Elements of this naming scheme are described as follows:

- `/dev/`  
This is the directory in which all device files reside.
- `xx` (or `xxx`)  
The first two of three letters indicate the type of device on which the partition resides.  
These letters are usually `hd` (for IDE disks), `sd` (for SCSI disks), or `xvd` (for virtual disks).
- `Y`  
This letter indicates which device the partition is on—for example, `/dev/sda` (the first SCSI hard disk) or `/dev/xvdb` (the second virtual disk).
- `N`  
This number indicates the partition. For example, `/dev/sdb1` is the first partition on the second SCSI device and `/dev/xvda3` is the third partition on the first virtual disk.

## Using the `fdisk` Utility

- The `fdisk` utility provides an interactive interface.
- Basic `fdisk` commands include:
  - d: Delete a partition.
  - l: List the known partition types.
  - m: Print the available commands.
  - n: Add a new partition.
  - p: Print the partition table.
  - q: Quit without saving changes.
  - w: Write the table to disk and exit `fdisk`.
- To have the kernel re-read the partition table, use `partprobe device`.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `fdisk` utility also provides an interactive interface for manipulating the partition table of a disk device. `fdisk` understands DOS type partition tables and BSD or SUN type disk labels. `fdisk` does not understand the GUID Partition Table (GPT).

To use the interactive interface, enter the `fdisk` command followed by the device name:

```
# fdisk /dev/xvdd
```

If the device has a GPT, the following message is displayed:

```
WARNING: GPT (GUID Partition Table detected on '/dev/xvdd' ! The
util fdisk doesn't support GPT. Use GNU Parted.
```

Another warning message is displayed:

```
WARNING: DOS-compatible mode is deprecated. It's strongly
recommended to switch off the mode (command 'c') and change
display units to sectors (command 'u').
```

After the warning message, the `fdisk` command prompt appears:

```
Command (m for help) :
```

Enter `m` to display the `fdisk` commands.

## fdisk Commands

```
Command (m for help): m
a      toggle a bootable flag
b      edit bsd disklabel
c      toggle the dos compatibility flag
d      delete a partition
l      list known partition types
m      print this menu
n      add a new partition
o      create a new empty DOS partition table
p      print the partition table
q      quit without saving changes
s      create a new empty sun disklabel
t      change a partition's system id
u      change display/entry units
v      verify the partition table
w      write table to disk and exit
x      extra functionality (experts only)
```

To list the current partition table, enter p at the fdisk command prompt:

```
Command (m for help): p
Disk /dev/xvdd: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
output omitted...
      Device Boot Start      End      Blocks   Id  System
/dev/xvdd1            1    1959    15727616   83  Linux
```

The following example illustrates the steps to delete the existing partition and create two new partitions. The first new partition is 5 GB and the second partition uses all remaining space:

```
Command (m for help): d
Command (m for help): n
Command action
      e      extended
      p      primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1958, default 1): 1
Last cylinder, +cylinders or +size{K,M,G} (1-1958, default
1958): +5G
```

```
Command (m for help): p
Disk /dev/xvdd: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
output omitted...
      Device  Boot  Start      End      Blocks   Id  System
  /dev/xvdd1            1       654     5253223+   83  Linux

Command (m for help): n
      e      extended
      p      primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (655-1958, default 655): <ENTER>
Using default value 655
Last cylinder, +cylinders or +size{K,M,G} (1-1958, default
1958): <ENTER>
Using default value 1958

Command (m for help): p
      Device  Boot  Start      End      Blocks   Id  System
  /dev/xvdd1            1       654     5253223+   83  Linux
  /dev/xvdd2            655     1958    10474380   83  Linux
```

After creating the new partition table, enter the w command to write the table to the disk and exit the fdisk utility.

```
Command (m for help): w
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
Syncing disks.
```

```
#
```

### **partprobe Command**

This command informs the kernel of partition table changes. Run this command with the device name as an argument to require the operating system to re-read the partition table:

```
# partprobe /dev/xvdd
```

Use the -s option to display a summary of the device and partitions:

```
# partprobe -s /dev/xvdd
/dev/xvdd: msdos partitions 1 2
```

## cfdisk Utility



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This screenshot in the slide shows the user interface of the `cfdisk` utility, which is used to create, delete, and modify partitions on a disk device. Enter the `cfdisk` command and include the device that you want to partition as an argument. Example:

```
# cfdisk /dev/xvdd
```

Summary information for the disk device is displayed at the top of the window. The partition table is displayed in the middle of the window. Selectable commands are displayed in brackets at the bottom of the window.

The current partition is highlighted. Use the up and down arrow keys to select a partition from the list. Use the right and left arrows to select a command. All partition-specific commands apply to the current partition.

In the slide's example, all available space on the device is currently allocated. A [New] menu option is available when the device has unallocated space.

## parted Utility

- The parted utility provides a command-line interface:
  - parted [option] device [command [argument]]
- The parted utility also has an interactive mode:
  - parted device
- Interactive mode displays a (parted) prompt.
  - Enter help to view a list of available commands.
  - Enter help *command* to view detailed help on a specific command.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The GNU parted utility is also used to view the existing partition table, change the size of existing partitions, or add partitions from free space or additional hard drives. This utility is more advanced than the fdisk utility. It supports more disk label types and offers additional commands. parted syntax is:

```
parted [option] device [command [argument]]
```

Use parted interactively to enter commands one at a time. Include only the device as an argument to invoke interactive mode. Example:

```
# parted /dev/xvdd
GNU Parted 2.1
Using /dev/xvdd
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

From the (parted) prompt, enter a command or type help to view the list of available commands. Get additional help on a specific command by typing help plus the command. Example:

```
(parted) help mkpart
```

Entering `print` displays the partition table:

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdd: 16.1GB
Sector size (logical/physical): 512/512
Partition Table: msdos
Number  Start   End     Size    Type      File system  Flags
 1      32.3kB  5379MB  5379MB  primary
 2      5379MB  16.1GB   10.7GB  primary
```

The following example illustrates the steps to create a new partition table by using the `mklabel` command:

```
(parted) mklabel
New disk label type? gpt
```

The disk label type must be one of the following: aix, amiga, bsd, dvh, gpt, mac, msdos, pc98, sun, or loop. The following warning message is displayed:

```
Warning: The existing disk label on /dev/xvdd will be destroyed
and all data on this disk will be lost. Do you want to continue?
Yes/No?
```

Responding “Yes” creates a new GUID Partition Table and returns the `(parted)` prompt. Use the `mkpart` command to create a new partition:

```
(parted) mkpart
Partition name? []
File system type? [ext2] ?
Start? 1
End? 5GB
(parted)
```

With a GPT disk label, you are first prompted to optionally give the partition a name. You are not required to name your partitions. Next, you are prompted for a file system type. A large number of file system types are supported. You are then prompted for the Start and End parameters for the partition.

To display the new partition, enter the `print` command. Example:

```
(parted) print
Number  Start   End     Size    File system  Name  Flags
 1      1049kB  5000MB  4999MB  ext4
```

Notice that the columns of output differ depending on the type of partition table. To exit the `parted` utility, enter `quit`:

```
(parted) quit
```

# File System Types

- ext2
  - High performance for fixed disk and removable media
- ext3
  - Journaling version of ext2
- ext4
  - Supports larger files and file system sizes
- vfat
  - MS-DOS file system useful when sharing files between Windows and Linux
- Btrfs
  - Addresses scalability requirements of large storage systems



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

After partitioning the disk device, you still need to make a file system on the partitions. Making a file system writes information to the drive and creates order of the empty space. This file system-related data consumes a small percentage of the space. The remaining space on the disk drive is split into small, consistently sized segments called blocks.

Oracle Linux supports a large number of file system types, some of which are described as follows.

## **ext2**

The second extended file system was introduced in Linux in January 1993. ext2 supports a maximum file system size of 8 TB and a maximum file size of 2 TB. ext2 file systems are susceptible to corruption during a power failure or any unclean system shutdown. In this case, mounted ext2 file systems must be checked for consistency. This consistency check causes delays in boot time, especially when file systems contain a large number of files.

## **ext3**

The ext3 file system is an improvement on the ext2 file system and includes journaling capabilities. The journaling provided by the ext3 file system improves reliability and availability. The consistency checks required by ext2 file systems after an unclean shutdown are not necessary with ext3. ext3 supports a maximum file system size of 16 TB and a maximum file size of 2 TB. ext2 file systems are upgradeable to ext3 without reformatting.

A file system journal first performs a write operation in a journal. Then it performs the write on the file system itself and removes the entry from the journal. Journaling ensures that the file system is able to recover from power failures or system crashes by recovering from the journal and removing any incomplete entries.

#### **ext4**

The ext4 file system is a scalable successor to ext3. ext4 supports very large file systems and files, extents (contiguous physical blocks), pre-allocation, delayed allocation, faster file system checking, more robust journaling, and other enhancements. ext4 supports a maximum file system size of 1 EB and a maximum file size of 16 TB.

Journal options for ext3 and ext4 file systems can be specified on the command line by using `-J <journal-options>`. Journal options are comma-separated and may take an argument using the equals (=) sign. The following journal options are supported:

- **size=journal-size**: This creates an internal journal (stored inside the file system) with the size specified in megabytes.
- **device=external-journal**: This attaches the file system to the journal block device located on `external-journal`. The external journal must already have been created using the `mke2fs -O journal_dev external-journal` command.

#### **vfat**

The vfat file system (also known as FAT32) is an MS-DOS file system. It is supported by Linux but is not journaled and lacks many of the features available with the ext file system types. Because vfat file systems are readable by both Windows and Linux, they are useful for exchanging data between these operating systems.

#### **Btrfs**

The Btrfs (B-Tree file system) is a copy-on-write file system for Linux designed to address the expanding scalability requirements of large storage subsystems. Btrfs provides extent-based file storage with a maximum file size of 16 EB. Btrfs offers the ability to create both readable and writable snapshots and the capability to roll back to a prior, known-good state. Btrfs includes checksum functionality to ensure data integrity, as well as transparent compression to save space. Btrfs includes integrated logical volume management operations making it easy to add and remove capacity and to use different RAID levels.

Btrfs is not included in the list of file system types presented in the installation choices. To create a Btrfs file system during installation, you must initiate the installation with the boot option `btrfs`. To access the boot prompt, press the Esc key while the graphical boot screen is displayed. From the boot prompt, type the following command to create a Btrfs file system during installation:

```
boot: linux btrfs
```

A boot ISO is available starting with Oracle Linux 6.3 that boots up Unbreakable Enterprise Kernel as the install kernel and uses btrfs as the default file system for installation.

# Making File Systems

- Use the `mkfs` command to build a Linux file system:
  - `mkfs [options] device`
- The `mkfs` command is a wrapper for other utilities:
  - `mkfs.ext2`
  - `mkfs.ext3`
  - `mkfs.ext4`
- Defaults parameters are specified in `/etc/mke2fs.conf`.
- Use the `blkid` command to display block device attributes.
- Use the `e2label` command to display and modify the file system label.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The command to build a Linux file system on a device, or hard disk partition, is `mkfs`. The syntax for the command is:

```
mkfs [options] device
```

The `mkfs` command is actually a front end for the different file system builder utilities such as `mkfs.ext2` and `mkfs.ext4`. These utilities are executable directly from the command line. When using the `mkfs` wrapper, include the `-t fstype` option to specify the type of file system to be built. If not specified, the default file system type, `ext2`, is created.

To see which supported file system types are installed, use the `ls /sbin/mk*` command:

```
# ls /sbin/mk*
/sbin/mkdosfs  /sbin/mkfs.cramfs  /sbin/mkfs.ext4dev
/sbin.mkinitrd  /sbin/mkdumprd  /sbin/mkfs.ext2
/sbin/mkfs.msdos  /sbin/mkswap  /sbin/mke2fs  /sbin/mkfs.ext3
/sbin/mkfs.vfat  /sbin/mkfs  /sbin/mkfs.ext4
/sbin/mkhomedir_helper
```

Not all of these `mk*` files are used to create file systems. For example, `mkinitrd` is used to create an initial image used by the kernel for preloading block device modules. The `mkhomedir_helper` command is a helper utility that creates home directories. The `mkfs.msdos` and `mkfs.vfat` files are symbolic links to `mkdosfs`.

## Using `mkfs`

The default file system type created when using the `mkfs` command is ext2. As previously mentioned, `mkfs` is really a wrapper that calls other file system build utilities. Therefore, any of the following commands create an ext2 file system on the specified device:

```
# mkfs /dev/xvdd1
# mke2fs /dev/xvdd1
# mkfs.ext2 /dev/xvdd1
```

To create an ext3 file system, use any of the following commands:

```
# mkfs -t ext3 /dev/xvdd1
# mke2fs -t ext3 /dev/xvdd1
# mkfs.ext3 /dev/xvdd1
```

To create an ext4 file system, use any of the following commands:

```
# mkfs -t ext4 /dev/xvdd1
# mke2fs -t ext4 /dev/xvdd1
# mkfs.ext4 /dev/xvdd1
```

## Configuration File

A number of options are available to customize block size, fragment size, blocks per group, journal options, number of inodes, and other parameters. Without including any options, the defaults that are specified in the `/etc/mke2fs.conf` configuration file are used.

## File System Labels

A useful option for the file system build utilities is the `-L name` option. This assigns a label to the partition; this label can be used instead of the device name when mounting the file system. Labels are limited to a maximum size of 16 characters. For existing file systems, the `e2label` command is used to display or set a label.

File systems are automatically assigned a universally unique identifier (UUID). UUIDs can be used when mounting the file system. To display the UUID, the label, and the file system type, use the `blkid` command. The following examples illustrate creating different file systems, with and without a label, and displaying the information with the `blkid` command (only a partial UUID is displayed). To create an `ext2` file system and display information, type:

```
# mkfs /dev/xvdd1
# blkid /dev/xvdd1
/dev/xvdd1: UUID=:f6f32f..." TYPE="ext2"
```

To create an `ext3` file system and display information, enter:

```
# mkfs -t ext3 /dev/xvdd1
# blkid /dev/xvdd1
/dev/xvdd1: UUID="f6f32f..." SEC_TYPE="ext2" TYPE="ext3"
```

To create an `ext4` file system, and assign a label name, and display information, enter:

```
# mkfs -t ext4 -L ProjectA /dev/xvdd1
# blkid /dev/xvdd1
/dev/xvdd1: UUID="f6f32f..." TYPE="ext4" LABEL="ProjectA"
```

# Mounting File Systems

- Use the `mount` command to attach a device to the directory hierarchy:
  - `mount [option] device mount_point`
- Use the device name, the UUID, or the label:
  - `mount /dev/xvdd1 /test`
  - `mount UUID="uuid_number" /test`
  - `mount LABEL="label_name" /test`
- Use the `-o` flag to specify mount options:
  - `mount -o nouser,ro /dev/xvdd1 /test`
- Use the `umount` command to unmount file systems:
  - `umount /dev/xvdd1`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

File systems on different partitions and removable devices, such as CDs, DVDs, or USB flash drives, must be attached to the directory hierarchy to be accessed. To attach a partition or device, a mount point must be created. A mount point is simply a directory created with the `mkdir` command. After a directory, or mount point, is created, attach the partition by using the `mount` command. Syntax for the `mount` command is:

```
mount [options] device_file mount_point
```

The following example creates a mount point (`/test`) and attaches the partition:

```
# mkdir /test
# mount /dev/xvdd1 /test
```

Alternatively, mount the partition or device by referencing the UUID or label. The following example displays the UUID and label, using the `blkid` command, and mounts the partition by referencing each (partial UUID is used):

```
# blkid /dev/xvdd1
/dev/xvdd1: UUID="9d7ab..." TYPE="ext4" LABEL="ProjectA"
# mount UUID="9d7ab..." /test
# mount LABEL="ProjectA" /test
```

The `mount` command without any options displays all currently attached file systems:

```
# mount
/dev/xvdd1 on /test type ext4 (rw)
...
```

In this example, the `/dev/xvdd1` partition is mounted on `/test`. The file system type is `ext4` and is mounted for both reading and writing.

The `df` command also displays mounted file systems. Example:

```
# df -h
Filesystem Size Used Avail Use% Mounted on
/dev/xvdd1 5.0G 139M 4.6G 3% /test
...
```

The information in the `proc` file system displays mounted file systems. Example:

```
# cat /proc/mounts
...
```

## Mount Options

To specify mount options, use the `-o` flag followed by a comma-separated string of options. The following are some of the available options for the `mount` command:

- **auto**: Allows the file system to be mounted automatically by using the `mount -a` command
- **loop**: Mounts the image as a loop device
- **noauto**: Disallows the automatic mount of the file system by using the `mount -a` command
- **noexec**: Disallows the execution of binary files on the file system
- **nouser**: Disallows an ordinary user (other than `root`) to mount and unmount the file system
- **remount**: Remounts the file system in case it is already mounted
- **ro**: Mounts the file system for reading only
- **rw**: Mounts the file system for both reading and writing
- **user**: Allows an ordinary user (other than `root`) to mount and unmount the file system

For example, to mount the `/dev/xvdd1` partition on the `/test` mount point as read-only with only the `root` user able to mount and unmount the file system, enter:

```
# mount -o nouser,ro /dev/xvdd1 /test
```

To mount an ISO image by using the `loop` device (assuming that the ISO image is present in the current directory and the mount point exist), enter:

```
# mount -o ro,loop OracleLinux-R6-U1-Server-x86_64-dvd.iso
/media/cdrom
```

## Journaling Mount Options

The `ext3` and `ext4` file systems have three journaling levels that may be set with the `-o` option in the `mount` command or in the options section of `/etc/fstab`:

- **data=journal**: The highest level. The one that does the most journaling. This writes the journal entries for all the data and metadata changes. All data is committed into the journal before being written into the main file system.

- **data=ordered**: The default mode. All data is forced directly out to the main file system before its metadata is committed to the journal.
- **data=writeback**: The lowest level. Data ordering is not preserved. Data may be written into the main file system after its metadata has been committed to the journal.

## Unmounting File Systems

To unmount a file system, use the `umount` command. The partition name, the device name, or the mount point is used as an argument. Example:

```
# umount /dev/xvdd1  
# umount /test
```

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

## /etc/fstab File

- The /etc/fstab file is the file system mount table, and it:
  - Contains all the information needed by the `mount` command
  - Is read at boot time
- When creating a new file system, add a new entry to the file system mount table in the following format:
  - `/dev/xvda3 /boot ext4 defaults 0 0`
  - The first column is the device to mount.
  - The second column is the mount point.
  - The third column is the file system type.
  - The fourth column specifies mount options.
  - The fifth column is used by the `dump` command.
  - The last column is used by the `fsck` command.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The /etc/fstab file is called the “file system mount table” and contains all the information that the `mount` command needs to mount devices. When adding a new file system, create the appropriate entry in /etc/fstab to ensure that the file system is mounted at boot time. The following is an example of entries in the /etc/fstab file:

```
# cat /etc/fstab
/dev/xvda1  /boot  ext4  defaults  1  2
/dev/xvda2   /      ext4  defaults  1  1
/dev/xvda3   swap   swap  defaults  0  0
```

The first column is the device to mount. The UUID or the label name should be used in place of the device name, because device names could change. The second column is the mount point, except the swap partition entry. Swap is discussed later. The third column is the file system type. The fourth column specifies mount options. The fifth column is used by the `dump` command. The number 1 means to dump the file system and 0 means the file system does not need to be dumped. The last column is used by the `fsck` program to determine the order in which file system checks are done at reboot time. The root file system should be specified with a value of 1 and the other file systems should have a value of 2. A value of 0 does not check the file system.

# Maintaining File Systems

- The `fsck` command checks and repairs Linux file systems.
  - `fsck` runs at boot time based on configurable parameters.
  - Do not run `fsck` on mounted file systems.
- Use the `tune2fs` command to:
  - Configure the frequency of file system checks
  - Convert ext2 file systems to ext3
  - Adjust file system parameters on ext2, ext3, and ext4 file systems
  - Display current file system parameter values
- Use the `dumpe2fs` utility to print file system information.
- The `debugfs` utility is an interactive file system debugger.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The best tool for maintaining file systems is `fsck`, which checks and repairs Linux file systems. By default, `fsck` runs after 20 system reboots but should be run manually if your system runs for weeks or months with rebooting. The frequency of file system checks is changed by using the `tune2fs` command. Other utilities for performing file system maintenance include `dumpe2fs` and `debugfs`. The `dumpe2fs` utility prints the super block and blocks group information for the file system on the specified device. The `debugfs` utility is an interactive file system debugger.

## Using `fsck`

The `fsck` command accepts a device name, a mount point, a UUID, or a file system label as an argument. If no argument is given, `fsck` checks all file systems listed in `/etc/fstab`. Do not run `fsck` on mounted file systems, because it causes severe file system damage. To unmount the file system and run the `fsck` utility on `/dev/xvdd1`:

```
# umount /dev/xvdd1
# fsck /dev/xvdd1
...
fsck.ext2: Superblock invalid, trying backup blocks...
/dev/xvdd1 was not cleanly unmounted, check forced.
Resize inode not valid. Recreate<y>?
```

Notice that the `fsck` utility calls the `e2fsck` utility to check the file system. File system-specific commands are located in `/sbin`:

```
# ls/sbin/*fsck*
/sbin/dosfsck  /sbin/fsck.cramfs  /sbin/fsck.ext4  /sbin/fsck.vfat
/sbin/e2fsck  /sbin/fsck.ext2  /sbin/fsck.ext4dev  /sbin/fsck
/sbin/fsck.ext3  /sbin/fsck.msdos
```

If the file system is corrupted, you are prompted to respond to a series of questions during repair attempts. You can include the `-y` option to use “yes” as an answer to all questions. Additional options to `fsck` are given:

- **-s**: Serialize `fsck` operations. This is a good idea if you are checking multiple file systems and the checkers are in an interactive mode.
- **-A**: Walk through the `/etc/fstab` file and try to check all file systems in one run. This option is typically used from the `/etc/rc` system initialization file. The `root` file system is checked first. After that, file systems are checked in the order specified by the sixth field in the `/etc/fstab` file. File systems with a value of `0` in this field are skipped and are not checked.
- **-R**: When checking all file systems with the `-A` flag, skip the `root` file system (in case it is already mounted read-write).

## Using `tune2fs`

The `tune2fs` utility is mainly used to set file system check options, and to convert an ext2 file system to ext3. You should always use the `e2fsck` utility before and after using `tune2fs`. To convert an ext2 file system to ext3, enter:

```
# tune2fs -j block_device
```

The `block_device` argument contains the ext2 file system that you want to convert. The `-j` option adds an ext3 journal to the file system.

The most commonly used options to `tune2fs` are:

- **-c max-mount-counts**: Adjust the maximum mount count between two file system checks.
- **-C mount-count**: Set the number of times the file system has been mounted.
- **-i interval-between-checks [d|m|w]**: Adjust the maximum time between two file system checks.
- **-m reserved-blocks-percentage**: Set the percentage of reserved file system blocks.
- **-r reserved-blocks-count**: Set the number of reserved file system blocks.

Use the `tune2fs` command to adjust various tunable file system parameters on ext2, ext3, and ext4 file systems. Current values are displayed by using the `-l` option. Example:

```
# tune2fs -l /dev/xvda1
```

Alternatively, use the `dumpe2fs` command to display file system parameters:

```
# dumpe2fs /dev/xvda1
```

# Swap Space

- Swap space is used when there is insufficient RAM.
- Swap space is a partition, a file, or both.
- Use fdisk, cfdisk, or parted to create a swap partition.
- Use dd to create a swap file:
  - # dd if=/dev/zero of=/swapfile bs=1024 count=1000000
- Use mkswap to initialize a swap partition or file:
  - mkswap {device|file}
- Use swapon and swapoff to enable and disable devices and files for swapping:
  - swapon {device|file}
  - swapoff {device|file}



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Swap space is used in Linux when there is insufficient physical memory (RAM) on your system to store the data currently being processed. When your system needs more memory, inactive memory pages are written to the disk, freeing up physical memory. Increasing swap space should not be considered as a solution to memory shortages. Swap space is located on disk drives, which have slower access times than physical memory. If your system is swapping often, you should add more physical memory, not more swap space.

Swap space in Linux is either a normal file in the file system, called a swap file, or a separate partition, or a combination of swap partitions and swap files. A dedicated swap partition is much faster, but it is easier to change the size of a swap file. If you know how much swap space you need, use a swap partition. If you are unsure, experiment with a swap file first, then make a swap partition when you know your requirements.

The swap partition is listed in the partition table, referenced in /etc/fstab, and viewable in the /proc/swaps file. There are also command-line utilities to display information about your swap space. To view the swap partition in the partition table, enter:

```
# fdisk -l | grep swap
/dev/xvda3      778      1306      4238336      82      Linux swap /
Solaris
```

To view the swap partition (or file) in the /etc/fstab file, enter:

```
# grep swap /etc/fstab
/dev/xvda3 swap swap defaults 0 0
```

To display the contents of the /proc/swaps file, enter:

```
# cat /proc/swaps
Filename      Type      Size      Used      Priority
/dev/xvda3    partition 4238328 0         -1
```

## Swap Utilities

The mkswap command is used to initialize a swap partition or a swap file. The syntax is:

```
mkswap {device|file}
```

The swapon and swapoff utilities enable and disable devices and files for swapping. To display current swap information, use the swapon -s command. Output is identical to viewing the contents of /proc/swaps.

## Adding Swap Space

The swap partition or swap file must exist before it is initialized. Use fdisk or parted to create a swap partition. A swap file is created by using the dd command. Example:

```
# dd if=/dev/zero of=/swapfile bs=1024 count=1000000
```

To initialize a swap partition, type:

```
# mkswap /dev/xvdd1
```

To initialize a swap file, type:

```
# mkswap /swapfile
```

Initialized swap space is enabled by using the swapon command. To enable swapping on a swap file, enter:

```
# swapon /swapfile
```

To enable swapping on a swap partition, enter:

```
# swapon /dev/xvda3
```

Update the /etc/fstab file to enable the swap partition or swap file at boot:

```
/dev/xvda3 swap swap defaults 0 0
/swapfile swap swap defaults 0 0
```

## Viewing Swap Usage

View the /proc/meminfo file, or use other utilities such as free, top, and vmstat to view memory and swap space usage. Example:

```
# grep -i swap /proc/meminfo
SwapCached:          0 kB
SwapTotal:        5238328 kB
SwapFree:         5238328 kB
```

To view swap usage by using the free command, enter:

```
# free |grep -i swap
Swap: 5238328 0 5238328
```

# Quiz

Which of the following is the file system mount table for Oracle Linux?

- a. /etc/vfstab
- b. /etc/filesystem
- c. /etc/fstab
- d. /boot/filesystem

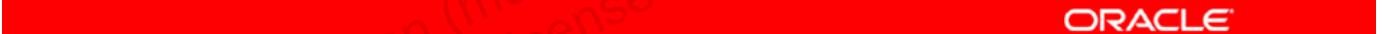


Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Quiz

Which of the following statements is true?

- a. Use the `umount` command to unmount file systems.
- b. Use the `swapoff` command to disable swap space.
- c. File systems must be unmounted and swap partitions must be disabled before repartitioning a disk drive.
- d. Do not run `fsck` on mounted file systems.
- e. All of the above.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Summary

In this lesson, you should have learned how to:

- Describe disk partitioning
- Use disk partitioning utilities
- Describe supported file system types
- Describe file system creation, mounting, and maintenance
- Describe and maintain swap space

## Practice 11: Overview

The practices for this lesson cover the following:

- Listing current disk partitions
- Partitioning a storage device
- Creating ext3 and ext4 file systems
- Increasing swap space

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# 12

## Storage Administration

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe the Linux device mapper
- Describe Logical Volume Manager (LVM)
- Configure LVM components
- Back up and restore volume group metadata
- Describe Linux kernel multi-disk (MD) driver
- Describe RAID and configure RAID devices

## Logical Volume Manager (LVM)

- LVM is a tool to facilitate the management of physical volumes, volume groups, and logical volumes.
  - Physical volume (PV): A physical storage device
  - Volume group (VG): Physical volumes are grouped together into storage pools called volume groups.
  - Logical volume (LV): Each volume group is divided into multiple logical volumes.
- File systems are created on logical volumes.
- Use LVM to increase the size of VGs and LVs “on the fly” (without interrupting operations).



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Linux device mapper (DM) provides an abstraction layer on top of the actual storage block devices and provides the foundation for Logical Volume Manager (LVM2), RAID, encryption, and other storage features. LVM2 manages multiple physical volumes and also supports mirroring and striping of logical volumes to provide redundancy and increase performance. To assist in understanding LVM, the following terms are defined:

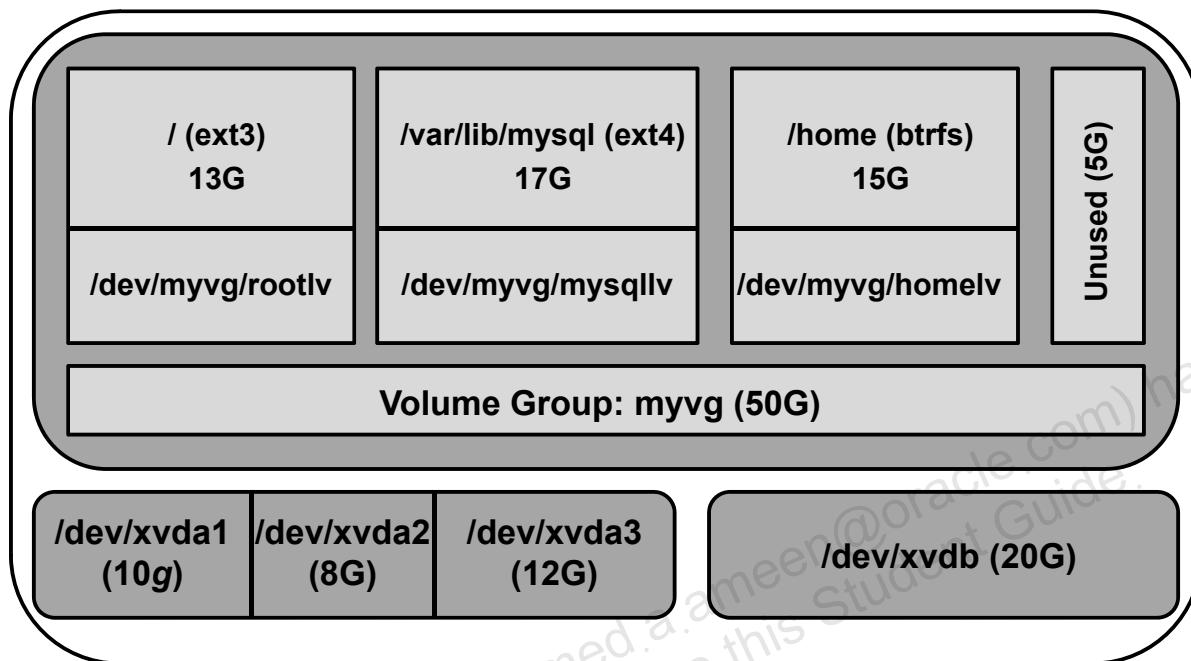
- **Physical volumes:** These are physical storage devices (hard drives, partitions, arrays).
- **Volume groups:** Physical volumes are grouped together into volume groups.
- **Logical volumes:** Each volume group is divided into multiple logical volumes.

Each logical volume is analogous to a standard disk partition. Logical volumes, therefore, function as partitions that can span multiple physical disks.

File systems, such as ext3 or ext4, can be created on logical volumes and connected to the directory hierarchy through mount points. As these “partitions” become filled with data, use LVM to increase their capacity from free space in the volume group. New physical storage devices are added to volume groups to increase the capacity of these groups.

With LVM, capacity is expanded in logical volumes “on the fly” (dynamically) without the need to back up the data on standard partitions, modify the partition table, and restore the data. Logical volume management does not interrupt usage and is transparent to users.

## LVM Configuration: Example



**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This slide illustrates a possible LVM configuration. There are four physical volumes (PV), with three of these being partitions on one drive, and the fourth being an entire hard drive:

- xvda1: 10 GB
- xvda2: 8 GB
- xvda3: 12 GB
- xvdb: 20 GB

All of the physical volumes are grouped into a single volume group (VG), named myvg. The storage capacity of this group is 50 GB, which is the total space of the four physical volumes.

The volume group is divided into three logical volumes (LV). The following lists the LV name, the size, the mount point, and the file system type of each logical volume:

- rootlv, 13 GB, / (root), ext3
- mysql1lv, 17 GB, /var/lib/mysql, ext4
- homelv, 15 GB, /home, btrfs

Finally, the illustration shows that there is 5 GB of unused space in the volume group. This is available to be allocated to any of the existing logical volumes, or to a new logical volume.

# Physical Volume Utilities

- Use the pvcreate command to create physical volumes:

```
# pvcreate -v /dev/xvdd1 /dev/xvdd2
```

- The following commands display physical volumes:

- pvdisplay
  - pvs
  - pvscan

- Use the pvremove command to remove physical volumes:

```
# pvremove /dev/xvdd1
```

- Additional PV commands are available.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The first step in implementing LVM is to create physical volumes. In addition to creating physical volumes, commands exist to display the attributes of physical volumes, remove physical volumes, and perform other functions on physical volumes.

## Creating Physical Volumes

Use the pvcreate command to create physical volumes. The syntax is:

```
pvcreate [options] device
```

You can initialize multiple disks or partitions for use by LVM in the same command. For example, the following command initializes two partitions. The `-v` option makes the output more verbose:

```
# pvcreate -v /dev/xvdd1 /dev/xvdd2
Set up physical volume for "/dev/xvdd1" with 6313482 available
sectors
Zeroing start of device /dev/xvdd1
Physical volume "/dev/xvdd1" successfully created
output omitted...
```

## Displaying Physical Volumes

Use the `pvdisplay` command to display attributes of physical volumes.

```
# pvdisplay
"/dev/xvdd1" is a new physical volume of "3.01 GiB"
--- NEW Physical volume ---
PV Name          /dev/xvdd1
output omitted...
```

In addition to `pvdisplay`, two other commands list information about physical volumes. The `pvs` command reports information about physical volumes in a more condensed form. The `pvscan` command scans all disks for physical volumes. Example:

```
# pvs
PV          VG   Fmt  Attr Psize  PFree
/dev/xvdd1    lvm2  a-   3.01g  3.01g
/dev/xvdd2    lvm2  a-   2.01g  2.01g
# pvscan
PV /dev/xvdd1      lvm2  [3.01GiB]
PV /dev/xvdd2      lvm2  [2.01GiB]
Total: 2 [5.02G9B] / in use: 0 [0     ] / in no VG: 2 [5.02 GiB]
```

## Removing Physical Volumes

Use the `pvremove` command to remove a physical volume, for example:

```
# pvremove /dev/xvdd1
Labels on physical volume "/dev/xvdd1" successfully wiped
# pvdisplay /dev/xvdd1
No physical volume label read from /dev/xvdd1
Failed to read physical volume "/dev/xvdd1"
```

## Additional PV Commands

The following are other commands that are associated with the manipulation of physical volumes:

- **`pvchange`**: Change the attributes of physical volumes.
- **`pvresize`**: Resize physical volumes.
- **`pvck`**: Check the consistency of physical volumes.
- **`pvmove`**: Move extents from one physical volume to another.

## Volume Group Utilities

- Use the `vgcreate` command to create volume groups:

```
# vgcreate -v myvolg /dev/xvdd1 /dev/xvdd2
```

- The following commands display volume groups:

- `vgdisplay`
- `vgs`
- `vgscan`

- Use the `vgremove` command to remove volume groups:

```
# vgremove myvolg
```

- Additional VG commands are available, for example:
  - `vgextend`: Add physical volumes to a volume group.
  - `vgreduce`: Remove physical volumes from a volume group.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The next step in implementing LVM is to assign the physical volumes to an existing or new volume group.

### Creating a Volume Group

Use the `vgcreate` command to create a new volume group. Space in a volume group is divided into “extents.” The default physical extent size is 4 MB. The syntax is:

```
vgcreate [options] volume_group_name physical_volume
```

For example, to create a volume group named `myvolg` by using the `/dev/xvdd1` and `/dev/xvdd2` physical volumes with a default physical extent size of 4 MB, enter:

```
# vgcreate -v myvolg /dev/xvdd1 /dev/xvdd2
Wiping cache of LVM-capable devices
Adding physical volume '/dev/xvdd1' to volume group 'myvolg'
Adding physical volume '/dev/xvdd2' to volume group 'myvolg'
Archiving volume group "myvolg" metadata (seqno 0).
Creating volume group backup "/etc/lvm/backup/myvolg" (seqno 1).
Volume group "myvolg" successfully created
```

## Displaying Volume Groups

Use the `vgdisplay` command to display attributes of volume groups:

```
# vgdisplay
--- Volume group ---
VG Name          myvolg
System ID
Format           lvm2
output omitted...
```

In addition to `vgdisplay`, two other commands list information about volume groups. The `vgs` command reports information about volume groups in a more condensed form. The `vgscan` command scans all disks for volume groups and rebuilds caches. Example:

```
# vgs
VG      #PV  #LV  #SN  Attr     Vsize   VFree
myvolg    2    0    0  wz--n-  5.01g  5.01g
# vgscan
Reading all physical volumes. This may take a while...
Found volume group "myvolg" using metadata type lvm2
```

## Removing Volume Groups

Use the `vgremove` command to remove a volume group, for example:

```
# vgremove myvolg
Volume group "myvolg" successfully removed
# vgdisplay
No volume groups found
```

## Additional VG Commands

The following commands are used to manipulate volume groups:

- **`vgcfgbackup`**: Back up volume group configurations.
- **`vgcfgrestore`**: Restore volume group configurations.
- **`vgchange`**: Change volume group attributes.
- **`vgck`**: Check the consistency of volume groups.
- **`vgconvert`**: Change the volume group metadata format.
- **`vgexport`**: Unregister volume groups from the system.
- **`vgextend`**: Add physical volumes to a volume group.
- **`vgimport`**: Register an exported volume group with the system.
- **`vgmerge`**: Merge volume groups.
- **`vgmknodes`**: Create special files for volume group devices in `/dev`.
- **`vgreduce`**: Remove physical volumes from a volume group.
- **`vgrename`**: Rename a volume group.
- **`vgsplit`**: Move physical volumes into a new or existing volume group.

The use of the `vgcfgbackup` and `vgcfgrestore` commands are discussed in a later slide.

# Logical Volume Utilities

- Use the `lvcreate` command to create logical volumes:

```
# lvcreate -v --size 2g --name myvol myvolg
```

- The following commands display logical volumes:
  - `lvdisplay`
  - `lvs`
  - `lvscan`

- Use the `lvremove` command to remove logical volumes:

```
# lvremove myvolg/myvol
```

- Additional LV commands are available, for example:
  - `lvextend`: Add space to a logical volume.
  - `lvreduce`: Reduce the size of a logical volume.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The next step in implementing LVM is to create logical volumes from the space allocated to the volume groups.

## Creating Logical Volumes

Use the `lvcreate` command to create a new logical volume. This command automatically creates the block device nodes in the `/dev` directory. The syntax is:

```
lvcreate [options] --size <size> --name LV_name VG_name
```

The `--size` option defines the size of the logical volume by allocating logical extents from the free physical extent pool of the volume group. For example, to create a logical volume named `myvol` from the volume group named `myvolg` with a size of 2 GB, enter:

```
# lvcreate -v --size 2g --name myvol myvolg
Setting logging type to disk
Finding volume group "myvolg"
Archiving volume group "myvolg" metadata (seqno 1).
Creating logical volume myvol
Create volume group backup "/etc/lvm/backup/myvolg" (seqno 2).
output omitted...
```

## Displaying Logical Volumes

Use the `lvdisplay` command to display the attributes of logical volumes.

```
# lvdisplay
--- Logical volume ---
LV Name          /dev/myvolg/myvol
VG Name          myvolg
LV UUID          urhkBs-Gahd...
LV Write Access  read/write
output omitted...
```

In addition to `lvdisplay`, two other commands list information about logical volumes. The `lvs` command reports information about logical volumes in a more condensed form. The `lvscan` command scans all disks for logical volumes. Example:

```
# lvs
LV      VG      Attr     LSize   Origin  Snap%  Move  Log  Copy%  Convert
myvol  myvolg  -wi-a-  2.00g
# lvscan
ACTIVE    '/dev/myvolg/myvol' [2.00 GiB] inherit
```

## Removing Logical Volumes

Use the `lvremove` command to remove a logical volume. You must include the volume group name as well as the logical volume name. You will be prompted to confirm your request.

Example:

```
# lvremove myvol
Volume group "myvol" not found
Skipping volume group myvol
# lvremove myvolg/myvol
Do you really want to remove active logical volume myvol? [y/n]: y
Logical volume "myvol" successfully removed
```

## Additional LV Commands

The following commands are used to manipulate logical volumes:

- **`lvchange`**: Change the attributes of logical volumes.
- **`lvconvert`**: Change logical volume layout.
- **`lvextend`**: Add space to a logical volume.
- **`lvmdiskscan`**: List devices that may be used as physical volumes.
- **`lvmsadc`**: Collect activity data.
- **`lvmsar`**: Create activity report.
- **`lvreduce`**: Reduce the size of a logical volume.
- **`lvrename`**: Rename a logical volume.
- **`lvresize`**: Resize a logical volume.

## Making Logical Volumes Usable

- Final steps:
  - Create a file system on the logical volume.
  - Create a mount point.
  - Attach the logical volume to the directory hierarchy.
- The `lvcreate` command creates two entries in the `/dev` directory for each logical volume. Example:
  - `/dev/mapper/myvolg-myvol`
  - `/dev/myvolg/myvol`
- Either of these block device names are usable as arguments to the `mkfs` command:

```
# mkfs -t ext4 /dev/mapper/myvolg-myvol
# mkfs -t ext4 /dev/myvolg/myvol
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The last step in implementing LVM is to create a file system on the logical volume, create a mount point, and attach the logical volume to the directory hierarchy. There is nothing new here, these steps were discussed in the lesson titled “Oracle Linux File Systems.” Logical volumes do not require a file system to be usable. For example, they can be used as ASM (Automatic Storage Management) disks or as a raw device.

The only thing different from creating a file system on a disk partition, and creating a file system on a logical volume, is the name of the block device in the `/dev` directory. The `lvcreate` command creates two entries in the `/dev` directory for each logical volume. For example, when creating the logical volume named `myvol` from the volume group named `myvolg`, the following two block device names in the `/dev` directory were automatically created:

```
/dev/mapper/myvolg-myvol
/dev/myvolg/myvol
```

Use either of these device names as arguments to the `mkfs` command when making a file system. For example, to make an `ext4` file system on the `myvol` logical volume, enter either of the following commands:

```
# mkfs -t ext4 /dev/mapper/myvolg-myvol
# mkfs -t ext4 /dev/myvolg/myvol
```

The `blkid` command displays the same output (and the same UUIDs) when querying either of the logical volume device names:

```
# blkid /dev/mapper/myvolg-myvol  
/dev/mapper/myvolg-myvol: UUID="9fa64e..." TYPE="ext4"  
# blkid /dev/myvolg/myvol  
/dev/myvolg/myvol: UUID="9fa64e..." TYPE="ext4"
```

Create a mount point and mount the new logical volume file system, for example:

```
# mkdir /test  
# mount /dev/mapper/myvolg-myvol /test
```

Create an entry in `/etc/fstab` to mount the file system at boot time.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

## Backing Up and Restoring Volume Group Metadata

- LVM metadata contains configuration details of volume groups.
- Metadata backups and archives are automatically created on every volume group and logical volume configuration change.
  - Backups are stored in /etc/lvm/backup.
  - Archives are stored in /etc/lvm/archive.
- Configuration settings are stored in /etc/lvm/lvm.conf.
- Use the vgcfgbackup command to manually back up LVM metadata.
- Use the vgcfgrestore command to restore from a backup to recover from corrupted or missing metadata.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

LVM metadata contains configuration details of LVM volume groups. By default, metadata backups and archives are automatically created on every volume group and logical volume configuration change. Settings can be changed in the LVM configuration file, /etc/lvm/lvm.conf. The lvm dumpconfig command displays configuration settings.

Metadata backups are stored in the /etc/lvm/backup directory. Metadata archives are stored in the /etc/lvm/archive directory. You can manually back up the metadata by using the vgcfgbackup command. For example, the following command backs up the metadata of the myvolg volume group to the /etc/lvm/backup/myvolg file:

```
# vgcfgback myvolg
```

Omit the name of the volume group to back up metadata for all volume groups. Use the -f <filename> option to give the backup file a specific file name.

The following are examples of error messages you might get if the metadata area is corrupted or incorrect:

```
Couldn't find device with uuid '...'.
```

```
Couldn't find all physical volumes for volume group myvolg.
```

You can use the vgcfgrestore command to restore volume group metadata from a backup. Provide the name of the volume group as an argument to the vgcfgrestore command.

# Redundant Array of Independent Disks (RAID)

- The multi-disk (MD) driver supports software RAID.
  - MD organizes disk drives into RAID devices (arrays).
- Common RAID levels are supported:
  - Linear RAID: Concatenated drives
  - RAID-0: Striping
  - RAID-1: Mirroring
  - RAID-5: Distributed parity
  - RAID-6: Dual-distributed parity
  - Nested RAID levels:
    - RAID 0+1: Mirrored striping
    - RAID 1+0 (or RAID 10): Striped mirror



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

In addition to logical volume management with LVM2, the Linux kernel supports “software RAID” with the multi-disk (MD) driver. MD organizes disk drives into RAID devices, or arrays, and provides different RAID levels. RAID devices are virtual devices created from two or more real block devices.

RAID combines multiple disk drives into an array and allows data to be spread across the drives to increase capacity, achieve redundancy, and increase performance.

## Supported RAID Levels

The following RAID levels are the most commonly used levels supported by Oracle Linux:

- **Linear RAID:** Linear RAID simply groups drives together to create a larger virtual drive. Data is written to the first drive until it is full, and then it is written to the next drive. There is no redundancy or performance benefit. Reliability is actually decreased, because the entire array cannot be used if any one drive fails.
- **RAID-0:** RAID-0 is called striping and provides an increase in performance but offers no redundancy. Data is broken down into stripes and written across all the drives, rather than filling up the first drive before moving on to the next as is the case with Linear RAID. In addition, as is the case with Linear RAID, the entire array cannot be used if any one drive fails.

- **RAID-1:** RAID-1 is called mirroring and provides redundancy by writing identical data to each drive in the array. If one drive fails, the mirror drive satisfies I/O requests. RAID-1 is expensive because the same information is written to all of the disks in the array.
- **RAID-5:** RAID-5 is the most common type of RAID and uses striping with distributed parity. RAID-5 is able to recover from the loss of one drive in the array. Parity information is calculated based on the contents of the rest of the drives in the array. This information is used to reconstruct data when one drive in the array fails. The reconstructed data also satisfies I/O requests to the failed drive before it is replaced, and repopulates the failed disk after it has been replaced. With RAID-5, the parity is distributed across all drives in the array.
- **RAID-6:** RAID-6 uses striping with double distributed parity. RAID-6 is able to recover from the loss of two drives in the array. RAID-6 is commonly used when data redundancy and preservation, and not performance, is of most importance.

### Nested RAID Levels

Nested RAID levels, also known as hybrid RAID, combine standard RAID levels for additional performance and/or redundancy. One example is **RAID 0+1**, which is a mirror (RAID-1) of striped (RAID-0) disks. Another example is **RAID 1+0**, sometimes called **RAID 10**, which is a stripe of mirrors.

Many Oracle Database customers use one of these nested RAID levels, but have the RAID implemented in the SAN (storage area network) arrays. RAID 5 and RAID 6 provide the redundancy needed, but add overhead to calculate parity. This can impact the performance of write-intensive databases.

## mdadm Utility

- Use the mdadm command to build, manage, and monitor Linux MD devices (software RAID devices).
- To create a device:

```
# mdadm --create /dev/md0 --level=1 --raid-devices=2  
/dev/xvdd1 /dev/xvdd2
```

- To query a device:

```
# mdadm --query /dev/md0  
# mdadm --detail /dev/md0  
# cat /proc/mdstat
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

### Creating a RAID Device

The mdadm command is used to build, manage, and monitor Linux MD devices (software RAID devices). The basic syntax to create a new RAID array is:

```
mdadm --create <md_device> --level=<RAID_level> --raid-  
devices=<#> devices
```

For example, to create a RAID-1 device (/dev/md0) consisting of two block devices (/dev/xvdd1 and /dev/xvdd2), enter:

```
# mdadm --create /dev/md0 --level=1 -raid-devices=2 /dev/xvdd1  
/dev/xvdd2
```

Information about the RAID device to be created will be displayed along with the following prompt:

Continue creating array?

Respond with “y” to create the array. View the /proc/mdstat file to check the status of your MD RAID devices:

```
# cat /proc/mdstat  
Personalities : [raid1]  
md0 : active raid1 xvdd2[1] xvdd1[0]
```

## Querying a RAID Device

You can also use the `mdadm` command to view information about the RAID device:

```
# mdadm --query /dev/md0
/dev/md0: 2.01GiB raid1 2 devices, 0 spares.
```

To see even more detail, enter:

```
# mdadm --detail /dev/md0
/dev/md0:
      Version : 1.2
      Creation Time : Fri Nov 25...
      Raid Level : raid1
      Array Size : 2103479 (2.01 GiB 2.15 GB)
      Used Dev Size : 2103479 (2.01 GiB 2.15 GB)
      Raid Devices : 2
      Total Devices : 2
      Persistence : Superblock is persistent
      Update Time : Fri Nov 25...
      State : clean
      Active Devices : 2
      Working Devices : 2
      Failed Devices : 0
      Spare Devices : 0
output omitted...
```

## Making RAID Devices Usable

1. Create a file system on the RAID device.
2. Create a mount point.
3. Attach the RAID device to the directory hierarchy.
4. Update the `mdadm` configuration file:
  - `/etc/mdadm.conf`
  - `ARRAY /dev/md0 devices=/dev/xvdd1,/dev/xvdd2`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The last step in implementing a RAID device is to create a file system on the device, create a mount point, and attach the device to the directory hierarchy. Again, this is nothing new. These steps are necessary for standard partitions, logical volumes, and RAID devices. You can create logical volumes on top of RAID block devices.

Assuming that the RAID device name is `/dev/md0` and that you want to create an ext4 file system on the device and mount it to `/raid`, enter:

```
# mkfs -t ext4 /dev/md0
# mkdir /raid
# mount /dev/md0 /raid
```

The last step is to update the `mdadm` configuration file, `/etc/mdadm.conf`. It is useful to store the RAID configuration information in this file. This will help `mdadm` to assemble existing arrays at system boot. You can either copy and adapt the sample configuration file from `/usr/share/doc/mdadm-3.2.1/mdadm.conf-example`, or create the file from scratch. The following entry would suffice for the RAID device created in the practices for this lesson:

```
ARRAY /dev/md0 devices=/dev/xvdd1,/dev/xvdd2
```

# Quiz

Which of the following commands is used to build, manage, and monitor software RAID devices?

- a. raidadm
- b. lvadm
- c. mdadm
- d. dmsetup

## Summary

In this lesson, you should have learned how to:

- Describe the Linux device mapper
- Describe Logical Volume Manager (LVM)
- Configure LVM components
- Back up and restore volume group metadata
- Describe Linux kernel multi-disk (MD) driver
- Describe RAID and configure RAID devices

## Practice 12: Overview

The practices for this lesson cover the following:

- Creating Linux LVM partitions
- Creating a logical volume
- Creating a file system and mounting a logical volume
- Backing up volume group metadata
- Creating a logical volume snapshot
- Increasing the capacity of a logical volume
- Restoring volume group metadata
- Creating a RAID device



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# 13

## Network Configuration

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

- Describe network interface configuration files
- Use command-line network interface utilities
- Describe ARP and the ARP cache
- Describe network interface bonding
- Describe VLAN configuration
- Use the route utility to manipulate a routing table
- Use the NetworkManager tool to configure network connections
- Use the system-config-network utility



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Network Interfaces

- Each physical network device has an associated network interface configuration file.
- Network interface configuration files are located in the `/etc/sysconfig/network-scripts` directory.
- Configuration file names are `ifcfg-interface` where `interface` is `eth0`, `eth1`, `ppp0`, `irlan0`, `plip0`, and so on.
- Configuration parameters include:
  - `DEVICE=eth0`
  - `BOOTPROTO=none`
  - `TYPE=Ethernet`
  - `HWADDR=00:16:3E:00:01:02`
  - `IPADDR=192.0.2.102`
  - `NETMASK=255.255.255.0`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Linux handles network communications through software configuration files and the physical networking devices in your system. Each physical network device has an associated configuration file, named `ifcfg-<interface>`, located in the `/etc/sysconfig/network-scripts` directory.

```
# ls /etc/sysconfig/network-scripts/ifcfg*
ifcfg-eth0  ifcfg-eth1  ifcfg-lo
```

In this example, there are two Ethernet interfaces, represented by `ifcfg-eth0` and `ifcfg-eth1`, and one loopback interface (`ifcfg-lo`). The system uses these files during the boot process to configure the network interfaces. The following is a sample:

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
IPV6INIT=no
PEERDNS=yes
```

```
TYPE=Ethernet
HWADDR=00:16:3E:00:01:02
IPADDR=192.0.2.102
NETMASK=255.255.255.0
BROADCAST=192.0.2.255
NM_CONTROLLED=yes
```

A description of each of these configuration parameters follows:

**DEVICE=name:** The name of the physical device (except for dynamically-allocated PPP devices where it is the logical name)

**BOOTPROTO=protocol:** Where *protocol* is one of the following:

- **none:** No boot-time protocol is used.
- **bootp:** Use BOOTP (bootstrap protocol).
- **dhcp:** Use DHCP (Dynamic Host Configuration Protocol).

**ONBOOT=answer:** Where *answer* is one of the following:

- **yes:** This device is activated at boot time.
- **no:** This device is not activated at boot time.

**USERCTL=answer:** Where *answer* is one of the following:

- **yes:** Non-root users can control this device.
- **no:** Non-root users cannot control this device.

**IPV6INIT=answer:** Where *answer* is one of the following:

- **yes:** Enable IPv6 on this interface. If **IPV6INIT=yes**, the following parameters would also be set in this file:
  - **IPV6ADDR=IPv6 address**
  - **IPV6\_DEFAULTGW=**The default route through the specified gateway
  - **no:** Disable IPv6 on this interface.

**PEERDNS=answer:** Where *answer* is one of the following:

- **no:** Do not modify /etc/resolv.conf.
- **yes:** Modify /etc/resolv.conf if the DNS directive is set. If using DHCP, then yes is the default.

**PEERNIS=no:** Prevent overwriting /etc/yp.conf.

**PEERNTP=no:** Prevent overwriting /etc/ntp.conf.

**TYPE=device\_type:** The type of network interface device

**HWADDR=MAC-address:** The hardware address of the Ethernet device

**IPADDR=address:** The IP address assigned to the interface

**NETMASK=mask:** The netmask value

**BROADCAST=address:** The broadcast address. This directive is not required, because the value is calculated automatically with the ipcalc command.

**NM\_CONTROLLED=answer:** Where *answer* is one of the following:

- **yes:** Specifies that the card is controlled by the NetworkManager
- **no:** Specifies that the card is not controlled by the NetworkManager

## Additional Network Configuration Files

- `/etc/hosts` associates host names with IP addresses.
  - Larger networks would use DNS to perform this resolution.
  - Specify the IP address of the loopback device.
- `/etc/resolv.conf`:
  - Provides access to DNS
  - Identifies DNS name server(s) and search domain
- `/etc/sysconfig/network` specifies routing and host information for all network interfaces.
- `/etc/nsswitch.conf` lists the order of host name searches.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

In addition to the individual network interface configuration files in the `/etc/sysconfig/network-scripts` directory, there are other, more global network configuration files. These files are:

- `/etc/hosts`
- `/etc/resolv.conf`
- `/etc/sysconfig/network`
- `/etc/nsswitch.conf`

### `/etc/hosts`

This file associates host names with IP addresses. It resolves, or looks up, an IP address when the host name is known. Larger networks would use DNS (Domain Name Service) to perform this resolution. Even if using DNS, include in this file a line specifying the IP address of the loopback device (127.0.0.1) as `localhost.localdomain`. A sample `/etc/hosts` file follows. The first column contains the IP address. The second column is the fully qualified host names. Additional columns contain host name aliases:

```
# cat /etc/hosts
127.0.0.1      localhost.localdomain      localhost
192.0.2.101    host01.example.com        host01
```

**/etc/resolv.conf**

The resolver configuration file provides access to DNS. This file usually has at least two lines, one line specifying the IP address of a DNS server (or name server) and the other specifying the search domain. The following example shows three name servers and the search domain:

```
# cat /etc/resolv.conf
search us.oracle.com
nameserver 152.68.154.3
nameserver 10.216.106.3
nameserver 193.32.3.252
```

**/etc/sysconfig/network**

This file specifies routing and host information for all network interfaces. The following is a sample:

```
# cat /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=host02.example.com
GATEWAY=192.0.2.1
```

A description of each of these configuration parameters follows:

**NETWORKING=answer:** Where *answer* is one of the following:

- **yes:** Networking is enabled.
- **no:** Networking is disabled.

**NETWORKING\_IPV6=answer:** Where *answer* is one of the following:

- **yes:** IPv6 is enabled.
- **no:** IPv6 is disabled.

**HOSTNAME=hostname:** The host name of the machine.

**GATEWAY=address:** The IP address of the network's gateway.

**/etc/nsswitch.conf**

This file is the system databases and name service switch configuration file. It provides sources for common configuration databases and name resolution mechanisms. Entries in this file identify the database name in the first field, then a colon, then a list of possible resolution mechanisms in the second field. The order in which the mechanisms are listed determines the order in which queries on the specified database are resolved.

The following example indicates that host name resolution is attempted first by querying local files, that is, /etc/hosts, then by querying the DNS server if the host name is not resolved:

```
# cat /etc/nsswitch.conf
...
hosts:      files dns
...
...
```

# Command-Line Network Interface Utilities

- **ifconfig** is used:
  - At boot time to configure kernel-resident network interfaces
  - To display the status of an interface
  - To configure (non-persistent) properties
- **ifup** and **ifdown** are:
  - Interface control scripts
  - Used to activate and deactivate network interfaces
- **ethtool**
  - **ethtool** is used to query and set low-level network interface properties.
  - Changes made by **ethtool** do not persist after a reboot.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## **ifconfig**

At boot time, the **ifconfig** utility configures the kernel-resident network interfaces. You can also use it to display the status of an interface, or for debugging or system tuning. If no arguments are given, **ifconfig** displays the status of the active interfaces. Example:

```
# ifconfig
eth0      Link encap:Ethernet  Hwaddr 00:16:3E:00:01:03
          inet addr: 192.0.2.103  Bcast: 192.0.2.255...
                      ...
          UP BROADCAST RUNNING MULTICAST...
                      ...

```

The output includes the current IP address as shown with `inet addr:`. The status of the interface, `UP BROADCAST...`, is also provided.

To display the status of all interfaces, even those that are down, use the **ifconfig -a** command. To display the status of a specific interface, include the name of the interface as an argument. Example:

```
# ifconfig eth0
```

You can also use the `ifconfig` utility to set properties and activate a network interface. The following example sets the IP address and activates the interface:

```
# ifconfig eth0 10.1.1.1 netmask 255.255.255.0 up
```

Setting network interface properties with the `ifconfig` utility is not persistent. To make settings permanent, set the properties in their respective `/etc/sysconfig/network-scripts/ifcfg-<interface>` file.

## Interface Control Scripts

Interface control scripts are used to activate and deactivate network interfaces. There are two primary interface control scripts in the `/etc/sysconfig/network-scripts` directory, `ifdown` and `ifup`.

The `ifup` and `ifdown` interface scripts are symbolic links to scripts in the `/sbin` directory. When either of these scripts are called, they require the interface to be specified as an argument. For example, to activate the `eth0` interface:

```
# ifup eth0
```

The following control scripts also exist in the `/etc/sysconfig/network-scripts` directory:

- **`ifup-aliases`**: Configures IP aliases from interface configuration files when more than one IP address is associated with an interface
- **`ifup-ippp` and `ifdown-ippp`**: Bring ISDN interfaces up and down, respectively
- **`ifup-ipv6` and `ifdown-ipv6`**: Bring IPv6 interfaces up and down, respectively
- **`ifup-plip`**: Brings a PLIP interface up
- **`ifup-plusb`**: Brings a USB interface for network connections up
- **`ifup-post` and `ifdown-post`**: Contains commands to be executed after an interface is brought up or down
- **`ifup-ppp` and `ifdown-ppp`**: Bring a PPP interface up or down
- **`ifup-routes`**: Adds static routes for a device as its interface is brought up

To manipulate all network scripts simultaneously, use the `service` command. The following example starts and stops all network interfaces:

```
# service network restart
```

## ethtool

`ethtool` allows you to query and set properties of the network device. This is useful for diagnosing possible mismatched settings that affect performance. The settings that `ethtool` controls are considered low-level or device settings. The changes that `ethtool` makes are not permanent and do not persist through a reboot. To make the changes permanent, change the `/etc/sysconfig/network-scripts/ifcfg-<interface>` file for the device.

`ethtool` can be used to configure options such as speed, full or half duplex, autonegotiate, and other properties. To display a list of available options, use the `-h` option:

```
# ethtool -h
```

The following example configures the `eth0` interface to 100 Mb/sec, half duplex without trying to autonegotiate:

```
# ethtool -s eth0 speed 100 autoneg off duplex half
```

# Address Resolution Protocol (ARP)

- ARP resolves an IP address to the MAC address.
- IP addresses and associated MAC addresses are cached in an ARP table.
- By default, entries are cached for 60 seconds.
- Use the `arp` command to display, add, or delete entries in the ARP cache.
  - For example, to display all entries:  
`# arp -n`
- Alternatively, use the `ip neigh` command to modify the ARP cache.
  - For example, to delete all entries:  
`# ip neigh flush all`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Address Resolution Protocol (ARP) resolves an IP address to the MAC address. The MAC address is a 48-bit physical hardware address, which is burned into the network interface card (NIC). Network applications use the IP address to communicate with another device but the MAC address is needed to ensure network packets are delivered. Use the `ifconfig` command to view the IP address and the MAC address assigned to a specific NIC. The following example displays the MAC address, `00:16:3E:00:01:03`, and the IP address, `192.0.2.103`, for the `eth0` network interface:

```
# ifconfig
eth0      Link encap:Ethernet  Hwaddr 00:16:3E:00:01:03
          inet addr: 192.0.2.103 Bcast: 192.0.2.255...
          ...
          ...
```

For performance reasons, ARP caches resolve IP addresses and associate MAC addresses in an ARP cache (or table). By default, entries are cached for 60 seconds. This value can be modified on a per-network interface basis. For example, the following file stores the timeout value for the `eth0` interface:

```
# cat /proc/sys/net/ipv4/neigh/eth0/gc_stale_time
```

Use the `arp` command to display the ARP cache, to delete an ARP entry, or to add an entry to the cache. The following example displays the ARP cache. Use the `-n` option to show the IP addresses instead of host names:

```
# arp -n
Address          Hwtype   Hwaddress           Flags Mask
Iface
192.0.2.1        ether     fe:ff:ff:ff:ff:ff  C      eth0
192.0.2.102      ether     00:16:3e:00:01:01  C      eth0
192.168.1.254    ether     00:16:3e:00:01:01  C      eth1
```

Use the `-d` `hostname` option to remove an entry for the specified host. Example:

```
# arp -d 192.0.2.102
```

Use the `arp -h` command or refer to the `man` pages for a complete list of options and arguments to the `arp` command.

```
# arp -h
```

Usage:

```
arp [-vn ] [<HW>] [-i <if>] [-a] [<hostname>]<-Display ARP
cache
```

...

The `man` page for `arp` says, “This program is obsolete. For replacement check `ip neighbor`.” The ARP table is also known as the neighbor table. The following commands create new neighbor records or update existing ones. The `neighbor` command can be shortened:

- `ip neigh add` – Add a new neighbor entry.
- `ip neigh change` – Change an existing entry.
- `ip neigh replace` – Add a new entry or change an existing entry.
- `ip neigh delete` – Delete a neighbor entry.
- `ip neigh show` – List neighbor entries.
- `ip neigh flush` – Flush neighbor tables.

For example, the following command clears all entries in the ARP cache with verbosity:

```
# ip -s -s neigh flush all
192.0.2.1 dev eth0 lladdr fe:ff:ff:ff:ff:ff ref 3 used ...
192.0.2.101 dev eth0 lladdr 00:16:3e:00:01:01 ref 4 used ...
*** Round1, deleting 2 entries ***
*** Flush is complete after 1 round ***
```

# Network Interface Bonding

- Network interface bonding:
  - Combines multiple network connections into a single logical interface
  - Is used to increase throughput and provide redundancy
- Example of creating a bonding interface file:

```
/etc/sysconfig/network-scripts/ifcfg-bond0
DEVICE=bond0
```
- Physical interface files need MASTER and SLAVE directives:

```
MASTER=bond0
SLAVE=yes
```
- Load the bonding kernel module.
- You can also use the ifenslave command-line utility.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Network interface bonding is called by many names: Port Trunking, Channel Bonding, Link Aggregation, NIC teaming, and others. It combines or aggregates multiple network connections into a single channel bonding interface. This allows two or more network interfaces to act as one, to increase throughput and to provide redundancy or failover.

The Linux kernel comes with the bonding driver for aggregating multiple network interfaces, such as `eth0` and `eth1`, into a single logical interface such as `bond0`. The behavior of the bonded interfaces depends upon the mode. Modes provide either hot standby or load-balancing services. Additionally, link integrity monitoring may be performed.

## Creating a Bonding Interface File

To create a bonding interface, create a file in the `/etc/sysconfig/network-scripts` directory called `ifcfg-bondN`, where  $N$  is number for the interface, such as 0. The contents of the file are similar to the Ethernet interface configuration file except the `DEVICE` directive is set to `bond0` rather than `eth0`. Example:

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond0
DEVICE=bond0
IPADDR=192.168.1.1
NETMASK=255.255.255.0
...
...
```

After creating the bonding interface configuration file, the network interfaces to be bound together must be configured by adding the `MASTER` and `SLAVE` directives to their configuration files. The following example configures `eth0` as part of the `bond0` interface:

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
...
MASTER=bond0
SLAVE=yes
...
```

Add the `MASTER` and `SLAVE` directives to the `ifcfg-eth1` file to include `eth1` as part of the `bond0` interface.

### **Loading the Kernel Module**

You must create the `/etc/modprobe.d/bonding.conf` file containing the following entry:

```
alias bond0 bonding
```

This file can be named anything you like, but the name must end with a `.conf` extension. The existence of this file ensures that the bonding kernel module is loaded when the bonding interface is brought up. You must have an entry in this file for all configured bonding interfaces.

### **Using ifenslave**

You can also use the `ifenslave` utility to create and control the bond devices from the command line. The `ifenslave` command is used to attach and detach slave network devices to a bonding device.

The following commands load the bonding kernel module, set up a bonding device, and enslave two Ethernet devices to it:

```
# modprobe bonding
# ifconfig bond0 192.168.0.1 netmask 255.255.255.0
# ifenslave bond0 eth0 eth1
```

### **Bonding Module Parameters**

Do not specify options for the bonding device in `/etc/modprobe.d/bonding.conf` or in the deprecated `/etc/modprobe.conf` file. Parameters for the bonding kernel module must be specified as a space-separated list in the `BONDING_OPTS` directive in the `ifcfg-bondN` interface file:

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond0
...
BONDING_OPTS="bonding parameters separated by spaces"
```

A complete list of bonding module parameters is available at  
<http://www.kernel.org/doc/Documentation/networking/bonding.txt>.

This same document is in `/usr/share/doc/iputils-20071127/README.bonding`.

## Virtual Local Area Networks

- A VLAN is a group of machines that can communicate as if they were attached to the same broadcast domain.
- With VLANs, network switches (not routers) create the broadcast domain.
- Switch ports are assigned to a VLAN ID, and all ports assigned to a single VLAN are in a single broadcast domain.
- To create the `ifcfg-eth0.5` file for VLAN ID 5 on `eth0`:
  - `DEVICE=eth0.5`
  - `VLAN=yes`
- Alternatively, use the `vconfig` command:
  - `# vconfig add eth0 5`
- View the `/proc/net/vlan` directory to get detailed information about VLAN interfaces.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

A virtual local area network (VLAN) is a group of machines that can communicate as if they were attached to the same broadcast domain. VLANs allow systems to be grouped together regardless of their physical location. With VLANs, network switches (not routers) create the broadcast domain. Switch ports are assigned to a VLAN ID, other than 1, which is the default VLAN, and all ports assigned to a single VLAN are in a single broadcast domain.

For example, because switches can exchange information, some ports on switch A can be in VLAN 10 and other ports on switch B can be in VLAN 10. Broadcasts between these devices are not seen on any other port in any other VLAN, other than 10.

To implement VLANs in Linux, create an interface configuration file just as you do for an Ethernet interface or a bonding interface. The interface configuration file name includes the VLAN ID in its name and also includes the `VLAN=yes` directive in the file. For example, if your VLAN ID is 5 and you want to include `eth0` in this VLAN, copy the `ifcfg-eth0` file to file name `ifcfg-eth0.5`:

```
# cd /etc/sysconfig/network-scripts  
# cp ifcfg-eth0 ifcfg-eth0.5
```

In `ifcfg-eth0.5`, change the `DEVICE` directive and add the `VLAN` directive as follows:

```
DEVICE=eth0.5  
VLAN=yes
```

Assign the correct IP address by using DHCP or static IP. Save the file and restart the network. Do not modify the `ifcfg-eth0` file. You now have a VLAN-device, or a virtual Ethernet device, which represents the virtual LANs on the physical LAN:

- `eth0`: The regular network interface
- `eth0.5`: The virtual interface that uses untagged frames

This process can be used to configure VLAN on a bonding interface. For example, to configure VLAN ID 10 on `bond0`, copy `ifcfg-bond0` to `ifcfg-bond0.10`. Edit this new `ifcfg-bond0.10` file and make the following changes:

```
DEVICE=bond0.10  
VLAN=yes
```

Assign the correct IP address by using DHCP or static IP. Save the file and restart the network:

```
# service network restart
```

### Using the vconfig Command

Another method of configuring VLAN devices is to use the `vconfig` command. This command allows you to create and remove VLAN devices on a VLAN-enabled kernel. For example, to add VLAN ID 5 for `eth0`:

```
# vconfig add eth0 5
```

This `vconfig add` command creates a VLAN-device on `eth0`, which becomes the `eth0.5` interface. You can use the `ifconfig` command to see device information:

```
# ifconfig eth0.5
```

The following example uses the `ifconfig` command to assign an IP address to the VLAN interface:

```
# ifconfig eth0.5 192.168.1.100 netmask 255.255.255.0 broadcast  
192.168.1.255 up
```

Use the following commands if you want to delete the VLAN interface:

```
# ifconfig eth0.5 down  
# vconfig rem eth0.5
```

You can view the `/proc/net/vlan` directory to get detailed information about VLAN interfaces. For example, to view information on the two VLAN devices examples presented in this lesson:

```
# cat /proc/net/vlan/eth0.5  
# cat /proc/net/vlan/bond0.10
```

## route Utility

- The `route` utility is used to display or manipulate the IP routing table.
- The default route, GATEWAY, is configured in the `/etc/sysconfig/network` file.
- To display the routing table:
  - `route -n`
  - `netstat -r`
- To add an entry to the routing table:
  - `route add default gw 192.0.2.2`
  - `route add -net 192.18.21.0 netmask 255.255.255.0 eth0`
- Configure permanent static routes in the `/etc/sysconfig/network-scripts/route-interface` file.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `route` utility displays or manipulates the IP routing table. Its primary use is to set up static routes to specific hosts or networks through a network interface that has been configured by using the `ifconfig` command.

To create a default route, include a GATEWAY entry in the `/etc/sysconfig/network` file.  
Example:

```
# cat /etc/sysconfig/network
...
GATEWAY=192.0.2.1
```

Network traffic destined for hosts on another network would be handled by the 192.0.2.1 gateway on the local area network.

### Displaying the Routing Table

Use the `route -n` command to display the routing table. Here is part of an example routing table:

Destination	Gateway	Genmask	Flags	Metric	Ref...
default	192.0.2.1	0.0.0.0	UG	0	0...
192.0.2.0	0.0.0.0	255.255.255.0	U	0	0...

In this example, the gateway IP address of 192.0.2.1 was obtained from the entry in the /etc/sysconfig/network file. You can also use the `netstat -r` command to display the routing table.

The routing table in this example shows that any packet destined for the local network 192.0.2 does not use a gateway. Packets destined for other addresses are routed through the gateway at 192.0.2.1.

## **Adding a Route**

Use the `route add` command to add a static route. The following example adds a default route, which is used if no other route matches. All network packages using this route are “gateawayed” through the 192.0.2.2 IP address:

```
# route add default gw 192.0.2.2
```

The following example adds a route to the network 192.18.21.x through the `eth0` interface:

```
# route add -net 192.18.21.0 netmask 255.255.255.0 eth0
```

## **Deleting a Route**

Use the `route del` command to delete an entry from the routing table, for example:

```
# route del default gw 192.0.2.2
```

```
# route del -net 192.18.21.0 netmask 255.255.255.0 eth0
```

## **Configuring Permanent Static Routes**

To make static routes permanent, configure them for each interface. Static route configuration is stored in a /etc/sysconfig/network-scripts/route-interface file. For example, static routes for the `eth0` interface would be stored in the /etc/sysconfig/network-scripts/route-`eth0` file.

The route-interface file has two formats:

- IP command arguments
- Network/netmask directives

The IP command arguments format uses the following syntax:

```
x.x.x.x/x via x.x.x.x dev interface
```

Use the term `default` to create a default gateway, for example:

```
default via x.x.x.x dev interface
```

The following example creates a static route to the 192.168.2.0/24 subnet through an `eth1` interface (10.10.10.1):

```
# cat /etc/sysconfig/network-scripts/route-eth1
198.168.2.0/24 via 10.10.10.1 dev eth1
```

You can also use the network/netmask directives format for route-interface files. The format is as follows:

```
ADDRESS0=X.X.X.X NETMASK0=X.X.X.X GATEWAY0=X.X.X.X
```

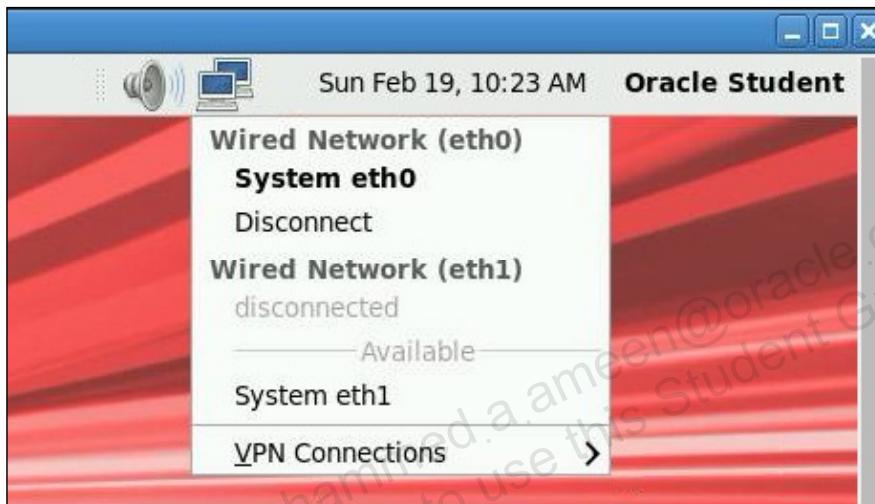
The following example shows using the IP command arguments to define the same entry:

```
ADDRESS0=198.168.2.0
NETMASK0=255.255.255.0
GATEWAY0=10.10.10.1
```

Start at 0 (as shown) and increment by one for each additional static route.

## NetworkManager

- NetworkManager:
  - Dynamically detects and configures network connections
  - Includes a GNOME Notification Area applet
- Click the icon to display the drop-down menu.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

ORACLE

NetworkManager dynamically detects and configures network connections. It includes a GNOME Notification Area applet that provides network status information and GUI configuration tools to manipulate network connections and interfaces.

Network configuration tasks are performed by using NetworkManager's Notification Area applet. If the applet does not appear in the GNOME panel, as root user, use the following commands to ensure that the package is installed and the service is started:

```
# yum install NetworkManager  
# service NetworkManager start
```

Start the GNOME Notification Area applet, as a non-root user, by using the following commands :

```
$ nm-applet &
```

The applet has multiple states that serve as visual indicators for the different types of connections that you have. Hold the mouse pointer over the applet icon to display tool tip information. Clicking the icon displays a drop-down menu. A sample menu is shown in the slide.

This sample menu indicates that there are two wired networks available (`eth0` and `eth1`) and you are currently connected to `eth0`, as indicated by **System eth0**. Wireless networks, if present, would also be listed on this menu.

Because `eth0` is active, it also has a **Disconnect** option. Clicking **Disconnect** displays an X character in a red box on the icon in NetworkManager's Notification Area to indicate that the network is inactive. Click the network name to reconnect. The X character in the red box disappears.

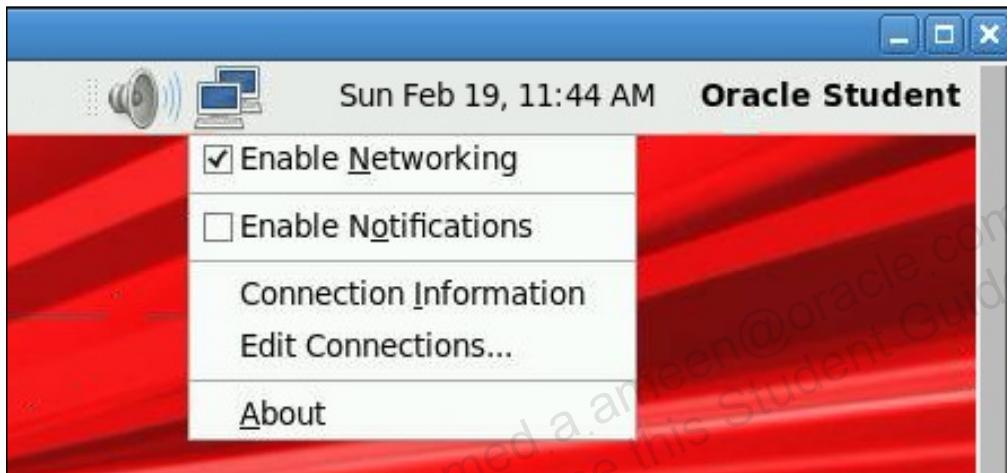
### `nm-tool`

NetworkManager includes the `nm-tool` command, which displays information about each network interface:

```
# nm-tool
NetworkManager Tool
State: connected
- Device: eth0 [ System eth0]
Type:          Wired
Driver:        vif
State:         connected
Default:       yes
HW Address:   00:16:3E:00:01:03
Capabilities:
  Carrier Detect: yes
Wired Properties
  Carrier:      on
IPv4 Settings:
  Address:      192.0.2.103
  Prefix:       24 (255.255.255.0)
  Gateway:      192.0.2.1
- Device: eth1
Type:          Wired
...
```

## NetworkManager

1. Right-click the icon to display the drop-down menu.
2. Select **Edit Connections** from the menu to display the Network Connections window.



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Right-clicking the icon displays a different drop-down menu, as shown in the slide. It is from this menu that network interface connections are configured.

The **Enable Networking** box must be selected. Selecting the **Enable Notifications** box causes NetworkManager to notify you of network connection status changes. Clicking **Connection Information** displays an informative Connection Information window, which lists the connection type, hardware address, IP address, and other useful information.

Clicking **Edit Connections** opens the Network Connections window, allowing you to configure the different types of connections. You can also open this window by using the `nm-connection-editor` command:

```
# nm-connection-editor &
```

## Network Connections Window



ORACLE®

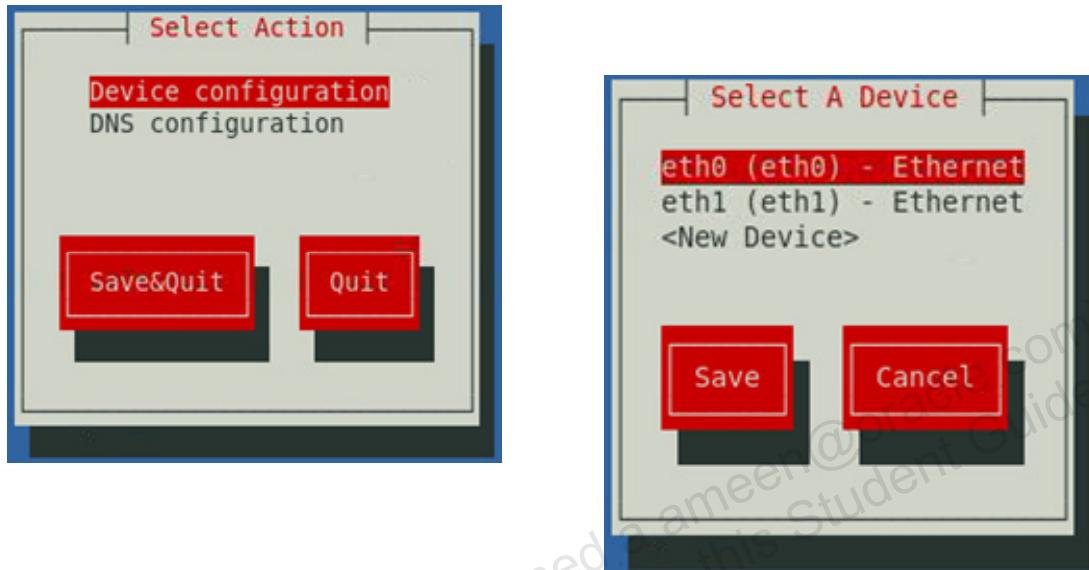
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The example in the slide shows the use of the Network Connections window to configure wired connections. The slide shows the two Ethernet connections that were seen on the NetworkManager's Notification Area applet menu. Select an entry from the list, then click **Edit** to modify parameters such as:

- MAC address
- MTU
- 802.1x security parameters
- IPv4 settings
- IPv6 settings

## system-config-network Utility

```
# system-config-network
```



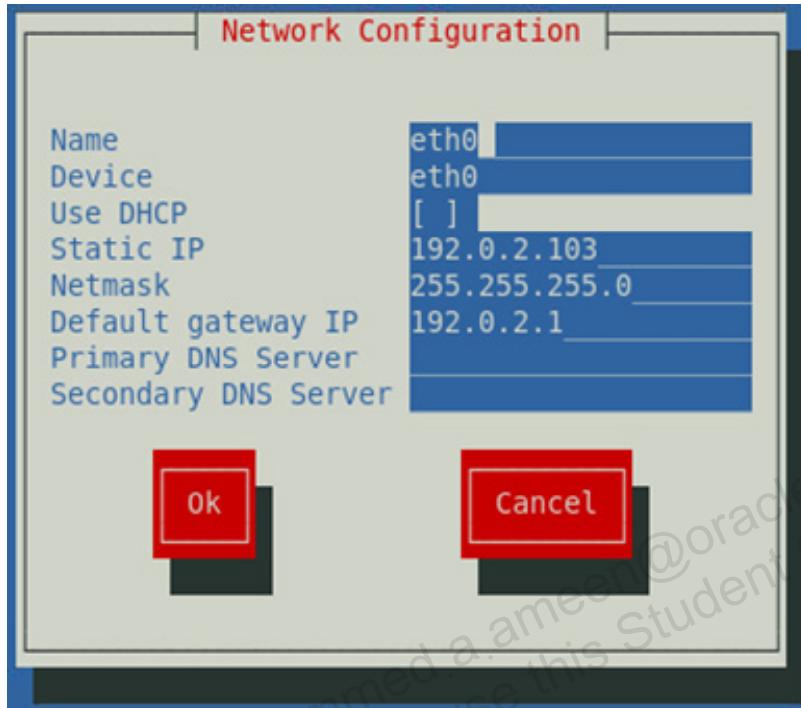
ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

A text-based user interface (TUI) exists to enter network interface configuration information. Enter the `system-config-network` command to display the first of the two screens shown in the slide. Use the Tab key or arrow keys to highlight a selection.

Highlight **Device configuration** and press **Enter** to set network interface properties. The second screen appears, from which an existing device, or a new device, can be configured. In this example, selecting **eth0 (eth0) – Ethernet** displays the screen on the next page.

## Device Configuration



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

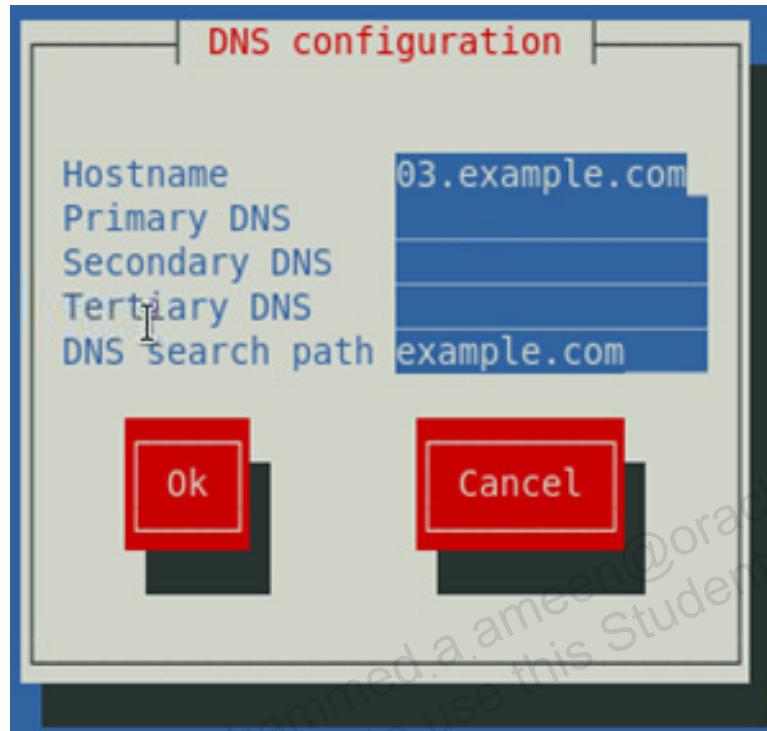
Enter the desired information, then highlight **Ok** and press **Enter**.

To use Dynamic Host Configuration Protocol (DHCP), highlight **Use DHCP** and press the space bar to toggle the on/off selection. An asterisk (\*) appears in this field when selected and you cannot enter **Static IP**, **Netmask**, or **Default gateway IP** information. Systems configured to use DHCP automatically obtain this information from a DHCP server.

Information entered and saved through this utility is written to the respective /etc/sysconfig/network-scripts/ifcfg-<interface> file.

The state of the network interface does not change. You still must activate the interface by using either the `ifup` command or the service `network start` (or `restart`) command.

# DNS Client Configuration



ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Selecting **DNS Configuration** from the initial screen displays the screen shown on this page. This screen is used to update the `/etc/resolv.conf` file.

To save all changes, select **Save&Quit** from the initial screen.

# Quiz

Which of the following statements is true?

- a. Network interface configuration files are located in the /etc/sysconfig/network directory.
- b. The NetworkManager applet can be used only to configure wired Ethernet devices.
- c. Routing tables can be displayed by using the netstat -r command.
- d. The ifup eth0 command activates the eth0 interface.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Summary

In this lesson, you should have learned how to:

- Describe network interface configuration files
- Use command-line network interface utilities
- Describe ARP and the ARP cache
- Describe network interface bonding
- Describe VLAN configuration
- Use the `route` utility to manipulate routing table
- Use the NetworkManager tool to configure network connections
- Use the `system-config-network` utility



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 13: Overview

The practices for this lesson cover the following:

- Configuring the `eth1` network interface
- Using NetworkManager
- Using the `system-config-network` utility
- Accessing the Public Yum Server

# 14

## File Sharing

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe NFS
- Configure NFS server and client
- Describe the `exportfs` utility
- Describe and configure automounter
- Describe and configure `vsftpd`

## Introduction to NFS

- NFS allows a Linux server to share directory hierarchies with Linux clients over a network.
- NFS servers export the directory, and NFS clients mount the exported directory.
- Oracle Linux supports three versions:
  - NFSv2
  - NFSv3
  - NFSv4
- NFS relies on Remote Procedure Calls (RPC) between clients and servers.
- Several `nfs` and `rpc` services work together, depending on which version of NFS is implemented.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

A Network File system (NFS) allows a server to share directory hierarchies (file systems) with remote systems over a network. NFS servers *export* the directory and NFS clients *mount* the exported directory. The server directory then appears to the client systems as if they were local directories. NFS reduces storage needs and improves data consistency and reliability, because users are accessing files that are stored on a centralized server.

The Linux 2.6 (and later) kernel supports three versions of the NFS protocol:

- NFS version 2 (NFSv2), specification is RFC 1094
- NFS version 3 (NFSv3), specification is RFC 1813
- NFS version 4 (NFSv4), specification is RFC 3530

NFS relies on Remote Procedure Calls (RPC) between clients and servers. RPC services are controlled by the `rpcbind` service. The `rpcbind` service replaces `portmap`, which was used in previous versions of Linux to map RPC program numbers to IP address port number combinations. `rpcbind` responds to requests for RPC services and sets up connections to the requested RPC service.

`rpcbind` is not used with NFSv4, because the server listens on well-known TCP port 2049. The mounting and locking protocols have also been incorporated into the NFSv4 protocol, so NFSv4 does not interact with the `lockd` and `rpc.statd` daemons either.

The following are descriptions of other services used by various versions of NFS:

#### **nfs**

Starting this service starts the NFS server and other RPC processes needed to service requests for shared NFS file systems. Example:

```
# service nfs start
Starting NFS services: [ OK ]
Starting NFS quotas: [ OK ]
Starting NFS daemon: [ OK ]
Starting NFS mountd: [ OK ]
```

To display the names of the services (process IDs are examples only):

```
# service nfs status
rpc.svcgssd is stopped
rpc.mountd (pid 9874) is running...
nfsd (pid 9871 9870 9869 9868 9867 9866 9865 9864) is running...
rpc.rquotad (pid 9859) is running...
```

#### **rpc.nfsd**

This is the NFS server process that implements the user level part of the NFS service. The main functionality is handled by the `nfsd` kernel module. The user space program merely specifies what sort of sockets the kernel server should listen on, what NFS versions it should support, and how many kernel threads it should use.

Notice that multiple (8) `nfsd` threads are running in the `service nfs status` output example. Reading the `/proc/fs/nfsd/threads` file also shows the number of threads. Writing a number to this file causes the number of threads to be changed to that number.

Example:

```
# cat /proc/fs/nfsd/threads
8
# echo 7 > /proc/fs/nfsd/threads
# service nfs status | grep nfsd
nfsd (pid 9870 9869 9868 9867 9866 9865 9864) is running...
```

#### **rpc.mountd**

This is the NFS mount daemon that implements the server side of the mount requests from NFSv2 and NFSv3 clients. It checks that the requested NFS share is currently exported by the NFS server, and that the client is allowed to access it. For NFSv4, the `rpc.mountd` daemon is required only on the NFS server to set up the exports.

#### **rpc.rquotad**

This process provides user quota information for remote users. It is started automatically by the `nfs` service and does not require user configuration. The results are used by the `quota` command to display user quotas for remote file systems and by the `edquota` command to set quotas on remote file systems.

**nfslock**

Starting this service starts the RPC processes that allow NFS clients to lock files on the server. `nfslock` is needed only for NFSv2 and NFSv3.

**rpc.statd**

This process implements the Network Status Monitor (NSM) RPC protocol, which notifies NFS clients when an NFS server is restarted without being gracefully brought down. This is not used with NFSv4. `rpc.statd` does not require user configuration and is started automatically by the `nfslock` service:

```
# service nfslock start
Starting NFS statd: [ OK ]
```

**lockd**

This is a kernel thread that runs on both clients and servers. It implements the Network Lock Manager (NLM) protocol, which allows NFSv2 and NFSv3 clients to lock files on the server. It is started automatically whenever the NFS server is run and whenever an NFS file system is mounted. This process, along with `rpc.statd`, is stopped when stopping the `nfslock` service:

```
# service nfslock stop
Stopping NFS locking: [ OK ]
Stopping NFS statd: [ OK ]
```

**rpc.svcgssd/rpc.gssd**

These are the `rpcsec_gss` daemons. The `rpcsec_gss` protocol gives a means of using the `gss-api` generic security API to provide security for protocols using RPC. Before exchanging any RPC requests using `rpcsec_gss`, the RPC client must first establish a security context with the RPC server. Refer to `man exports` for a description of implementing RPCSEC\_GSS security.

**rpc.idmapd**

This provides NFSv4 client and server upcalls, which map between on-the-wire NFSv4 names (which are strings in the form of `user@domain`) and local UIDs and GIDs. For `idmapd` to function with NFSv4, `/etc/idmapd.conf` must be configured. This service is required for use with NFSv4, although not when all hosts share the same DNS domain name.

# NFS Server Configuration

- Install the `nfs-utils` package.
- The main configuration file for the NFS server is `/etc(exports`.
  - It contains a list of exported directory hierarchies that remote systems can mount.
- The format of `/etc(exports` entries is:
  - `dir client1(options) [client2(options) ...]`
- Client options include (defaults are listed first):
  - `ro / rw`
  - `sync / async`
  - `wdelay / no_wdelay`
  - `no_all_squash / all_squash`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

To begin configuring a system as an NFS server, install the `nfs-utils` package:

```
# yum install nfs-utils
```

The main configuration file for the NFS server is `/etc(exports`. The `/etc(exports` file stores a list of exported directory hierarchies that remote systems can mount. The format for entries is:

```
export-point client1(options) [client2(options) ... ]
```

The `export-point` is the absolute path name of the directory hierarchy to be exported. One or more client systems, each with specific options, can mount `export-point`. There are no spaces between the client attribute and the open bracket. When no client options are specified, the following default settings apply:

- **ro**: Read-only. Client hosts cannot change the data shared on the file system. To allow client hosts to make changes to the file system, specify the `rw` (read/write) option.
- **sync**: The NFS server replies to requests only after changes made by previous requests are written to disk. `async` specifies that the server does not have to wait.
- **wdelay**: The NFS server delays committing write requests when it suspects another write request is imminent. To disable the delay, use the `no_wdelay` option. `no_wdelay` is available only if the default `sync` option is also specified.

- **root\_squash**: Prevents root users connected remotely from having root privileges, effectively “squashing” the power of the remote root user. Requests appear to come from the user nfsnobody, an unprivileged user on the local system, or as specified by anonuid. To disable root squashing, specify the no\_root\_squash option.
- **no\_all\_squash**: Does not change the mapping of remote users. To squash every remote user (including root), use the all\_squash option.

To specify the user ID (UID) and group ID (GID) that the NFS server should assign to remote users, use the anonuid and anongid options as follows:

```
export-point client (anonuid=uid,anongid=gid)
```

The anonuid and anongid options allow you to create a special user and group account for remote NFS users to share. By default, access control lists (ACLs) are supported by NFS. To disable this feature, specify the no\_acl option when exporting the file system.

You can use wildcard characters, such as (\*) and (?) in client names. You can also export directories to all hosts on an IP network. To do this, specify an IP address and netmask pair as address/netmask. Either of the following forms is valid:

- 192.168.1.0/24
- 192.168.1.0/255.255.255.0

Other client options exist. Refer to `man exports` for descriptions of all options.

### /etc(exports Examples

In the following example, a client system with the IP address of 192.0.2.102 can mount the /export/directory with read/write permissions. All writes to the disk are asynchronous:

```
/export/directory 192.0.2.102(rw,async)
```

The following example exports the /exports/apps directory to all clients, converts all connecting users to the local anonymous nfsnobody user, and makes the directory read-only:

```
/exports/apps *(all_squash, ro)
```

The following example exports the /spreadsheets/proj1 directory with read-only permissions to all clients on the 192.168.1.0 subnet, and read-write permissions to the client system named mgmtpc:

```
/spreadsheets/proj1 192.168.1.0/24(ro) mgmtpc(rw)
```

# Starting the NFS Service

- Start rpcbind before starting the nfs services:
  - # service rpcbind start
  - # service nfs start
  - # service nfslock start
- Use the chkconfig command to automatically start the services at boot time.
- Specify configuration options and arguments in /etc/sysconfig/nfs.
- To display exported file systems:
  - # showmount -e

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Start the NFS Service

The rpcbind service must be started before starting nfs:

```
# service rpcbind start
```

If the rpcbind service is running, then the nfs service can be started. Restart nfs after making any configuration changes in /etc(exports or run the exportfs -a command.

```
# service nfs start
```

Start the nfslock service:

```
# service nfslock start
```

Use the chkconfig command to automatically start the services at boot time, for example:

```
# chkconfig rpcbind on; chkconfig nfs on; chkconfig nfslock on;
```

Specify configuration options and arguments by placing them in /etc/sysconfig/nfs.

Use the showmount -e command to display exported file systems:

```
# showmount -e
```

```
Export list for host03.example.com
```

```
/Test *
```

## exportfs Utility

- `exportfs` exports or unexports directories, and is run from the command line.
  - No need to change `/etc/exports`
  - No need to restart NFS service
- Syntax for the command:
  - `exportfs [options] [client:dir ...]`
- Example:
  - `# exportfs -i -o rw *:/Dev`
- This example does the following:
  - Exports `/Dev`
  - To all clients systems (`*`)
  - Allows read-write permission (`-o rw`)
  - Ignores `/etc/exports` entries (`-i`)



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

You can also configure an NFS server from the command line by using `exportfs`. This command allows the `root` user to selectively export or unexport directories without changing `/etc/exports` and without restarting the NFS service. The syntax for the command is:

```
exportfs [options] [client:dir ...]
```

The `client` argument is the name of the client system that `dir` is exported to. The `dir` argument is the absolute path name of the directory being exported. The following is a list of some of the options:

- `-r`: Re-export the entries in `/etc/exports` and synchronize `/var/lib/nfs/etab` with `/etc/exports`. The NFS server runs `exportfs -r` when the service starts (this command is in the `nfs` init script, `/etc/init.d/nfs`). The `/var/lib/nfs/etab` file is the master export table. `rpc.mountd` reads this file when a client sends an NFS mount command.
- `-a`: Export the entries in `/etc/exports` but do not synchronize `/var/lib/nfs/etab`. Run `exportfs -a` after making any configuration changes.
- `-i`: Ignore the entries in `/etc/exports` and use only command-line arguments.
- `-u`: Unexport one or more directories.
- `-o`: Specify client options as specified in `/etc/exports`.

# NFS Client Configuration

- Use the `mount` command to mount exported file systems.
- Syntax for the command:
  - `mount -t nfs -o options host:/remote/export /local/directory`
- Example:
  - `# mount -t nfs -o ro,nosuid abc:/home /abc_home`
- This example does the following:
  - Mounts `/home` from remote host `abc`
  - On local mount point `/abc_home`
  - Read-only, and prevents users from running a setuid program (-o `ro,nosuid` options)
- Update `/etc/fstab` to mount NFS shares at boot time.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Use the `mount` command to mount exported file systems (NFS shares) on the client side. Syntax for the command is:

```
mount -t nfs -o options host:/remote/export /local/directory
```

The following are descriptions of the arguments:

- **-t nfs:** Indicates that the file system type is `nfs`. With this option, `mount` uses NFSv4 if the server supports it; otherwise, it uses the highest version supported by the server.
- **-o options:** A comma-delimited list of mount options
- **host:/remote/export:** The host name exporting the file system, followed by a colon, followed by the absolute path name of the NFS share
- **/local/directory:** The mount point on the client system

For example, to mount the `/home` directory exported from host `abc` with read-only permissions (`ro` option) on local mount point `/abc_home`, and prevent remote users from gaining higher privileges by running a setuid program (`nosuid` option):

```
# mount -t nfs -o ro,nosuid abc:/home /abc_home
```

For a list of options used for NFS mounts, see the MOUNT OPTIONS section of `man nfs`.

For a list of client mount options, see the FILESYSTEM INDEPENDENT MOUNT OPTIONS section of `man mount`.

## /etc/fstab

To mount NFS shares at boot time, add entries to the file system mount table, /etc/fstab. Entries are in the following format:

```
server:/exported-filesystem local_mount_point nfs options 0 0
```

For example, the /etc/fstab entry that replicates the `mount` command on the previous page is:

```
abc:/home /abc_home nfs ro,nosuid 0 0
```

The `df` command displays mounted file systems, including NFS-mounted file systems. For NFS mounts, the “File system” column displays the `server:/exported-filesystem` information. Use the `-T` option to include a “Type” column:

```
# df -hT
```

# Automounting File Systems

- Remote file systems are mounted only when accessed.
- Install the `autofs` package.
  - `autofs`: Kernel module
  - `automount`: Userspace daemon
- The main configuration file is `/etc/auto.master`.
- Format of `/etc/auto.master` entries:
  - `/key map-file [options]`
- Format of map-file entries:
  - `key [options] location`
- Direct maps, indirect maps, host maps



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Automounting is an alternative to creating NFS mount entries in `/etc/fstab` or using the `mount` command from the command line to mount NFS shares. Automounting mounts remote file systems when they are accessed, rather than maintaining these remote mounts at all times. When the remote file systems are inactive, they are unmounted. This frees up system resources and improves overall system performance.

Automounting consists of two components:

- `autofs`: Kernel module
- `automount`: User-space daemon

To implement automounting, first install the `autofs` package:

```
# yum install autofs
```

The main configuration file, known as the master map-file, is `/etc/auto.master`. This file lists mount points, known as keys, and corresponding map-files that indicate which remote file systems can be mounted on the key. The format for entries in `/etc/auto.master` is:

```
/key      map-file      [options]
```

Automounting supports direct maps, indirect maps, and host maps. Direct maps use a special key, `/-`, in `/etc/auto.master`. Indirect maps specify a relative path name in their map-file. Host maps use a special map, `-hosts`, in the `/etc/auto.master` file.

Options included in the master map-file apply to all entries in the associated map-file, unless the map-file specifies mount options. The following is a sample listing of the /etc/auto.master file:

```
# cat /etc/auto.master
/-          auto.direct
/misc       /etc/auto.misc
/net        -hosts
+auto.master
```

The first entry represents a direct map, which begins with / -. The second entry represents an indirect map, the relative path name is included in the associated map-file. The third entry represents a host map, and uses the special map -hosts. The last entry is an example of how to include a map from its source as if it were present in the master map. Maps to be included are preceded with a plus sign (+).

## Map-Files

Map-files have the following format:

```
key [options] location
```

The key can be a single directory name for an indirect map or the absolute path name of the mount point for direct mounts. Mount options can be included in map-files. Any options specified in map-files override options specified in the master map-file. The location is the exported NFS file system, or a local file system, or any other supported file system type.

### Direct Maps

The first line in the sample /etc/auto.master listing is a direct map. With direct maps, the key is always / - and the map-file, auto.direct in this example, contains the absolute path name of the directory to be mounted. The following is a sample of the auto.direct map-file:

```
/usr/man -ro,soft host01:/usr/man
```

This entry mounts the file system /usr/man from the server host01. automount creates the /usr/man directory if it does not already exist. If /usr/man does exist and is not empty, the mounted file system hides the local existing file system.

### Indirect Maps

Indirect maps are more common than direct maps. The following is an example of an indirect map named /etc/auto.misc:

```
# cat /etc/auto.misc
xyz      -fstype=nfs           host01:/xyz
cd       -fstype=iso9600,ro,nosuid,nodev :/dev/cdrom
abc      -fstype=ext3           :/dev/hda1
kernel   -ro,soft,intr         ftp.kernel.org:/pub/linux
windoz   -fstype=smbfs          ://windoz/c
```

The key field is relative to the actual location of the autofs mount point, /misc, from the master map-file, /etc/auto.master. For example, entering the command cd /misc/xyz mounts the /xyz directory from machine host01 locally on /misc/xyz. Only the /misc mount point needs to exist on the local machine. For indirect maps, the key is created when the file system is accessed and then removed when the file system is unmounted.

The second and third entries are examples of automounting local file systems:

```
cd           -fstype=iso9600,ro,nosuid,nodev      :/dev/cdrom
abc          -fstype=ext3                         :/dev/hda1
```

The location field is the local file system path preceded with a colon (:). Entering the `ls /misc/cd` command would display the contents of the `iso` file on the `cdrom`. Entering the `ls /misc/abc` command would display the contents of the `ext3` file system on the `hda1` device.

The fourth line is an NFS mount (excluding the `-fstype` option defaults to NFS), which mounts the `/pub/kernel` directory from `ftp.kernel.org` on local mount point `/misc/kernel`:

```
kernel      -ro,soft,intr                      ftp.kernel.org:/pub/linux
```

The last line mounts a share exported from a Windows machine on `/misc/windoz`:

```
windoz     -fstype=smbfs                      :://windoz/c
```

## Host Map

The third line in the sample `/etc/auto.master` file gives the following entry:

```
/net        -hosts
```

When `-hosts` is given as the map, the `automount` daemon creates a subdirectory under the `/key` directory for every server listed in the `/etc/hosts` file. For example, entering the following command mounts all exports from `host03` over the `/net/host03` directory:

```
# cd /net/host03
```

All exports are mounted with the “`no-suid, nodev, intr`” options by default.

## Start the `autofs` Service

The main configuration file for `autofs` is `/etc/sysconfig/autofs`.

To start the `autofs` service:

```
# service autofs start
```

Changes made to map-files become effective immediately. If the master map-file, `/etc/auto.master`, is modified, the `autofs` service must be restarted.

## Introduction to vsftpd

- vsftpd allows a system to function as an FTP server.
- vsftpd includes the following configuration files and directories:
  - /etc/vsftpd/vsftpd.conf
  - /etc/vsftpd/ftpusers
  - /etc/vsftpd/user\_list
  - /var/ftp
- To start the service:
  - service vsftpd start
- To start automatically at boot time:
  - chkconfig vsftpd on



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

File Transfer Protocol (FTP) is a commonly used method of downloading and uploading files between systems on a network. FTP sites are typically public sites that allow anonymous users to log in and download software and documentation without needing a user account on the remote system.

The FTP server daemon included with Oracle Linux is called “very secure FTP,” or vsftpd. To install the vsftpd package:

```
# yum install vsftpd
```

The following configuration files are installed with the package:

- **/etc/vsftpd/vsftpd.conf:** The main configuration file for vsftpd
- **/etc/vsftpd/ftpusers:** A list of users not allowed to log in to vsftpd
- **/etc/vsftpd/user\_list:** This file contains users who are denied access when the userlist\_deny directive is set to YES (default) in /etc/vsftpd/vsftpd.conf or users who are allowed access when userlist\_deny is set to NO.
- **/var/ftp:** The directory containing files served by vsftpd. It also contains the /var/ftp/pub directory for anonymous users.

To start the vsftpd service:

```
# service vsftpd start
```

## vsftpd Configuration Options

- Local and anonymous users can download files by default.
  - local\_enable=YES
  - anonymous\_enable=YES
- Users can upload files by default, too.
  - write\_enable=YES
- Additional configuration parameters in /etc/vsftpd/vsftpd.conf include:
  - listen
  - userlist\_enable
  - userlist\_deny
  - no\_anon\_password
  - xferlog\_enable
  - xferlog\_file



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The vsftpd service allows local and anonymous users to log in without any additional configuration. When a user logs in, they can download files from the /var/ftp directory on the vsftpd server and upload files by default.

These and other options are configured in /etc/vsftpd/vsftpd.conf. The following lists some of the more common configuration parameters:

- **listen**: This setting decides whether vsftpd should run stand-alone (default) and listen to the FTP port itself or allow itself to be run by a superserver, such as xinetd.
- **userlist\_enable**: This setting tells vsftpd to read /etc/vsftpd/user\_list and use that as a list of users to allow or not allow on the server.
- **userlist\_deny**: When set to yes, vsftpd blocks all users in the user\_list. When set to no, it allows only users in the user\_list.
- **local\_enable**: This setting allows users in /etc/passwd to log in with their accounts.
- **anonymous\_enable**: This setting allows anonymous connections to the server.
- **no\_anon\_password**: This setting allows anonymous connections without a password (otherwise, users must provide an email address as a password).
- **write\_enable**: When set to yes, this setting allows users to upload files to the server and create directories.

- **anon\_mkdir\_write\_enable**: When set to yes, this setting allows anonymous users to create directories.
- **anon\_other\_write\_enable**: When set to yes, this setting allows anonymous users to make other changes to the file system, such as deleting, renaming, and modifying existing files.
- **anon\_upload\_enable**: This setting allows anonymous users to upload files to the server.
- **ascii\_download\_enable**: This setting allows conversion of text files transferred from the server to other operating systems. This can be a good idea if you are transferring text files from UNIX systems to Mac OS or Windows.
- **ascii\_upload\_enable**: This setting allows conversion of text files uploaded to the server.
- **xferlog\_enable**: This setting activates logging of uploads and downloads.
- **xferlog\_file**: This setting names the upload/download log file. The default is /var/log/vsftpd.log.

# Quiz

Which of the following statements are true?

- a. NFS allows Linux clients to mount exported file systems from remote Linux systems.
- b. Automounter allows NFS shares to be automatically mounted.
- c. The `vsftpd` daemon enables a system to be configured as an FTP server.

## Summary

In this lesson, you should have learned how to:

- Describe NFS
- Configure NFS server and client
- Describe the `exportfs` utility
- Describe and configure automounter
- Describe and configure `vsftpd`

## Practice 14: Overview

The practices for this lesson cover the following:

- Configuring an NFS server and an NFS client
- Using automounter
- Configuring an FTP server
- Downloading a file from an FTP server
- Configuring xinetd

# 15

## OpenSSH

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe OpenSSH
- Describe OpenSSH configuration files
- Configure OpenSSH server and client
- Use the `ssh` command
- Use the `scp` command
- Use the `sftp` command
- Use the `ssh-keygen` command
- Use `ssh-agent` and `ssh-add`

# Introduction to OpenSSH

## OpenSSH:

- Is a suite of secure network connectivity tools:
  - `ssh`: Secure shell command
  - `scp`: Secure copy command
  - `sftp`: Secure FTP (file transfer protocol) command
  - `sshd`: The OpenSSH daemon
  - `ssh-keygen`: Creates DSA or RSA authentication keys
- Is a secure alternative to `telnet`, `rcp`, `rsh`, `rlogin`, and `ftp`
- Encrypts all communication between the client and server
- Supports both the SSH1 and SSH2 protocols
- Provides X11 forwarding and port forwarding



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

OpenSSH (Secure Shell) is a suite of network connectivity tools that provides secure communications between systems. OpenSSH tools include the following:

- `ssh`: Secure shell logs on or runs a command on a remote system
- `scp`: Secure copy
- `sftp`: Secure `ftp` (file transfer protocol)
- `sshd`: The OpenSSH daemon
- `ssh-keygen`: Creates DSA or RSA host/user authentication keys:
  - DSA (Digital Signature Algorithm)
  - RSA is named for the designers Rivest, Shamir, and Adleman.

Unlike other tools such as `telnet`, `rcp`, `rsh`, `rlogin`, and `ftp`, OpenSSH tools encrypt all communication between the client and server systems, including passwords. Each network packet is encrypted using a key known only by the local and remote systems.

OpenSSH supports both versions of SSH, SSH protocol version 1 (SSH1) and SSH protocol version 2 (SSH2). Additionally, OpenSSH provides a secure means to use graphical applications over a network by using X11 forwarding. It also provides a way to secure otherwise insecure TCP/IP protocols by using port forwarding.

# OpenSSH Configuration Files

- Global files are stored in the /etc/ssh directory.
- User files are stored in the ~/.ssh directory.
- Global files include:
  - ssh\_config: The default OpenSSH client configuration file
  - sshd\_config: The configuration file for the sshd daemon
  - Various DSA and RSA public and private key files
- User files include:
  - config: Overrides global ssh\_config file
  - known\_hosts: Contains host keys of SSH servers accessed by the user
  - Various user DSA and RSA public and private key files



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

OpenSSH clients and servers have several configuration files. Global configuration files are stored in the /etc/ssh directory. User configuration files are stored in an .ssh directory in user home directories (~/.ssh).

## /etc/ssh: Global Files

The following are brief descriptions of the global configuration files:

- **moduli**: Contains key exchange information used to establish a secure connection
- **ssh\_config**: The default OpenSSH client configuration file. Entries are overridden by a user's ~/.ssh/config file.
- **sshd\_config**: The configuration file for the sshd daemon
- **ssh\_host\_dsa\_key**: The DSA private key for version SSH2
- **ssh\_host\_dsa\_key.pub**: The DSA public key for version SSH2
- **ssh\_host\_key**: The RSA private key for version SSH1
- **ssh\_host\_key.pub**: The RSA public key for version SSH1
- **ssh\_host\_rsa\_key**: The RSA private key for version SSH2
- **ssh\_host\_rsa\_key.pub**: The RSA public key for version SSH2

## ~/.ssh: User Files

OpenSSH creates the `~/.ssh` directory and the `known_hosts` file automatically when you connect to a remote system. The following are brief descriptions of the user-specific configuration files:

- **authorized\_keys**: Contains a list of authorized public keys for SSH servers. The server authenticates the client by checking its signed public key within this file.
- **id\_dsa**: The DSA private key of the user
- **id\_dsa.pub**: The DSA public key of the user
- **id\_rsa**: The RSA private key for version SSH2
- **id\_rsa.pub**: The RSA public key for version SSH2
- **identity**: The RSA private key for version SSH1
- **identity.pub**: The RSA public key for version SSH1
- **known\_hosts**: Contains host keys of SSH servers accessed by the user. OpenSSH automatically adds entries each time the user connects to a new server.

# OpenSSH Configuration

- To configure an OpenSSH server:
  - The following packages are installed by default:
    - openssh
    - openssh-server
  - Start the sshd daemon:
    - service sshd start
- To configure an OpenSSH client:
  - The following packages are installed by default:
    - openssh
    - openssh-client
  - There are no services to start on the client.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## OpenSSH Server

To begin configuring a system as an OpenSSH server, install the following packages (these are installed by default):

```
# yum install openssh  
# yum install openssh-server
```

Start the sshd daemon:

```
# service sshd start
```

Use the chkconfig command to automatically start the sshd service at boot time:

```
# chkconfig sshd on
```

## OpenSSH Client

To configure a system as an OpenSSH client, install the following packages (these are installed by default):

```
# yum install openssh  
# yum install openssh-client
```

There are no services to start for OpenSSH clients.

## Using OpenSSH Utilities

- All OpenSSH utilities require a remote user account.
- The first time you connect to an OpenSSH server, the OpenSSH client prompts you to confirm that you are connected to the correct system:

```
$ ssh host03  
The authenticity of host 'host03 (192.0.2.103)'  
can't be established. RSA key fingerprint is  
...  
Are you sure you want to continue connecting  
(yes/no)? yes  
Warning: Permanently added 'host03,192.0.2.103'  
(RSA) to the list of known hosts.
```

- The user's `~/.ssh/known_hosts` file is updated.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

All of the OpenSSH tools require that you have a user account on the remote system. Each time you attempt to connect to a remote system, you must provide a username and password for the remote system.

When you connect to an OpenSSH server for the first time, the OpenSSH client prompts you to confirm that you are connected to the correct system. The following example uses the `ssh` command to connect to a remote host named `host03`:

```
$ ssh host03  
The authenticity of host 'host03 (192.0.2.103)' can't be  
established. RSA key fingerprint is ...  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'host03,192.0.2.103' (RSA) to the  
list of known hosts.
```

Host validation is one of OpenSSH's major features. The command checks to make sure that you are connecting to the host that you think you are connecting to. When you enter `yes`, the client appends the server's public host key to the user's `~/.ssh/known_hosts` file, creating the `~/.ssh` directory if necessary. The next time you connect to the remote server, the client compares this key to the one the server supplies. If the keys match, you are not asked if you want to continue connecting.

If someone tries to trick you into logging in to their machine so that they can sniff your SSH session, you will receive a warning similar to the following:

```
@@@@@@@WARNING: POSSIBLE DNS SPOOFING DETECTED! @
@@@@@@@The RSA host key for ... has changed,
and the key for the according IP address ...
is unchanged. This could either mean that
DNS SPOOFING is happening or the IP address for the host
and its host key have changed at the same time.
Offending key for IP in /home/<user>/.ssh/known_hosts:10
@@@@@@@WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
@@@@@@@IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle
attack) !
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is ...
Please contact your system administrator.
Add correct host key in /home/<user>/.ssh/known_hosts to get rid
of this message.
Offending key in /home/<user>/.ssh/known_hosts:53
RSA host key for ... has changed and you have requested strict
checking.
Host key verification failed.
```

If you ever get a warning like this, stop and determine whether there is a reason for the remote server's host key to change (such as if SSH was upgraded or the server itself was upgraded). If there is no good reason for the host key to change, then you should not try to connect to that machine until you have contacted its administrator about the situation.

# Using the ssh Command

- The ssh command:
  - Allows you to connect to a remote system
  - Allows you to execute a command on a remote system
- The format of the command is:
  - `ssh [options] [user@] host [command]`
- Examples:
  - `ssh host03`
  - `ssh root@host03`
  - `ssh host03 ls`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `ssh` command allows you to connect to a remote system, or to execute a command on a remote system.

The format of the `ssh` command is:

```
ssh [options] [user@] host [command]
```

The `host` argument is the name of the OpenSSH server that you want to connect to, and is the only required argument. For example, to connect to a remote host named `host03`, you only need to enter the following:

```
$ ssh host03
```

This command attempts to connect to the remote host with the same username that you are logged on as on the local system. You are prompted for only the remote user's password. To connect to a remote host as a different user, provide the `user@` argument:

```
$ ssh root@host03
```

To execute a command on a remote system, include the command as an argument. `ssh` logs you in, executes the command, and then closes the connection, for example:

```
$ ssh host03 ls
```

## Using the `scp` Command

- Use `scp` to copy files or directories to or from a remote system.
- To copy to a remote system, the format is:
  - `scp [options] local-file [user@] to-host [:remote-file]`
- Examples:
  - `scp test host03`
  - `scp test host03:new_test`
- To copy from a remote system, the format is:
  - `scp [options] [user@] from-host:remote-file local-file`
- Examples:
  - `$ scp host03:new_test .`
  - `$ scp host03:new_test newer_test`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `scp` command allows you to copy files or directories (use the `-r` option to copy directories) between remote systems. A connection is established, files are copied, and the connection closes.

To copy a file to a remote system (upload), the format of the `scp` command is:

```
scp [options] local-file [user@] to-host [:remote-file]
```

For example, to copy a file named `test` to the remote user's home directory on `host03`:

```
$ scp test host03
```

To copy the same file to the same location but rename it to `new_test`:

```
$ scp test host03:new_test
```

To copy a file from a remote system (download), the format of the `scp` command is:

```
scp [options] [user@] from-host:remote-file local-file
```

For example, to copy a file named `new_test` from user's home directory on remote `host03`:

```
$ scp host03:new_test .
```

To copy a file named `new_test` from user's home directory on remote `host03` and rename it to `newer_test`:

```
$ scp host03:new_test newer_test
```

## Using the sftp Command

- sftp is a secure alternative to, and is functionally the same as, ftp.
- The format to connect to a remote system is:
  - sftp [options] [user@] host
- Example:
  - sftp host03
- You are presented with the sftp> prompt after connecting:
  - sftp>
- Enter help or ? to display a list of sftp commands.
- To upload a file (copy to remote system):
  - sftp> put *filename*
- To download a file (copy from a remote system):
  - sftp> get *filename*



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The sftp command is a secure alternative to ftp and is functionally the same as ftp. Use sftp instead of ftp when logging on to a server that is running the OpenSSH daemon, sshd.

The format to connect to a remote system is:

```
sftp [options] [user@] host
```

The following example assumes that you are logged on to your local system as user oracle and are connecting to a remote system named host03:

```
$ sftp host03
Connecting to host03...
oracle@host03's password:
sftp>
```

After providing the correct password, you are presented with an sftp> prompt as shown. Enter help or ? to display a list of available commands. The following example uploads a file, or copies the file from the local system to the remote system:

```
sftp> put newer
```

Enter exit, quit, or bye to close the connection and exit sftp.

## Using the ssh-keygen Command

- The ssh-keygen command generates authentication key pairs.
  - ssh-keygen -t rsa (generates RSA keys)
  - ssh-keygen -t dsa (generates DSA keys)
- ssh-keygen generates two keys:
  - Private key
  - Public key
- Specify a passphrase to encrypt the private part of the key.
- To allow remote connectivity without supplying a password:
  1. Copy the public key to ~/.ssh on the remote system.
  2. Do one of the following:
    - Rename the public key file name to authorized\_keys.
    - Append the public key to the authorized\_keys file on the remote system to allow connection from multiple clients.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Use the ssh-keygen command to generate a public/private authentication key pair. Authentication keys allow a user to connect to a remote system without supplying a password. Keys must be generated for each user separately. If you generate key pairs as the root user, only the root can use the keys.

The following example creates the public and private parts of an RSA key:

```
$ oracle@host03: ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/oracle/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/oracle/.ssh/id_rsa.
Your public key has been saved in /home/oracle/.ssh/id_rsa.pub.
The key fingerprint is:...
The key's randomart image is:...
```

Use the ssh-keygen -t dsa command to generate DSA keys.

You have the option of specifying a passphrase to encrypt the private part of the key. If you encrypt your personal key, you must supply the passphrase each time you use the key. This prevents an attacker, who has access to your private key and can impersonate you and access all the computers you have access to, from being able to do so. The attacker still needs to supply the passphrase.

The `ssh-key` command in the example generated two keys in the `~/.ssh` directory:

```
$ oracle@host03: ls ~/.ssh  
id_rsa  
id_rsa.pub
```

To log on to, or copy files to, a remote system without supplying a password, copy the public key (`~/.ssh/id_rsa.pub` in this example) to `~/.ssh/authorized_keys` on the remote system. Set the remote `~/.ssh` directory permissions to 700. You can then use the `ssh` or `scp` tools to access the remote system without supplying a password.

To allow multiple connections, append the public key to the `authorized_keys` file on the remote system instead of copying it. The following example appends the public key:

```
$ cat id_rsa.pub >> authorized_keys
```

You can improve system security even further by disabling the standard password authentication, and enforcing the key-based authentication. To do so, set the `PasswordAuthentication` option to `no` in the `/etc/ssh/sshd_config` configuration file as follows:

```
PasswordAuthentication no
```

This disallows users whose keys are not in the `authorized_keys` file of the specific user on the server to connect via `ssh`. The connection is denied and the following message appears:

```
$ ssh host01  
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

Setting the `PasswordAuthentication` option to `yes`, which is the default, permits a user to use a password for authentication.

## Using ssh-agent

- ssh-agent is an authentication agent that handles passwords for SSH private keys.
  - Use ssh-keygen to generate authentication key pairs as described in the previous slide.
  - Provide a passphrase, for example “password”, when creating the key pairs.
  - Copy the public key to `~/.ssh/authorized_keys` on the remote system as described in the previous slide.
- To use ssh-agent:
  - `$ exec ssh-agent $SHELL`
- Use ssh-add to add the keys:
  - `$ ssh-add`
  - Enter passphrase for `/home/oracle/.ssh/id_rsa`: **password**
  - Identity added: `/home/oracle/.ssh/id_rsa` ...



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The ssh-agent program is an authentication agent that handles passwords for SSH private keys. Use ssh-add to add the keys to the list maintained by ssh-agent. After you add a private key password to ssh-agent, you do not need to enter it each time you connect to a remote host with your public key.

Use the `ssh-keygen` command to generate authentication key pairs as described in the previous slide. Provide a passphrase, for example “password”, when creating the key pairs. Copy the public key to `~/.ssh/authorized_keys` on the remote system as described in the previous slide.

To add the private key password to ssh-agent, enter the following command:

```
$ exec ssh-agent $SHELL
```

The next step is to use the `ssh-add` command to add the key.

```
$ ssh-add
```

```
Enter passphrase for /home/oracle/.ssh/id_rsa: password
```

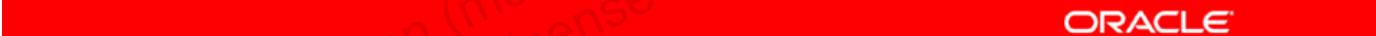
```
Identity added: /home/oracle/.ssh/id_rsa ...
```

In this example, the passphrase is remembered for only the current login session and will be forgotten when you log out.

# Quiz

OpenSSH connectivity tools encrypt all network traffic, including passwords.

- a. True
- b. False



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Summary

In this lesson, you should have learned how to:

- Describe OpenSSH
- Describe OpenSSH configuration files
- Configure OpenSSH server and client
- Use the `ssh` command
- Use the `scp` command
- Use the `sftp` command
- Use the `ssh-keygen` command
- Use `ssh-agent` and `ssh-add`

## Practice 15: Overview

The practices for this lesson cover the following:

- Connecting to a remote server by using ssh
- Configuring OpenSSH to connect without a password

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# 16

## Pluggable Authentication Modules (PAM)

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe the purpose of PAM
- Describe PAM configuration files
- Describe PAM authentication modules
- Describe PAM module types
- Describe PAM control flags
- Walk through PAM authentication examples

# Introduction to PAM

- PAM allows you to configure how applications use authentication to verify the identity of a user.
- Configuration files are located in the /etc/pam.d directory.
- Each configuration file has the same, or a similar, name as the application it authenticates, for example:
  - login, halt, reboot, sudo, sshd, samba
- Each configuration file lists authentication modules that contain the authentication code.
- Authentication modules are shared libraries located in /lib/security (and /lib64/security).
- PAM documentation includes man pages for most modules and SAG in /usr/share/doc/pam-<version>.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Pluggable Authentication Modules (PAM) is an authentication mechanism that allows you to configure how applications use authentication to verify the identity of a user. Configuration files, located in the /etc/pam.d directory, describe the authentication procedure for the application. Each of these configuration files have names that are the same as, or similar to, the name of the application that they authenticate. For example, there is a configuration file for the login process named login, a configuration file for the reboot application named reboot, and many others. Following is a partial list of the configuration files in /etc/pam.d:

```
# ls /etc/pam.d
atd      authconfig    crond   halt     login     passwd
poweroff   reboot      sshd     su       sudo     system-config-date
```

Each PAM configuration file contains a group of directives that define the authentication module as well as any controls or arguments. The directives for configuration file entries are:

- module\_type
- control\_flag
- module\_name
- module\_arguments

## PAM Configuration Files

Each of these configuration files contains a list, or stack, of calls to authentication modules. For example, the following are the contents of the `su` configuration file, which lists four authentication modules:

```
# cat /etc/pam.d/su
auth sufficient pam_rootok.so
auth required pam_wheel.so use_uid
auth include system-auth
account sufficient pam_succeed_if.so uid = 0 use_uid quiet
account include system-auth
password include system-auth
session include system-auth
session optional pam_xauth.so
```

The authentication modules used for the `su` application are named `pam_rootok.so`, `pam_wheel.so`, `pam_succeed_if.so`, and `pam_xauth.so`.

The `pam_wheel.so` and `pam_succeed_if.so` modules include module arguments.

The `system-auth` file is not an authentication module but is a common configuration file for PAM-ified services. This file is located in the `/etc/pam.d` directory. Contents of the `system-auth` file are appended to the `su` configuration file and processed as if they were part of the file.

## PAM Authentication Modules

PAM provides several authentication modules in shared libraries, which are located in `/lib/security` (or `/lib64/security` for 64-bit Linux). The following is a partial list of the authentication modules:

```
# ls /lib64/security
pam_access.so      pam_cap.so        pam_chroot.so      pam_console.so
pam_ftp.so         pam_group.so     pam_krb5.so       pam_ldap.so
pam_limits.so      pam_mail.so      pam_selinux.so    pam_winbind.so
```

This pluggable, modular architecture provides flexibility in setting authentication policies for a system. With the authentication code separate from the application code, you can configure the authentication mechanism for a given application without ever touching the application.

## PAM Documentation

PAM has an extensive documentation set with much detail about both using PAM and writing modules to extend or integrate PAM with other applications. Almost all of the major authentication modules and configuration files with PAM have their own `man` pages. Also, the `/usr/share/doc/pam-<version>` directory contains the *PAM System Administrator's Guide* and a copy of the PAM standard, `rfc86.0.txt`.

## PAM Module Types

- The first column in the /etc/pam.d configuration file (auth in this example) is the module type:
  - auth sufficient pam\_rootok.so
- Module types represent a different aspect of the authorization process.
- Four types are available:
  - auth: Proves the user is authorized to use the service
  - account: Determines whether an already authenticated user is allowed to use the service
  - password: Updates user authentication credentials
  - session: Configures and manages user sessions



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The contents of the /etc/pam.d/reboot configuration file are listed here:

```
auth      sufficient   pam_rootok.so
auth      required     pam_console.so
account  required     pam_permit.so
```

The first column is the PAM module interface, or module type indicator. Four module types are available. Only the types `auth` and `account` determine authorization to run a command.

- **auth**: Proves that the user is authenticated, or authorized to use the service. This may be done by requesting and verifying the validity of a password. Modules with this interface can also set credentials, such as group memberships or Kerberos tickets.
- **account**: Verifies whether an already authenticated user is allowed access to the application. For example, it could check whether a user account has expired or whether a user is allowed to use this service at a particular time of day.
- **password**: Is used when a user tries to update his or her authentication token
- **session**: Configures and manages user sessions. Performs tasks that are needed to allow access, such as mounting a user's home directory and unmounting when the service is terminated.

## PAM Control Flags

- The second column in the /etc/pam.d configuration file (sufficient in this example) is the control flag:
  - auth sufficient pam\_rootok.so
- Each PAM module generates a success or failure result.
- Control flags tell PAM what to do with the result:
  - required: The module must pass before access is granted. The user is not notified immediately if the module fails.
  - requisite: This is similar to required except that the user is notified immediately if the module fails.
  - sufficient: Failure is not necessarily fatal, depending on other module test results.
  - optional: The module result is ignored unless only this is the only module.
  - include: This includes lines from another file.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The contents of the /etc/pam.d/reboot configuration file are listed as follows:

```
auth      sufficient    pam_rootok.so
auth      required      pam_console.so
account  required      pam_permit.so
```

The second column is the PAM control flag. PAM reads the stack from top to bottom and calls the modules listed in the configuration file. Each PAM module generates a success or failure result when called. Control flags tell PAM what to do with the result.

Modules can be stacked in a particular order, and the control flags determine how important the success or failure of a particular module is to the overall goal of authenticating the user to the service. The available control flags are listed:

- **required:** All required modules are tried and all must “pass” before access is granted by PAM. If the module fails at this point, the user is not notified until all modules in the stack have been executed.
- **requisite:** Similar to required, in which success of the module is required for authentication to continue. However, if the module fails, no further modules are executed. The user is notified immediately of the first failed required or requisite module test.

- **sufficient**: Success indicates that this module type has succeeded and no subsequent required modules of this type are executed. Failure is not fatal to the stack of this module type, however. PAM processes the remaining modules listed to decide whether access is allowed.
- **optional**: The module result is generally ignored. A module flagged as optional becomes necessary for successful authentication when it is the only module in the stack for a particular service.
- **include**: Unlike the other controls, this does not relate to how the module result is handled. This flag pulls in all lines in the configuration file that match the given parameter and appends them as an argument to the module.

PAM also includes some predefined actions in the control flag field, which are included within brackets as follows:

```
[value1=action1 value2=action2 ...]
```

The use of brackets in the control flag field gives you full control of PAM's actions. When the result returned by a function matches value, action is evaluated.

## PAM: Example #1

- The contents of the /etc/pam.d/reboot file:

```
#%PAM-1.0
auth      sufficient  pam_rootok.so
auth      required    pam_console.so
account  required    pam_permit.so
```

- Lines that begin with # are comments.
- Two lines have a module type of auth, meaning that the reboot application asks PAM to authenticate.
- The third line has a type of account, meaning that the reboot application asks PAM to check the account status.
- If the pam\_rootok.so module passes, authentication is allowed, because the control flag is sufficient.
- Otherwise, both required modules must pass.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

To see PAM in action, the following example steps through the processing of the reboot application's authentication stack. The contents of the /etc/pam.d/reboot configuration file are listed as follows:

```
# cat /etc/pam.d/reboot
#%PAM-1.0
auth      sufficient  pam_rootok.so
auth      required    pam_console.so
account  required    pam_permit.so
```

Lines that begin with # are comments and are not processed. The remaining lines tell PAM to do something as part of the authentication process. The first two lines have a module type of auth, meaning that the reboot application asks PAM to authenticate. The third line has a type of account, meaning that the reboot application asks PAM to check the account status. Processing of each line in the stack is described:

**auth sufficient pam\_rootok.so**

This line uses the pam\_rootok.so module to attempt to authenticate the user. The man page for pam\_rootok indicates that the module is used to gain only root access.

The name suggests what this module does. If user is `root`, then it is okay to allow access to the application. A more detailed description from man pages follows:

```
# man pam_rootok
```

`pam_rootok` is a PAM module that authenticates the user if the user's UID is 0. Applications that are created with `setuid-root` generally retain the UID of the user but run with the authority of an enhanced effective-UID. It is the real UID that is checked.

This test succeeds if the user's UID is 0. Because the control flag is `sufficient`, no other modules are consulted and the `reboot` command is executed. If this test fails, the next module is consulted.

```
auth      required      pam_console.so
```

This line uses the `pam_console.so` module to attempt to authenticate the user. Because the control flag is `required`, this module is required to succeed if the authentication stack is to succeed.

The man page for `pam_console` indicates that the module is used to determine the user owning the system console. If this user is already logged in at the console, `pam_console.so` checks whether there is a file in the `/etc/security/console.apps` directory with the same name as the service name (`reboot` in this case). If the file exists, authentication succeeds and control is passed to the next module.

```
account    required    pam_permit.so
```

This line uses the `pam_permit.so` module to check the account status of the user. However, the man page for this module reveals that `pam_permit.so` is a PAM module that always permits access. It does nothing else. This module is very dangerous. Use it with extreme caution. The man page does suggest that you add this line to your other login entries to disable account management, but continue to permit users to log in.

## PAM: Example #2

### Example # 2:

- Uses *value=action* pairs in the control flag field, allowing full control of PAM actions
  - [user\_unknown=ignore success=ok ignore=ignore default=bad]
- Uses authentication module arguments
  - pam\_unix.so nullok try\_first\_pass
  - pam\_succeed\_if.so uid >= 500 quiet
- Includes the contents of the common configuration file, system-auth
  - system-auth is included in nearly all individual service configuration files.
  - system-auth is auto-generated each time the authconfig command runs.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

To demonstrate a more complex example of PAM in action, the following is a partial listing of the login service's authentication stack:

```
# cat /etc/pam.d/login
#%PAM-1.0

auth [user_unknown=ignore success=ok ignore=ignore default=bad]
pam_securetty.so

auth      include    system-auth
account  required   pam_nologin.so
account  include    system-auth
password include    system-auth
```

Processing of each line in the stack is described:

```
auth [user_unknown=ignore success=ok ignore=ignore default=bad
pam_securetty.so
```

The first non-commented line calls the `pam_securetty.so` module, which allows root logins only if logging in on a secure TTY, as defined in the `/etc/securetty` file. This module has no effect on non-root users. This entry also contains specific actions for the control flag.

Several predefined control flag actions are available. This example uses the following actions and values:

- user\_unknown=ignore:
  - user\_unknown: The user is not known to the underlying authentication module.
  - ignore: The return status does not contribute to the return code.
- success=ok:
  - success: Successful function return
  - ok: If the module fails, the total stack state is fail. If the stack is already in fail status, the return code of this module does nothing.
- ignore=ignore:
  - ignore: Ignore underlying account modules regardless of whether the control flag is required, optional, or sufficient.
  - ignore: The return status does not contribute to the return code.
- default=bad:
  - default: All not explicitly mentioned values
  - bad: The return status is set to fail.

```
auth include system-auth
```

The next line includes the contents of a common configuration file, `system-auth`, into the `/etc/pam.d/login` file. The `system-auth` configuration file is included in nearly all individual service configuration files. It checks that the user who is logging in is authorized to do so, including verification of the username and password. The `system-auth` configuration file is auto-generated each time the `authconfig` command is executed. An example of `auth` entries in a `system-auth` file are:

```
auth required pam_env.so
auth sufficient pam_fprintd.so
auth sufficient pam_unix.so nullok try_first_pass
auth requisite pam_succeed_if.so uid >= 500 quiet
auth required pam_deny.so
```

The `pam_env.so` module allows the setting and unsetting of environmental variables. The `pam_fprintd.so` module is used for fingerprint authentication. The `pam_unix.so` module is the standard UNIX password authentication module. The `nullok` argument overrides the default action to not permit the user access to a service if the user's password is blank. The `try_first_pass` argument tries the previous stacked module's password before prompting the user for his or her password. The `pam_succeed_if.so` module tests account characteristics. In this case, it tests for `uid >= 500`. The `quiet` argument means do not log success or failure to the system log file. The `pam_deny.so` module is the locking-out PAM module and can be used to deny access.

```
account required pam_nologin.so
```

The next line uses the `pam_nologin.so` module. This module prevents users from logging in to the system when the `/etc/nologin.txt` file exists. This module has no affect on the `root` user.

The remaining entries also include the contents of the `system-auth` file for module types of account and password.

# Quiz

Which of the following are examples of PAM module types?

- a. requisite
- b. required
- c. auth
- d. account
- e. password
- f. sufficient
- g. session



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Summary

In this lesson, you should have learned how to:

- Describe the purpose of PAM
- Describe PAM configuration files
- Describe PAM authentication modules
- Describe PAM module types
- Describe PAM control flags
- Walk through PAM authentication examples

## Practice 16: Overview

The practices for this lesson cover the following:

- Configuring PAM for a single login session
- Configuring PAM to prevent non-root login

SELinux is referenced in the following practices:

- Practice 16-1: Configuring PAM for a Single Login Session
- Practice 17-2: Configuring a chroot Jail for ftp Users
- Practice 18-3: Preparing Disks for ASM Use



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# Introduction to SELinux

- Standard Linux security is based on DAC.
- SELinux provides finer grained control.
- SELinux runs in three modes:
  - Enforcing
  - Permissive
  - Disabled
- Display the SELinux mode with the `sestatus` or `getenforce` commands.
- SELinux also provides “Booleans.”



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

In some of the remaining practices, you are presented with a new topic in Oracle Linux, SELinux. SELinux stands for “Security-Enhanced Linux” and is covered in another course in the Oracle Linux curriculum map. Introductory information on SELinux is included here.

Standard Linux security is based on Discretionary Access Control (DAC). With DAC, access to files and devices are based solely on user identity and ownership. Each file can have read, write, and execute permissions for the owner of the file, for the group, and for other users.

SELinux was created by the US National Security Agency to provide a finer-grained level of control over files, processes, users, and applications in the system. It is an enhancement to the Linux kernel and it implements a different type of security called Mandatory Access Control (MAC). The MAC policy is centrally managed rather than being managed by a user.

SELinux runs in one of three modes:

- **Enforcing:** Access is denied to users and programs unless permitted by SELinux security policy rules.
- **Permissive:** The security policy rules are not enforced, but SELinux sends denial messages to a log file.
- **Disabled:** SELinux does not enforce a security policy because no policy is loaded in the kernel. Only DAC rules are used for access control.

You can use the `sestatus` command to display the SELinux mode, as well as some additional information about SELinux.

```
# sestatus  
SELinux status:      enabled  
...  
Current mode:        enforcing  
...
```

You can use the `getenforce` command to display the SELinux mode. This command displays the current mode: “Enforcing,” “Permissive,” or “Disabled”. Example:

```
# getenforce  
Enforcing
```

SELinux running in “Enforcing” mode is often the cause of a problem in Linux. In some of the practices in this course, you are directed to change the SELinux mode to “Permissive” to get a function of Linux to work properly. You can use the `setenforce` command to change the mode to either “Enforcing” (1) or “Permissive” (0). Example:

```
# setenforce 0  
# getenforce  
Permissive
```

SELinux also provides “Booleans,” which allow parts of an SELinux policy to be changed at run time, without reloading or recompiling an SELinux policy. You can display a list of Booleans, state information, and a description of the Boolean by running the following command:

```
# semanage boolean -l  
SELINUX boolean   State  Default Description  
ftp_home_dir     (off , off)    Allow ftp to read and write ...  
...
```

You can change the state of a specific Boolean to either `on` or `off` by using the `setsebool` command. For example, to turn the `ftp_home_dir` Boolean `on`:

```
# setsebool ftp_home_dir on
```

Use the `getsebool` command to display the state of a specific Boolean. Example:

```
# getsebool ftp_home_dir  
ftp_home_dir --> on
```

SELinux provides many other features and functions and configuration options. This brief introduction will help you to understand the tasks that you are directed to perform, and why, in the following practices in this course:

- Practice 16-1: Configuring PAM for a Single Login Session
- Practice 17-2: Configuring a chroot Jail for ftp Users
- Practice 18-3: Preparing Disks for ASM Use

# 17

## Security Administration

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe the chroot jail
- Use the chroot utility
- Describe iptables
- Use the Firewall Configuration Tool
- Describe iptables tables, chains, rules, and targets
- Use the iptables utility
- Describe TCP wrappers
- Configure TCP wrappers



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Several methods of securing your computer system are covered in this lesson.

## chroot Jail

- The `chroot` utility changes the apparent root directory.
- A program (process) runs with a root directory other than `/`.
- The artificial root directory is called a `chroot` jail.
- To the process, it appears that the directory it is running in is the root directory.
- A `chroot` jail limits the directory access of a potential attacker.
- A `chroot` jail is not intended to:
  - Defend against intentional tampering by privileged (root) users
  - Be used to block low-level access to system devices by privileged users
- The `chroot` jail directory must be populated with all files required by the process at their expected locations.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

As the name implies, a `chroot` operation changes the apparent root directory for a running process and its children. It allows you to run a program (process) with a root directory other than `/`. The program cannot see or access files outside the designated directory tree.

For example, you can run a program and specify its root directory as `/home/oracle/jail`. In this case, the program's root directory is actually `/home/oracle/jail`. The program would not be aware of, or able to access, any files above this directory in the hierarchy.

This artificial root directory is called a `chroot` jail. Its purpose is to limit the directory access of a potential attacker. The `chroot` jail locks down a given process and any user ID it is using so that the user sees only the directory that the process is running in. To the process, it appears that the directory it is running in is the root directory.

The `chroot` mechanism is not intended to defend against intentional tampering by privileged (root) users. It is also not intended by itself to be used to block low-level access to system devices by privileged users. A `chroot` root user can still create device nodes and mount the file systems on them.

For a `chroot` process to successfully start, the `chroot` directory must be populated with all required program files, configuration files, device nodes, and shared libraries at their expected locations.

## chroot Utility

- To use a chroot jail, use the following command:
  - # chroot *new\_root* [*command*]
- The *new\_root* directory becomes the artificial root.
- chroot changes to *new\_root* and runs the optional command.
  - Alternatively, it runs the SHELL variable if the command is omitted.
- The command fails unless the necessary files are copied into the *new\_root* directory before running chroot:

```
# chroot /home/oracle/jail  
chroot: failed to run command '/bin/bash': No  
such file or directory
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

To use a chroot jail, use the following command (*new\_root* must be an existing directory):

```
# chroot new_root [command]
```

The *new\_root* directory becomes the artificial root directory. chroot changes to *new\_root* and runs the optional command. Without specifying a command as an argument, chroot changes to *new\_root* and runs the value of the SHELL environment variable or /bin/sh if SHELL is not set.

For example, assuming SHELL is set to /bin/bash, and the /home/oracle/jail directory exists, running the chroot command results in the following:

```
# chroot /home/oracle/jail  
chroot: failed to run command '/bin/bash': No such file or  
directory
```

The /home/oracle/jail directory takes the name of /. chroot cannot find the /bin/bash within this chroot jail and returns the error message.

To implement a chroot jail, create the new root directory structure and copy all the necessary files into this new root directory before running the chroot command.

## Implementing a chroot Jail

- Make the necessary directories and copy all required files into these directories:
  - \$ mkdir /home/oracle/jail/bin
  - \$ cp /bin/bash /home/oracle/jail/bin
- Determine whether any shared libraries are required:
  - \$ ldd /bin/bash
- Create the lib (or lib64) directory and copy all required shared libraries into this directory:
  - \$ mkdir /home/oracle/jail/lib64
  - \$ cp /lib64/{...} /home/oracle/jail/lib64
- Execute the chroot command (as root):

```
# chroot /home/oracle/jail  
bash-4.1#
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

To implement a chroot jail and run /bin/bash, create the bin directory in the artificial root directory (/home/oracle/jail in this example), and copy /bin/bash into this directory:

```
$ mkdir /home/oracle/jail/bin  
$ cp /bin/bash /home/oracle/jail/bin
```

The /bin/bash command is dynamically linked to shared libraries. These libraries must also be copied into the chroot jail.

Use the ldd command to determine which libraries are required by the /bin/bash command:

```
$ ldd /bin/bash  
linux-vdso.so.1 => (0x00007fff83512000)  
libtinfo.so.5 => /lib64/libtinfo.so.5 (0x000000351cc00000)  
libdl.so.2 => /lib64/libdl.so.2 ...  
libc.so.6 => /lib64/libc.so.6 ...  
/lib64/ld-linux-x86-64.so.2
```

Copy each of these files into a lib64 directory in the artificial root directory.

Make the lib64 directory and copy the shared libraries into this directory:

```
$ mkdir /home/oracle/jail/lib64  
$ cp /lib64/{libtinfo.so.5,libdl.so.2,libc.so.6,ld-linux-x86-  
64.so.2} /home/oracle/jail/lib64
```

Now that all the required files are in their expected locations, running the chroot command (as root) results in the following:

```
# chroot /home/oracle/jail  
bash-4.1#
```

The command succeeded this time and the /bin/bash program executed. Entering pwd to print the current directory displays /, even though the actual directory is /home/oracle/jail:

```
bash-4.1# pwd  
/
```

The pwd command runs because it is a shell built-in command. Running any other command fails because bash cannot find the command. The process assumes it is in the root directory and has no visibility or knowledge of any files above this directory in the hierarchy.

For example, running the ls command fails:

```
bash-4.1# ls  
bash: ls: command not found
```

Use the exit command to exit the chroot jail.

```
bash-4.1# exit  
exit  
#
```

# Running Services in a chroot Jail

- DNS and FTP includes chroot jail options.
- DNS:
  - Install the bind-chroot package.
  - /var/named/chroot becomes the chroot for BIND files.
- FTP (vsftpd daemon):
  - Anonymous users are automatically placed in a chroot jail.
  - /var/ftp appears as /.
  - Local user home directories can be configured as chroot jails.
  - Set options in the /etc/vsftpd/vsftpd.conf file.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Two services are set up to take advantage of chroot jails. You can set up DNS so that named runs in a jail. The vsftpd FTP server can automatically start chroot jails for clients.

## DNS in chroot Jail

The bind-chroot package allows you to set up named to run in a chroot jail. When you install this package, the /var/named/chroot directory is created and becomes the chroot jail directory for all BIND files.

- The /var/named directory becomes /var/named/chroot/var/named.
- /etc/named\* files become /var/named/chroot/etc/named\* files.

Installing this package also sets the ROOTDIR shell variable to /var/named/chroot in the /etc/sysconfig/named file.

The advantage of running named in a chroot jail is that if a hacker enters your system via a BIND exploit, the hacker's access to the rest of your system is isolated to the files under the chroot jail directory.

## FTP Clients in chroot Jail

By default, anonymous users are placed in a chroot jail. When an anonymous user logs in to a vsftpd server, the user's home directory is /var/ftp. However, all that the user sees is /.

For example, a directory named `/var/ftp/upload` appears as `/upload` to an anonymous user. This prohibits anonymous users from being able to access any files above `/var/ftp` in the directory hierarchy.

Local users that access a `vsftpd` server are placed in their home directory. You can enable options in the `/etc/vsftpd/vsftpd.conf` file to put local users in a `chroot` jail, where the artificial root directory is the user's home directory. The following options exist in the `vsftpd` configuration file to implement a `chroot` jail for local users:

- `chroot_list_enable`
- `chroot_local_user`
- `chroot_list_file`

When a local user logs in to the `vsftpd` server, the `chroot_list_enable` directive is checked. If this directive is set to YES, the service checks the `/etc/vsftpd/chroot_list` file (by default) or another file specified by the `chroot_list_file` directive.

Another directive is then checked, `chroot_local_user`. If this directive is set to YES, then the `chroot_list` becomes a list of users to NOT `chroot`. If this directive is set to NO, the user is put into a `chroot` jail in his home directory.

## Introduction to `iptables`

- Packet filtering firewalls accept or deny network packets.
- The Linux kernel has built-in packet filtering functionality.
- `iptables` is used for IPv4 traffic.
  - `ip6tables` is used for IPv6 traffic.
- Two components:
  - `netfilter`: A kernel component that stores filtering rules
  - `iptables`: A user utility to maintain rules stored by `netfilter`
- Firewall Configuration Tool (GUI):
  - This tool creates basic `iptables` rules.
- Use the `iptables` command-line utility to create complex firewall configuration rules.
- Rules are stored in `/etc/sysconfig/iptables`.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

A packet filtering firewall reads incoming network packets and filters (allows or denies) each data packet based on the header information in the packet. You can create packet filters, or rules, that determine which packets are accepted and which are rejected. For example, you can create a rule to block a port. If a request is made to the port that is blocked by the firewall, the request is ignored. If a service is listening on a blocked port, it does not receive the packets and is effectively disabled.

The Linux kernel has built-in packet filtering functionality. Kernel versions earlier than 2.4 relied on `ipchains` for packet filtering. The 2.4 kernel (and above) includes `iptables`, which is similar to `ipchains`, but provides advanced features and more control of filtering network packets. `iptables` pertains to IPv4 packets. `ip6tables` is available for IPv6 packets.

The `iptables` mechanism is composed of two components:

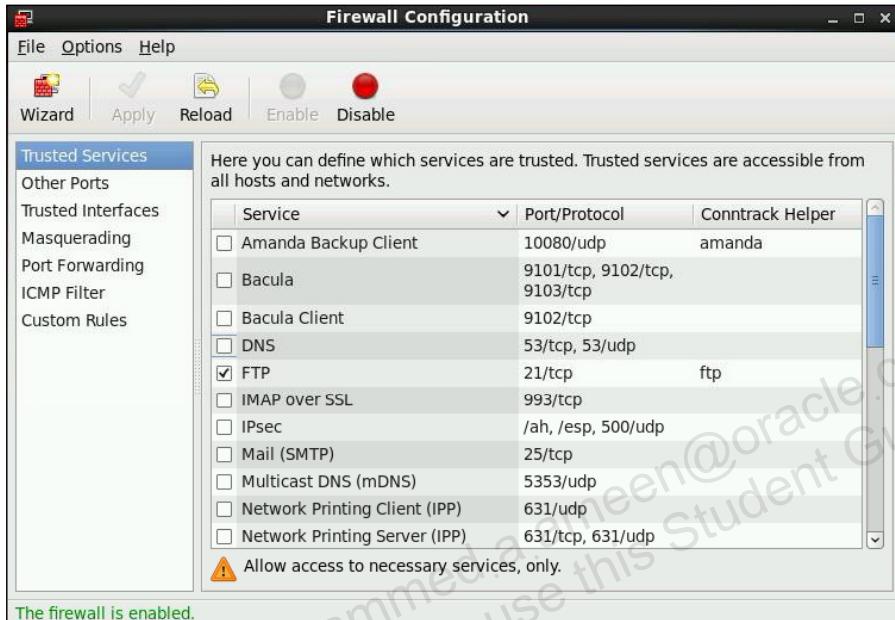
- `netfilter`: A kernel component consisting of a set of tables that store rules that the kernel uses to control network packet filtering
- `iptables`: A user utility to create, maintain, and display the rules stored by `netfilter`

Oracle Linux provides a GUI called the Firewall Configuration Tool for configuring a simple firewall. This tool creates basic `iptables` rules for implementing a general-purpose firewall.

To create a more complex firewall configuration, you can manually configure packet filtering rules by using the `iptables` utility. Rules are stored in the `/etc/sysconfig/iptables` file.

# Firewall Configuration Tool

system-config-firewall



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Firewall Configuration Tool allows you to configure a basic firewall. You can define which network services on your system remote users can access. For more complex rules, use the `iptables` utility.

Use the following command to launch the tool:

```
# system-config-firewall
```

If the command is not found, install the `system-config-firewall` package:

```
# yum install system-config-firewall
```

From the GUI, you can enable or disable the firewall:

- **Disable:** Provides complete access, does no security checking.
- **Enable:** Configures the system to allow outgoing packets and to reject incoming connections that are not in response to outbound requests, such as DNS replies or DHCP requests.

It is recommended that you configure a firewall for any system with an Internet connection. Only allow access to specific services running on your server as needed. Enable options in the Trusted Services list in the GUI to allow service requests to pass through the firewall. For example, if your system is configured as an FTP server (running `vsftpd`), select the FTP check box as shown. Click Apply to save any changes.

## iptables Terminology

- The netfilter component is a set of tables:
  - Filter: The default table
  - NAT: The Network Address Translation table
  - Mangle: The table used to alter certain fields in a packet
- Tables store rules, which consist of:
  - One or more match criteria
  - A single action, or target, such as ACCEPT, DROP, REJECT
- Rules are stored in chains. Filter table chains are:
  - INPUT: Inbound packets pass through this chain.
  - OUTPUT: Outbound packets pass through this chain.
  - FORWARD: Packets not addressed to the local system pass through this chain.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The netfilter component of iptables is a set of tables. The three main tables are described as follows:

- **Filter**: The default table. This table is used primarily to DROP or ACCEPT packets based on their content.
- **NAT**: The Network Address Translation table. Packets that create new connections are routed through this table.
- **Mangle**: This table is used to alter certain fields in a packet.

These tables store rules that the kernel uses to make network packet filtering decisions. A rule consists of one or more criteria and a single action, or target. If the criteria in a rule match the information in a network packet header, the action or target is applied to the packet.

Examples of targets include:

- **ACCEPT**: Continue processing the packet.
- **DROP**: End the packet's life without notice.
- **REJECT**: Similar to DROP, except it notifies the sending system that the packet was blocked. Use DROP if you do not want the sender to be notified.

Rules are stored in chains. Each rule in a chain is applied, in order, to a packet until a match is found. If there is no match, the chain's policy, or default action, is applied to the packet.

Each netfilter table has several built-in chains. The default netfilter table, named `filter`, contains the following built-in chains:

- **INPUT:** Inbound packets to the local system pass through this chain.
- **OUTPUT:** Packets created locally pass through this chain.
- **FORWARD:** Packets not addressed to the local system pass through this chain.

These chains are permanent and cannot be deleted. You can create additional, user-defined chains in this `filter` table.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Beginning iptables Maintenance

- To start the service:
  - # service iptables start
- To configure iptables to start at boot time:
  - # chkconfig iptables on
- To list iptables:
  - # iptables -L [*chain*]
- Each chain has a default policy:
  - The action to take (ACCEPT or DROP) if no rules match
- To set default policy:
  - # iptables -P *chain* DROP|ACCEPT
- To save configuration changes to iptables:
  - # service iptables save



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The firewall rules are active only if the `iptables` service is running. Start the service as follows. After changing the configuration, save the configuration and restart the service:

```
# service iptables start
```

To ensure that `iptables` starts at boot time, enter the following command to enable the service for run levels 2, 3, 4, and 5:

```
# chkconfig iptables on
```

You can turn off the `ip6tables` service if you are not using IPv6.

Use the `iptables` command to create, maintain, and display the rules stored by netfilter. Several options exist for the command. Long or short options are allowed. For example, to add rules to the end of a chain, use either of the following:

```
# iptables --append ...
# iptables -A ...
```

To remove rules from a chain, use either of the following:

```
# iptables --delete ...
# iptables -D ...
```

Use `iptables -h` or `iptables --help` to display all options.

Use the `-L` option, or `--list`, to list the current rules:

```
# iptables -L
Chain INPUT (policy ACCEPT)
target  prot opt source      destination
ACCEPT  all  --  anywhere   anywhere    state RELATED, ESTABLISHED
ACCEPT  icmp --  anywhere   anywhere
ACCEPT  all  --  anywhere   anywhere
ACCEPT  tcp   --  anywhere   anywhere    state NEW tcp dpt:ftp
ACCEPT  tcp   --  anywhere   anywhere    state NEW tcp dpt:ssh
REJECT  all  --  anywhere   anywhere   reject-with icmp-host-...
Chain FORWARD (policy ACCEPT)
target  prot opt source      destination
REJECT  all  --  anywhere   anywhere   reject-with icmp-host-...
Chain OUTPUT (policy ACCEPT)
target  prot opt source      destination
```

The rules in all three chains (INPUT, FORWARD, OUTPUT) of the default table, `filter`, are displayed. Include the chain as an argument to limit output to a specific chain. For example, to list the rules in the INPUT chain only:

```
# iptables -L INPUT
```

## Policies

Each `iptables` chain consists of a default policy and zero or more rules, which together define the overall ruleset for the firewall. If the information in a network packet header does not match any rule, the chain's policy, or default action, is applied to the packet. In this example, the policy for each chain is `ACCEPT`.

The default policy for a chain can be either `DROP` or `ACCEPT`. A more secure system would have a default of `DROP` and would allow only specific packets on a case-by-case basis. Set the default policy as follows, providing either the `DROP` or `ACCEPT` argument. This example blocks all incoming and outgoing network packets:

```
# iptables -P INPUT DROP
# iptables -P OUTPUT DROP
```

The FORWARD chain routes network traffic to its destination node. To create a `DROP` policy for these packets and to restrict internal clients from inadvertent exposure to the Internet, use the following rule:

```
# iptables -P FORWARD DROP
```

After establishing the default policies for each chain, create and save additional rules to meet your particular network and security requirements.

To save the rules so that they are loaded when the `iptables` service is started, use the following command:

```
# service iptables save
```

## Adding a Rule by Using the `iptables` Utility

- To add a rule to a chain, use the following syntax:
  - `iptables [-t <table>] -A <chain> <rule_specs> -j <target>`
- Command-line options and arguments:
  - `-t <table>`: Defaults to the `Filter` table if omitted
  - `-A <chain>`: Appends a rule to `<chain>`
  - `rule_specs`: Specifies the rule criteria
  - `-j <target>`: Specifies the action to take if a match occurs
- Example:
  - `# iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT`
  - Accept incoming packets if protocol is TCP and destination port is 80 (http).



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

To add a rule to a chain, use the following syntax:

```
iptables [-t <table>] -A <chain> <rule_specs> -j <target>
```

The command-line options and arguments are described as follows:

- `-t <table>` option: Specifies the table (`filter`, `nat`, `mangle`). If omitted, the `filter` table is used by default.
- `-A <chain>` option: Appends a rule to `<chain>`. The chain value depends on the table. If the table is `filter`, the possible chains are `INPUT`, `OUTPUT`, and `FORWARD`.
- `rule_specs`: Specifies the rule criteria, or how to match a network packet.
- `-j <target>` option: Specifies the target of the rule, or what action to take if the packet matches the rule. The target value depends on the table. If the table is `filter`, the possible targets are `ACCEPT`, `DROP`, and `REJECT`.

The following example allows access to TCP port 80 on the firewall:

```
# iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

Because no table is defined, the rule is written to the `filter` table. The chain is `INPUT`, so the rule is applied to incoming packets. The `rule_specs` consists of `-p tcp -m tcp --dport 80`. If information in the packet header matches the rule, the action taken is `ACCEPT`.

The rule\_specs in this example are defined as follows:

- **-p tcp**: Matches if the packet uses the TCP protocol. Protocol can also use the long option, `--protocol`. The specified protocol can be any protocol name or number listed in the `/etc/protocols` file. When omitted, the default is all.
- **-m tcp**: The `-m` option specifies match extensions. Match extensions are loaded implicitly when `-p` or `--protocol` is specified, or explicitly using the `-m` or `--match` option followed by the matching module name. Various extra command-line options become available, depending on the specific module. The module name in this example is `tcp`. Use the `-h` or `--help` option after the module has been specified to receive help specific to that module. For example (the optional exclamation point [!] matches packets that do not match the criterion):

```
# iptables -p tcp -h
...
tcp match options:
[!] --tcp-flags mask comp      match when TCP flags & mask == comp
                                (Flags: SYN ACK FIN RST URG PSH ALL...)
[!] --syn                      match when only SYN flag set
                                (equivalent to --tcp-flags SYN, RST...)
[!] --source-port port[:port]
    --sport ...                 match source port(s)
[!] --destination-port port[:port]
    --dport ...                  match destination port(s)
[!] --tcp-option number        match if TCP option set
• --dport 80: Matches if the destination port is 80
```

Save any changes so that they are loaded when the `iptables` service is started, using the following command:

```
# service iptables save
```

The new entry appears in the `/etc/sysconfig/iptables` file:

```
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

The `iptables -L` output displays the new entry as follows:

```
ACCEPT  tcp  --  anywhere  anywhere  tcp dpt:http
```

The TCP destination port of 80 is represented as `http` in the output because the `http` daemon listens for client requests on port 80.

## iptables Rule Specs

- **-p, --protocol protocol**: Matches if the packet uses *protocol*
- **-s, --source address [/mask]**: Matches if the packet came from *address*
- **-d, --destination address [/mask]**: Matches if the packet is going to *address*
- **-j, --jump target**: Specifies what to do if the packet matches the rule specification
- **-i, --in-interface name**: Matches if the packet came from interface *name*
- **-o, --out-interface name**: Matches if the packet is to be sent to interface *name*



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The **-p** (or **--protocol**) rule specification as shown in the previous example is commonly used as match criterion. The following describes some additional rule specifications for matching a network packet. Each of the rule specs can be preceded with an exclamation point (!) to have the inverse effect—that is, to match packets that do not match the criterion:

- **-p, --protocol protocol**: Matches if the packet uses *protocol*
- **-s, --source address [/mask]**: Matches if the packet came from *address*. The *address* can be a name or IP address and can include the optional *mask* with an IP address. The **--src** option is an alias and can also be used.
- **-d, --destination address [/mask]**: Matches if the packet is going to *address*. The *address* can be specified as described in the **--source** option. The **--dst** option can also be used.
- **-j, --jump target**: This specifies what to do if the packet matches the rule specification.
- **-g, --goto chain**: This specifies that the processing continues in a user-specified chain.
- **-i, --in-interface name**: Matches if the packet came from interface *name*
- **-o, --out-interface name**: Matches if the packet is to be sent to interface *name*

## More iptables Options

- **-D, --delete *chain rule\_spec/rule\_number*:** Removes a rule from *chain*
- **-I, --insert *chain rule\_spec/rule\_number*:** Inserts a rule in *chain* above an existing rule
- **-R, --replace *chain rule\_spec/rule\_number*:** Replaces an existing rule in *chain*
- **-F, --flush [*chain*]:** Deletes rules in *chain*
- **-N, --new-chain *chain*:** Creates a user-defined *chain*
- **-X, --delete-chain *chain*:** Deletes a user-defined *chain*



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Three iptables options have been discussed; the **-A** (or **--append**) option to add a rule to the end of a chain, the **-L** (or **--list**) option to list all rules, and the **-P** (or **--policy**) option to set the default policy. The following describes some of the other options available with the **iptables** command:

- **-D, --delete *chain rule\_spec/rule\_number*:** Removes a rule from *chain*. Define the rule to be removed by the *rule\_spec* or the *rule\_number*. To display rule numbers, use the following command:  

```
# iptables -L --line-numbers
```
- **-I, --insert *chain rule\_spec/rule\_number*:** Inserts a rule in *chain* above an existing rule that is specified by the *rule\_spec* or the *rule\_number*. If no existing rule is specified, the rule is inserted at the beginning of the chain.
- **-R, --replace *chain rule\_spec/rule\_number*:** Replaces an existing rule in *chain*
- **-F, --flush [*chain*]:** Deletes rules in *chain*. If you omit the *chain* argument, all rules in all chains are deleted.
- **-N, --new-chain *chain*:** Creates a new user-defined *chain*
- **-X, --delete-chain *chain*:** Deletes a user-defined *chain*

# NAT Table

- The netfilter kernel subsystem provides a nat table in addition to the default filter table to facilitate NAT.
- Use the following option to specify the nat table:
  - # iptables -t nat ...
- Built-in chains for the nat table:
  - PREROUTING: Alters packets when they arrive
  - OUTPUT: Alters locally generated packets before they are sent out
  - POSTROUTING: Alters packets before they are sent out
- Targets for the nat table:
  - DNAT: Alters the destination IP address
  - SNAT: Alters the source IP address
  - MASQUERADE: Facilitates use with DHCP



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Network Address Translation (NAT) is the process where a network device, usually a firewall, assigns a public address to a computer (or group of computers) inside a private network. NAT places private IP subnetworks behind one or a small pool of public IP addresses, masquerading all requests to one source rather than several. The main use of NAT is to limit the number of public IP addresses an organization or company must use, for both economy and security purposes.

The netfilter kernel subsystem provides a nat table in addition to the default filter table to facilitate NAT. The nat table is consulted when a packet that creates a new connection is encountered. Use the `iptables -t <table>` option to specify the nat table when adding, deleting, replacing, or displaying rules:

```
# iptables -t nat ...
```

Whereas the built-in chains for the filter table are INPUT, OUTPUT, and FORWARD, the following built-in chains exist for the nat table:

- PREROUTING: Alters packets, such as destination address, when they arrive
- OUTPUT: Alters locally generated packets before they are sent out
- POSTROUTING: Alters packets before they are sent out

The targets for the filter table are DROP, ACCEPT, and REJECT. The nat table has specific targets as well:

- **DNAT**: Alters the destination IP address on an inbound packet so that it is routed to another host
- **SNAT**: Alters the source IP address on an outbound packet so that it appears to come from a fixed IP address, such as a firewall or router
- **MASQUERADE**: Differs from SNAT in that it checks for an IP address to apply to each outbound packet, making it suitable for use with DHCP

The following example specifies that the nat table use the built-in PREROUTING chain to forward incoming HTTP requests to a dedicated HTTP server at 172.31.0.23. The rule changes the destination address of the packet.

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT  
--to 172.31.0.23:80
```

The following example allows LAN nodes with private IP addresses to communicate with external public networks:

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

This rule masks requests from LAN nodes with the IP address of the firewall's external device (in this case, eth0). POSTROUTING allows packets to be altered as they are leaving the firewall's external device. The -j MASQUERADE target masks the private IP address of a node with the external IP address of the firewall/gateway.

## TCP Wrappers

- A TCP wrapper provides basic traffic filtering of incoming network traffic.
- Specifically, a TCP wrapper provides or denies access to “wrapped” network services.
- Use `ldd` command to determine whether a network service is wrapped (linked to `libwrap.a`):

```
# ldd /usr/sbin/sshd | grep libwrap
libwrap.so.0 => /lib64/libwrap.so.0 ...
```
- TCP wrappers rely on two configuration files as the basis for access control:
  - `/etc/hosts.allow`
  - `/etc/hosts.deny`
- These files determine whether client access to network service is allowed or denied.

**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

TCP wrappers provide basic traffic filtering of incoming network traffic. Access to “wrapped” network services running on a Linux server from other systems can be allowed or denied. A TCP wrapped service is one that has been compiled against the `libwrap.a` library. Use the `ldd` command to determine whether a network service is linked to `libwrap.a`. The following example determines the absolute path name of the `sshd` service, and then lists the shared libraries linked to the `sshd` service, using the `grep` command to search for the `libwrap` library:

```
# which sshd
/usr/sbin/sshd
# ldd /usr/sbin/sshd | grep libwrap
libwrap.so.0 => /lib64/libwrap.so.0 (0x00007f769e067000)
```

TCP wrappers rely on two configuration files as the basis for access control:

- `/etc/hosts.allow`
- `/etc/hosts.deny`

When a client attempts to connect to a network service on a remote system, these files are used to determine whether client access is allowed or denied.

# TCP Wrappers Configuration

- Configuration files:
  - /etc/hosts.allow: Defines rules that allow client access to server daemons
  - /etc/hosts.deny: Defines rules that deny client access to server daemons
- The format for entries is the same for both files:
  - daemon\_list : client\_list [: command]
  - vsftpd : 192.168.2.\*
- The /etc/hosts.allow file is read first:
  - If the daemon-client pair matches, access is granted.
  - The entry in /etc/hosts.deny is ignored if the entry in /etc/hosts.allow grants access.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Use /etc/hosts.allow and /etc/hosts.deny to define rules that selectively allow or deny clients access to server daemons on local system. The format for entries is as follows for both files:

daemon\_list : client\_list [: command]

A description of each field follows:

- **daemon\_list**: A comma-separated list of daemons, or keyword ALL for all daemons
- **client\_list**: A comma-separated list of clients, or keyword ALL for all clients
- **command**: An optional command that is executed when a client tries to access a server daemon

To allow client access, add the client host name or IP address in /etc/hosts.allow. To deny client access, add its name or IP address in /etc/hosts.deny.

The /etc/hosts.allow file is read first and is read from top to bottom. If daemon-client pair matches the first line in the file, access is granted. If the line is not a match, the next line is read and the same check is performed. If all lines are read and no match occurs, the /etc/hosts.deny file is read, starting at the top. If a daemon-client pair match is found in the deny file, access is denied. If no rules for the daemon-client pair are found in either file, or if neither file exists, access to the service is granted.

Because access rules in `hosts.allow` are applied first, they take precedence over rules specified in `hosts.deny`. Therefore, if access to a service is allowed in `hosts.allow`, a rule denying access to that same service in `hosts.deny` is ignored.

The following are some examples of entries in the `/etc/hosts.allow` file:

To allow clients on the 192.168.2 subnet to access FTP (daemon is `vsftpd`):

```
vsftpd : 192.168.2.*
```

To allow all clients to access ssh, scp, and sftp (daemon is `sshd`):

```
sshd : ALL
```

Place the following entry in the `/etc/hosts.deny` file to deny FTP service to all clients except subnet 192.168.2.\* (this assumes the previous entry of `vsftpd:192.168.2.*` exists in `/etc/hosts.allow`):

```
vsftpd : ALL
```

Use the `.domain` syntax to represent any hosts from a given domain. The following example allows connections to `vsftpd` from any host in the `example.com` domain (if the entry is in `/etc/hosts.allow`):

```
vsftpd : .example.com
```

If this entry appears in `/etc/hosts.deny`, the connection is denied.

## TCP Wrapper Command Options

- Use the optional *command* argument to send connection banners, warn of attacks, and enhance logging.
- To display the contents of a banner file:
  - vsftpd : ALL : banners /etc/banners/
- To append an entry to a log file:
  - ALL : 200.182.68.0 : spawn /bin/echo `date` %c %d >> /var/log/intruder\_alert
- To elevate the logging level:
  - sshd : ALL : severity emerg
- To deny access from /etc/hosts.allow:
  - sshd : .example.com : deny

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

TCP wrappers are capable of more than allowing and denying access to services. With the optional *command* argument, they can send connection banners, warn of attacks from particular hosts, and enhance logging.

To implement a TCP wrapper banner for a service, use the *banner* option. This example implements a banner for `vsftpd`. You need to create a banner file anywhere on the system, giving it the same name as the daemon. In this example, the file is called `/etc/banners/vsftpd` and contains the following lines:

```
220-Hello, %c
220-All activity on ftp.example.com is logged.
220-Inappropriate use results in access privileges being
removed.
```

The `%c` token supplies a variety of client information. The `%d` token (not shown) expands to the name of the daemon that the client attempted to connect to. For this banner to be displayed to incoming connections, add the following line to the `/etc/hosts.allow` file:

```
vsftpd : ALL : banners /etc/banners/
```

TCP wrappers can warn you of potential attacks from a host or network by using the `spawn` directive. The `spawn` directive executes any shell command. In this example, access is being attempted from the 200.182.68.0/24 network. Place the following line in the `/etc/hosts.deny` file to deny any connection attempts from that network, and to log the attempts to a special file:

```
ALL : 200.182.68.0 : spawn /bin/echo `date` %c %d >>
/var/log/intruder_alert
```

To allow the connection and log it, place the `spawn` directive in the `/etc/hosts.allow` file.

The following entry in `/etc/hosts.deny` denies all client access to all services (unless specifically permitted in `/etc/hosts.allow`) and logs the connection attempt:

```
ALL : ALL : spawn /bin/echo "%c tried to connect to %d and was
blocked" >> /var/log/tcpwrappers.log
```

The log level can be elevated by using the `severity` option. Assume that anyone attempting to ssh to an FTP server is an intruder. To denote this, place an `emerg` flag in the log files instead of the default flag, `info`, and deny the connection. To do this, place the following line in `/etc/hosts.deny`:

```
sshd : ALL : severity emerg
```

This uses the default `authpriv` logging facility, but elevates the priority from the default value of `info` to `emerg`, which posts log messages directly to the console.

The following example states that if a connection to the SSH daemon (`sshd`) is attempted from a host in the `example.com` domain, execute the `echo` command to append the attempt to a special log file, and deny the connection. Because the optional `deny` directive is used, this line denies access even if it appears in the `/etc/hosts.allow` file:

```
sshd : .example.com \
: spawn /bin/echo `/bin/date` access denied >> /var/log/sshd.log \
: deny
```

Each option field (`spawn` and `deny`) is preceded by the backslash (\) to prevent failure of the rule due to length.

Refer to the man page on `hosts_options` for additional information and examples.

## Quiz

Which of the following has a specific purpose of allowing or denying access to network services?

- a. chroot jail
- b. iptables
- c. TCP wrappers

## Summary

In this lesson, you should have learned how to:

- Describe the chroot jail
- Use the chroot utility
- Describe iptables
- Use the Firewall Configuration Tool
- Describe iptables tables, chains, rules, and targets
- Use the iptables utility
- Describe TCP wrappers
- Configure TCP wrappers

## Practice 17: Overview

The practices for this lesson cover the following:

- Configuring a chroot jail
- Configuring a chroot jail for ftp users
- Configuring iptables
- Configuring a TCP wrapper

# 18

## Oracle on Oracle

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Prepare your Oracle Linux server for Oracle Database installation
- Create Oracle software user and group accounts
- Set kernel parameters for Oracle Database
- Set Oracle database shell limits
- Configure HugePages
- Configure Oracle Database Smart Flash Cache (DBSFC)
- Use Oracle pre-install RPM
- Install and use ASMLib

# Oracle Software User Accounts

- Oracle database software owner:
  - Is commonly named `oracle`
  - Runs the OUI and has full privileges to install, uninstall, and patch Oracle software
  - Cannot be `root`
- The owner of the `httpd` process is:
  - A low-privileged OS user
  - Usually provided by the `nobody` user
- Database operations require a few more users:
  - Members of `OSOPER` group can start, stop, back up, and recover the database.
  - Members of the `OSDBA` group have `OSOPER` privileges, can create and drop database, and create other `OSDBA` members.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Oracle software installation requires a Linux user to be a designated Oracle software owner. The Oracle software owner runs the OUI (Oracle Universal Installer) to install Oracle Database and has full privileges to install, uninstall, and patch the Oracle software. The OUI cannot be run as the `root` user. The name of the Oracle software owner is commonly `oracle`, but you can use a different name.

The Oracle software installation also requires a low-privileged OS user to be the owner of the `httpd` process. This is usually provided by the `nobody` user.

Database operations require a few more users. A user who is a member of the `OSOPER` group can start, stop, back up, and recover the database. A user who is a member of the `OSDBA` group can create, drop database, and create other DBA privileged users, in addition to the privileges of the `OSOPER`.

Ordinary database users can have OS accounts on the database server, but it is not necessary. It is common for database users to connect to the database through a client or application server without any OS account. OS user accounts may be required by the database application for batch jobs or specialized external processes. The Oracle default installation does not require any ordinary database user to have OS accounts.

With Grid Infrastructure & ASM there is now a new user called `grid` and three new groups: `asmadmin`, `asmdba`, and `asmoper`.

# Oracle Software Group Accounts

- OSDBA:
  - This is commonly named dba.
  - Members of the OSDBA group have database administration privileges (SYSDBA).
- OSOPER:
  - This is commonly named oper.
  - Members of the OSOPER group have limited database administration privileges (SYSOPER).
- Oracle Inventory group:
  - This is commonly named oinstall.
  - All installed Oracle software is registered in this inventory.
  - Oracle software owner (oracle) is a member of this group.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Oracle Database Installation Guide names three group identifiers:

- **OSDBA (dba):** Identifies OS accounts that have database administration privileges (SYSDBA)
- **OSOPER (oper):** Identifies OS accounts that have limited database administration privileges (SYSOPER)
- **Oracle Inventory group (oinstall):** Identifies the owner of the Oracle software

An OSDBA group is the only group that must be created to manage the database files. By default, this group is dba, but can have a different group name. SYSDBA is a high-level administrative privilege much like that of the root user on Linux. The members of the OSDBA group own the database files and have the privilege to connect to the database without a password, using AS SYSDBA through OS authentication.

The OSOPER group members connect to the database using the AS SYSOPER mechanism. This group has a restricted set of privileges. Each database can have its own OSDBA and OSOPER groups.

During installation, one inventory is created per system and all Oracle software installed on a server is registered in this inventory. The inventory group name is oinstall, and the Oracle software owner (oracle) is a member of this group. This user is also a member of the OSDBA and OSOPER groups.

# System Resource Tuning

- An Oracle database instance requires certain system resources.
- Shared memory must be adjusted for database use.
- Shared memory system uses semaphores, which must be adjusted.
- Each dedicated server process requires a network port.
- Larger network buffers are recommended.
- The maximum number of open files per process must be increased.
- Shell limit settings must be increased.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Oracle Database instance requires certain system resources. Kernel resources are controlled by kernel parameters. Shell limits are controlled by the settings in the shell configuration files.

Oracle uses SYSV UNIX shared memory. The kernel parameters for shared memory must be adjusted for database use. The shared memory system also uses semaphores to coordinate shared memory access. Every Oracle instance requires a set of semaphores.

The Oracle instance communicates via network connections. Each dedicated server process requires a network port. In a shared server environment, each dispatcher requires a port.

Oracle recommends that you change the network buffers to allow larger defaults for send and receive buffers and a larger maximum buffer size. These changes are helpful to optimize network performance when there are high-bandwidth applications, such as RAC and GigE network interfaces.

Because an Oracle database often has a large number of open files, the kernel default setting for the maximum number of open files per process is too small.

Shell limit settings are typically used to prevent any one user from consuming so many resources that it prevents other users from being able to work. The typical user settings are too low for the Oracle software owner. The `oracle` user may have hundreds of processes executing and thousands of files open.

# Linux Shared Memory

- Three shared memory-related kernel parameters:
  - SHMMNI: The maximum number of systemwide shared memory segments
  - SHMMAX: The maximum size of each segment
  - SHMALL: The maximum number of shared memory pages system wide
- For Oracle database, set SHMMAX  $\geq$  the largest SGA.
- Shared memory kernel parameters are set in:  
`/etc/sysctl.conf`
- Parameters are viewable in:  
`# ls /proc/sys/kernel/sh*`  
`shmall shmmmax shmmni`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The following are the memory-related kernel parameters:

- SHMMNI: The maximum number of systemwide shared memory segments
- SHMMAX: The maximum size of each segment
- SHMALL: The maximum number of shared memory pages system wide

Shared memory is allocated in segments. A segment is not necessarily as large as the maximum size; it is only as big as is allocated. If a process needs a larger shared memory area than can be allocated in one segment, it allocates multiple segments. Database instances often allocate multiple segments to accommodate a large SGA (System Global Area).

For Oracle Database, the SHMMAX parameter limits the size of each of the shared memory segments on the system. It should be equal to or larger than the largest SGA on the system; otherwise the SGA is made up of multiple memory segments.

The memory-related kernel parameters are set in the `/etc/sysctl.conf` file:

- `kernel.shmmni = 4096`
- `kernel.shmmmax = 4398046511104`
- `kernel.shmall = 4294967296`

# Semaphores

- Semaphores are a method of controlling access to critical resources.
- The Oracle instance uses semaphores to control access to shared memory.
- Semaphores are allocated based on the PROCESSES initialization parameter.
- All four semaphore parameters are set by a single `kernel.sem` parameter in `/etc/sysctl.conf`:
  - `semmsl`: Maximum number of semaphores per set
  - `semnns`: Total number of semaphores in the system
  - `semopm`: Maximum number of operations per `semop` call
  - `semnni`: Maximum number of semaphore sets



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Semaphores are a robust method of controlling access to critical resources. The Oracle instance uses semaphores primarily to control access to shared memory. Semaphores are allocated based on the PROCESSES initialization parameter. The PROCESSES initialization parameter determines the maximum number of operating system processes that can be connected to Oracle Database concurrently.

Each Oracle instance tries to allocate two semaphore sets at startup. Immediately after startup, the instance releases one set of semaphores. This method prevents exhaustion of the semaphore resources. Each set allocates at least as many semaphores as the value of PROCESSES. If it does not, the Oracle instance gets more sets to satisfy the number of semaphores that it needs. If the instance cannot allocate enough semaphores (either in one set or in multiple sets), the instance does not start.

You can adjust the kernel parameters for semaphores. Semaphore settings are positional. All four of the semaphore parameters are set by a single kernel parameter, `kernel.sem`, in `/etc/sysctl.conf` and viewable in `/proc/sys/kernel/sem`. The four parameters are:

- `semmsl`: Maximum number of semaphores per set
- `semnns`: Total number of semaphores in the system
- `semopm`: Maximum number of operations per `semop` call
- `semnni`: Maximum number of semaphore sets

A `semop` call is a call to a function that actually uses the semaphores (for example, testing, setting, and clearing).

The following are the minimum required values. System administrators and DBAs might need to tune these values higher for production workloads, as per the documentation.

- For `semmsl`: 250 or the largest `PROCESSES` parameter of an Oracle database plus 50
- For `semnms`: 32000 or sum of the `PROCESSES` parameters for each Oracle database, adding the largest one twice, and adding an additional 25 to 50 for each database
- For `semopm`: 100
- For `semnni`: 128

Because these parameters are positional, the following illustrates setting the parameters as indicated in `/etc/sysctl.conf`:

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

Parameters are viewable in:

```
# cat /proc/sys/kernel/sem
250      32000      100      128
```

# Network Tuning

- Socket parameters:
  - An IP port is assigned to a server process when it starts.
  - An IP port is used to communicate with the user process.
  - The default range is 32768 through 61000.

```
# cat /proc/sys/net/ipv4/ip_local_port_range
9000 65500
```
- TCP/IP window size parameters:
  - Define read (`rmem`) and write (`wmem`) window sizes.
  - Set the default and maximum memory allocated for the network send and receive buffers.

```
# ls /proc/sys/net/core/ [rw]mem*
rmem_default          wmem_default
rmem_max              wmem_max
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

An IP port is assigned to a database-dedicated server process when it starts. The IP port is used to communicate with the user process. By default, the range available is 32768 through 61000. In some databases with a very large number of users, the default range of ports that is available to non-root processes might not be adequate. In the following example, the IP port range is set to be from port 9000 through 65500:

```
# cat /proc/sys/net/ipv4/ip_local_port_range
9000 65500
```

On systems that use a firewall, a shared server configuration, or connection multiplexing, the number of needed ports can be greatly reduced.

TCP/IP window size parameters define the read (`rmem`) and write (`wmem`) window sizes for a TCP/IP packet. These parameters set the default and maximum memory allocated for the network send and receive buffers. Defaults are defined, and because TCP/IP communications occur with other machines, which can have different settings, you can adjust the sizes upward to attain compatibility. You cannot adjust them beyond the specified maximum value.

```
# ls /proc/sys/net/core/ [rw]mem*
rmem_default  (262144)          wmem_default  (262144)
rmem_max     (4194304)          wmem_max     (1048576)
```

## Setting the File Handles Parameter

- The File Handles parameter (`fs.file-max`) determines the maximum number of file handles that the Linux kernel allocates.
- The Oracle database background processes open all data files, logs, and other supporting files.
- The parameter must be high enough to include all the data files within your database and all supporting files.
- Set the kernel parameter in `/etc/sysctl.conf`:  
`fs.file-max = 6815744`
- View the setting in:  
`# cat /proc/sys/fs/file-max`  
6815744



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The File Handles parameter (`fs.file-max`) determines the maximum number of file handles that the Linux kernel allocates. The Oracle database background processes open all the data files in addition to redo logs, the alert log, and other supporting files. Therefore, `fs.file-max` must be high enough to include all the data files within your database and all supporting files.

This value is set in `/etc/sysctl.conf` and is viewable in `/proc/sys/fs/file-max`:

```
# cat /proc/sys/fs/file-max  
6815744
```

## Asynchronous IO (AIO)

- AIO is the kernel subsystem used to ensure that Oracle databases run properly on Linux.
- AIO allows a process to initiate several I/O operations without having to block or wait for any to complete.
- The process can retrieve the results of the I/O later.
- Set the maximum number of allowable concurrent requests kernel parameter in /etc/sysctl.conf:

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

- View the setting in:

```
# cat /proc/sys/fs/aio-max-nr  
1048576
```



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Asynchronous IO (AIO) kernel subsystem is used to make system calls asynchronously in a generic fashion to ensure that Oracle databases run properly on Linux. The idea behind AIO is to allow a process to initiate several I/O operations without having to block or wait for any to complete. At some later time, or after being notified of I/O completion, the process can retrieve the results of the I/O.

The /proc/sys/fs/aio-max-nr file is the maximum number of allowable concurrent requests.

```
# cat /proc/sys/fs/aio-max-nr  
1048576
```

## Oracle-Related Shell Limits

- Three shell limits must be set for the `oracle` user:
  - `nofile`: Number of open file descriptors
  - `nproc`: Number of processes available to a single user
  - `stack`: Size of the stack segment of the process
- Soft limit versus hard limit
  - A hard limit can be changed only by `root`.
  - A soft limit can be changed by the user, up to the value of the hard limit.
- Define limits in the `/etc/security/limits.conf` file.
- Edit the `/etc/pam.d/login` file.
- A user can change a soft limit by using the `ulimit` command, for example:
  - `$ ulimit -Sn 50`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

You must set three limits in order for an Oracle database to function properly. These apply to the `oracle` Linux user. You can typically set these limits to a high value.

The `nofile` limit is the maximum number of files that the user can have open at one time. The `oracle` user opens initialization files, data files, redo log files, and other files; therefore, this limit needs to be set high enough to have all those files open simultaneously.

The `nproc` limit is the maximum number of processes a given user can run at once. The `oracle` Linux user owns and starts all the background processes, server processes, and possibly the parallel query and dispatcher processes. This number must be set high enough to accommodate that. You must set this parameter high enough to manage the highest number of sessions in the database, plus some for other processes.

The `stack` limit is the size of the stack segment of the process.

For each of these settings, there is a soft limit and a hard limit. The hard limit can be changed only by the `root` user. The soft limit serves as the limit for the resource at any given time, which the user cannot exceed. But the user can change the soft limit, up to the value of the hard limit. The purpose of a limit is to prevent runaway situations where resources are being used up beyond what was intended by the processes running in the user space. Allowing the soft limit to be adjusted by the user, but never exceeding the `root`-defined hard limit, provides flexibility along with control.

## Setting Shell Limits

The following example sets hard and soft limits for the `oracle` user. Two different files are modified:

1. Add the following to the `/etc/security/limits.conf` file:

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

2. Add or edit the following lines in the `/etc/pam.d/login` file:

```
session required pam_limits.so
```

The `pam_limits.so` file is a Pluggable Authentication Module (PAM) that sets limits on the system resources that can be obtained in a user session. By default, limits are taken from the `/etc/security/limits.conf` file.

After a user has started a shell, the user can use the `ulimit` command to adjust the hard limit and soft limit for this specific shell. The hard limit cannot be increased after it is set, and the soft limit cannot be increased above the hard limit. In the following example, the `ulimit` command has no effect, it is setting the hard limit and soft limit to the same value that they have already been set to:

```
$ ulimit -u 16384 -n 65536
```

If the user issues the `ulimit -Sn 50` command (which sets the soft limit for the number of open files to 50), any attempt to open more than that results in an error. The user could still set it higher (for example, `ulimit -Sn 100`), which would result only in errors when the number of open file requests exceeds 100. However, the soft limit cannot be set higher than the hard limit.

Because a process inherits these settings from the shell (from which it is started) at the time that it is started, if you change the settings, any processes would have to be restarted for them to take effect. For example, if the shell limit values were changed, the Oracle database would have to be shut down and restarted.

# HugePages

- HugePages:
  - Allow larger pages to manage memory
  - Are crucial for faster Oracle database performance
  - Are useful in both 32- and 64-bit configurations
  - Are integrated into the Linux kernel with release 2.6
  - Have been back-ported to some 2.4 kernels (2.4.21), but are implemented differently
  - Decrease page table overhead
  - Provide faster overall memory performance
  - Must be reserved during system startup
  - Are not swappable—there is no page-in/page-out overhead
- HugePage sizes vary from 2 MB to 256 MB, based on the kernel version and the hardware architecture.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

HugePages is a feature of the Linux kernel. HugePages allow larger pages to manage memory as the alternative to small 4 KB page sizes (16 K for IA64). HugePages are crucial for faster Oracle database performance on Linux if you have large RAM and SGA. If your combined database SGA is large (for example, more than 8 GB—but HugePages can also be important for smaller databases), you need HugePages configured. The HugePages feature is useful in both 32- and 64-bit configurations and is integrated into the Linux kernel with release 2.6.

## HugePages Facts/Features

- The HugePages feature is back-ported to some 2.4 kernels. Kernel versions 2.4.21-\* have this feature, but it is implemented in a different way. The difference from the 2.6 implementation is the organization within the source code and the kernel parameters that are used for configuring HugePages.
- HugePages can be allocated dynamically, but they must be reserved during system startup. Otherwise, the allocation might fail, because the memory is already paged in mostly 4 KB.
- HugePages are not subject to reservation/release after system startup unless there is system administrator intervention (basically changing the HugePages configuration).
- HugePages are not swappable; therefore, there is no page-in/page-out mechanism overhead. HugePages are universally regarded as pinned (never swapped to secondary storage).

- No kswapd operations: The kernel swap daemon, kswapd, gets very busy if there is a very large area to be paged (13 million page table entries for 50 GB memory) and uses an incredible amount of CPU resource. When HugePages are used, kswapd is not involved in managing them.
- HugePages allow fewer translations to be loaded into the Translation Lookaside Buffer (TLB). A TLB is a buffer (or cache) in a CPU that contains parts of the page table. This is a fixed-size buffer used for faster virtual address translation. A hugetlb is an entry in the TLB that points to a HugePage. HugePages are implemented via hugetlb entries (a HugePage is handled by a “hugetlb page entry”). The “hugetlb” term is also used synonymously with a HugePage.
- TLB entries cover a larger part of the address space when using HugePages. There are fewer TLB misses before the entire SGA, or most of it, is mapped in the TLB.
- Fewer TLB entries for the SGA also means more room for other parts of the address space.
- Decreased page table overhead: A page table is the data structure of a virtual memory system in an operating system to store the mapping between virtual addresses and physical addresses. This means that on a virtual memory system, the memory is accessed by first accessing a page table and then accessing the actual memory location implicitly.
- Eliminated page table lookup overhead: Because the pages are not subject to replacement, page table lookups are not required.
- Faster overall memory performance: On virtual memory systems, each memory operation is actually two abstract memory operations. Because there are fewer pages to work on, the possible bottleneck on page table access is clearly avoided.
- Oracle 11g Automatic Memory Management (AMM) and HugePages are not compatible. You must disable AMM on 11g to be able to use HugePages.

### Size of a HugePage

HugePage sizes vary from 2 MB to 256 MB based on kernel version and hardware architecture. The following table shows the sizes of HugePages on different configurations:

<u>Hardware Platform</u>	<u>Source Code Tree</u>	<u>Kernel 2.4</u>	<u>Kernel 2.6</u>
Linux x86 (IA32)	i386	4 MB	4 MB
Linux x86-64 (AMD64, EM64T)	x86_64	2 MB	2 MB
Linux Itanium (IA64)	ia64	256 MB	256 MB
IBM Power Based Linux (PPC64)	ppc64/powerpc	N/A	16 MB
IBM zSeries Based Linux	s390	N/A	N/A
IBM S/390 Based Linux	s390	N/A	N/A

### Configuring HugePages

Configuring your Linux OS for HugePages is a delicate process. If you do not configure properly, the system may experience serious problems such as:

- HugePages not used (HugePages\_Total = HugePages\_Free), wasting the amount of memory configured for HugePages
- Poor database performance
- System running out of memory or excessive swapping
- Some or all database instances cannot be started
- Crucial system services failing (for example, CRS)

# Configuring HugePages

- Guidelines exist for different OS versions and hardware architectures.
- Configuring HugePages on 64-bit Linux:
  - Set the memlock user limit in `/etc/security/limits.conf` slightly smaller than installed RAM.
  - Disable AMM by setting `MEMORY_TARGET` and `MEMORY_MAX_TARGET` to zero.
  - Use the `hugepages_settings.sh` script to calculate the recommended value for the `vm.nr_hugepages` parameter.
  - Edit `/etc/sysctl.conf` and set the `vm.nr_hugepages` parameter.
  - Reboot your system.
- To check: `grep HugePages /proc/meminfo`

**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

General guidelines exist to configure HugePages for more than one Oracle RDBMS instance. The following guidelines exist for the different OS versions and hardware architectures:

- “How to Configure RHEL 3.0 32-bit for Very Large Memory with ramfs and hugepages”
- “How to Configure Asianux 1.0 32-bit for Very Large Memory with ramfs and hugepages”
- “How to Configure RHEL 4 32-bit for Very Large Memory with ramfs and hugepages”
- “How to Configure SuSE SLES 9 32-bit for Very Large Memory with ramfs and hugepages”
- “How to Configure HugePages on 64-bit Linux”

## HugePages on 64-bit Linux

The following are the configuration steps for configuring HugePages on 64-bit Linux. The configuration steps provided here are primarily for Oracle Linux, but the same concepts and configurations should apply to other Linux distributions. These configuration steps guide you to do a persistent system configuration, which requires a reboot of the system.

**Step 1:** Have the memlock user limit set in the `/etc/security/limits.conf` file. Set the value (in KB) slightly smaller than installed RAM. If you have 64-GB RAM installed, set:

```
soft    memlock    60397977  
hard    memlock    60397977
```

There is no harm in setting this value larger than your SGA requirements. The parameters are set by default on:

- Oracle Linux with Oracle-validated package installed
- Oracle Exadata DB compute nodes

**Step 2:** Log on again to the Oracle product owner account (for example, oracle) and check the memlock limit:

```
$ ulimit -l
60397977
```

**Step 3:** If you have Oracle Database 11g or later, the default database created uses the Automatic Memory Management (AMM) feature, which is incompatible with HugePages. Disable AMM before proceeding. To disable AMM, set the initialization parameters MEMORY\_TARGET and MEMORY\_MAX\_TARGET to 0 (zero).

**Step 4:** Make sure that all your database instances are up (including ASM instances) as they would run on production. Use the `hugepages_settings.sh` script in Document 401749.1 to calculate the recommended value for the `vm.nr_hugepages` kernel parameter:

```
$ ./hugepages_settings.sh
...
Recommended setting: vm.nr_hugepages = 1496
```

You can also calculate a proper value for the parameter yourself but that is not advised if you do not have extensive experience with HugePages.

**Step 5:** Edit the `/etc/sysctl.conf` file and set the `vm.nr_hugepages` parameter:

```
vm.nr_hugepages = 1496
```

This will cause the parameter to be set properly with each reboot.

**Step 6:** Stop all the database instances and reboot the server.

The performed configuration is basically based on the RAM installed and combined size of SGA of database instances that you are running. If any of the following changes occur, you should revise your HugePages configuration to make it suitable to the new memory framework:

- Changes to the amount of RAM installed for the Linux OS
- New database instance(s) introduced
- Changes to SGA size or configuration for one or more database instances

### Check and Validate the Configuration

After the system is rebooted, make sure that your database instances (including the ASM instances) are started. Automatic startup via OS configuration or CRS, or manual startup (whichever method you use) should have been performed. Check the HugePages state from `/proc/meminfo`:

```
# grep HugePages /proc/meminfo
HugePages_Total:      1496
HugePages_Free:       485
HugePages_Rsvd:       446
HugePages_Surp:        0
```

The values in the output will vary. To make sure that the configuration is valid, the `HugePages_Free` value should be smaller than `HugePages_Total` and there should be some `HugePages_Rsvd`. The sum of `HugePages_Free` and `HugePages_Rsvd` may be smaller than your total combined SGA as instances allocate pages dynamically and proactively as needed.

# Oracle Database Smart Flash Cache (DBSFC)

- DBSFC:
  - Is available for both Oracle Solaris and Oracle Linux customers with the 11g R2 database
  - Allows you to extend the Oracle Buffer Cache in memory (SGA) using secondary flash-based storage
  - Helps with read-only/read-mostly workloads
- When a block gets modified, it is modified in the standard database buffer cache, written to disk and copied over into the flash cache.
- A subsequent read can then be from this fast storage instead of from the originating data files.
- See <http://www.oracle.com/technetwork/articles/servers-storage-admin/smart-flash-cache-oracle-perf-361527.html>.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Oracle Database Smart Flash Cache (DBSFC) feature is available for both Oracle Solaris and Oracle Linux customers with the 11g R2 database. DBSFC allows you to extend the Oracle Buffer Cache in memory (SGA) by using secondary flash-based storage. This flash-based storage can be presented to the database through a file on a file system on flash storage, through a raw disk device (flash-based), or by adding flash storage to Oracle ASM and creating a region inside ASM. See <http://www.oracle.com/technetwork/articles/servers-storage-admin/smart-flash-cache-oracle-perf-361527.html> for more information.

DBSFC is a read-only cache extension that helps with read-only/read-mostly workloads. It contains clean blocks that are removed from the buffercache/sga and now first get placed in this extended cache. A subsequent read can then be from this fast storage instead of from the originating data files. When a block gets modified, it is modified in the standard database buffer cache, written to disk, and copied over into the flash cache.

The white paper referenced previously provides the details on how to use it and how to configure it in an Oracle Linux environment. But to summarize, you simply specify `DB_FLASH_CACHE_FILE` and `DB_FLASH_CACHE_SIZE` in the Oracle Initialization File, `init.ora`. Any Oracle Database customer using Oracle Linux can use this feature.

# Oracle Pre-Install RPM

Oracle RDBMS Pre-Install RPM for Oracle Linux 6:

- Completes most pre-installation configuration tasks
- Is different from Oracle Validated RPM for Oracle Linux 5
- Downloads and installs various software packages and specific versions needed for database installation
- Creates the user `oracle` and the groups `oinstall` and `dba`
- Modifies kernel parameters in `/etc/sysctl.conf`
- Sets hard and soft shell resource limits in `/etc/security/limits.d` directory
- Sets `numa=off` in the kernel boot parameters for `x86_64` machines



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Oracle RDBMS Pre-install RPM package is designed specifically for Oracle Linux 6 to aid in the installation of the Oracle Database. You can complete most pre-installation configuration tasks by using this package, which is now available from the Unbreakable Linux Network or from the Oracle Public Yum repository.

This package was formerly known as `oracle-validated`. For Oracle Linux 6 and newer, the name of the package was changed to `oracle-rdbms-server-<version>-preinstall`. As of this writing, there are two versions of the `oracle-rdbms-preinstall` RPM:

- `oracle-rdbms-server-11gR2-preinstall`
- `oracle-rdbms-server-12cR1-preinstall`

There is no “oracle-validated” RPM for Oracle Linux 6, and there is no “oracle-rdbms-preinstall” RPM for Oracle Linux 5. The key difference between `oracle-validated` and the new pre-install RPM is that the new package sets the bare minimums for Oracle DB installations, instead of the testing maximums set with `oracle-validated`. The new pre-install RPM configures an Oracle Linux 6 machine so that you can immediately run the OUI database installation. The pre-install package is available for `x86_64` only. Specifically, the package:

- Downloads and installs the various software packages and specific versions needed for database installation, with package dependencies resolved via `yum`.

The pre-install package also performs the following tasks:

- It creates the user `oracle` and the groups `oinstall` and `dba`, which are the defaults used during database installation.
- It modifies kernel parameters in `/etc/sysctl.conf` to change settings for shared memory, semaphores, the maximum number of file descriptors, and so on.
- The “11g R2” package sets hard and soft shell resource limits in `/etc/security/limits.conf`, such as the number of open files, the number of processes, and stack size to the minimum required based on the Oracle Database 11g Release 2 Server installation requirements. The “12c R1” version sets limits by using a file in the `/etc/security/limits.d` directory.
- It sets `numa=off` in the kernel boot parameters for `x86_64` machines.

Further details for the “11g R2” version are available at <http://oss.oracle.com/pipermail/el-errata/2012-March/002727.html>.

# Oracle ASM

- For stand-alone or Oracle RAC databases, you must have space available on Oracle ASM.
  - Creating Oracle Clusterware files on block or raw devices is no longer supported.
- ASM performs the functions of a volume manager and a file system.
- ASM consists of a specialized Oracle instance and a set of disk groups that are managed through the ASM instance.
- A disk group is a set of disk devices that ASM manages.
  - Each disk device can be a partition, a logical volume, a RAID array, or a single disk.
  - ASM spreads data evenly across the disk group to optimize performance and usage.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

If you install stand-alone or Oracle RAC Databases, you must have space available on Oracle ASM for Oracle Clusterware files (voting disks and Oracle Cluster Registries), and for Oracle Database files. Creating Oracle Clusterware files on block or raw devices is no longer supported for new installations.

ASM consists of a specialized Oracle instance and a set of disk groups that are managed through the ASM instance. ASM performs the functions of a volume manager and a file system. ASM can be used for single instance or clustered databases. When using Oracle ASM for either the Oracle Clusterware files or Oracle Database files, Oracle creates one Oracle ASM instance on each node in the cluster, regardless of the number of databases.

The ASM instance manages disks in disk groups. An ASM instance must be configured and running before a database instance can access ASM files. This configuration is performed automatically if the Database Configuration Assistant is used for database creation.

A disk group is a set of disk devices that ASM manages as a single unit. Each disk device is a block device: a partition, logical volume, a RAID array, or a single disk. ASM spreads data evenly across all the devices in the disk group to optimize performance and usage. You can add or remove disk devices from a disk group without shutting down the database. When you add or remove devices, ASM rebalances the files across the disk group. You can create multiple disk groups to handle specific tasks, such as database backup and recovery operations, in addition to database file storage activities.

## **Grid Installation Owner and ASMOPER**

During installation, in the Privileged Operating System Groups window, it is now optional to designate a group as the OSOPER for ASM group. If you choose to create an OSOPER for ASM group, then you can enter a group name configured on all cluster member nodes for the OSOPER for ASM group. In addition, the Oracle Grid Infrastructure installation owner no longer is required to be a member.

## **Oracle ASM Job Role Separation Option with SYSASM**

The SYSASM privilege that was introduced in Oracle ASM 11g release 1 (11.1) is now fully separated from the SYSDBA privilege. If you choose to use this optional feature, and designate different operating system groups as the OSASM and the OSDBA groups, then the SYSASM administrative privilege is available only to members of the OSASM group. The SYSASM privilege can also be granted using password authentication on the Oracle ASM instance.

OSASM is an operating system group that is used exclusively for Oracle ASM. Members of the OSASM group can connect as SYSASM using operating system authentication and have full access to Oracle ASM.

You can designate OPERATOR privileges (a subset of the SYSASM privileges, including starting and stopping Oracle ASM) to members of the OSOPER for ASM group.

Providing system privileges for the storage tier by using the SYSASM privilege instead of the SYSDBA privilege provides a clearer division of responsibility between Oracle ASM administration and database administration, and helps to prevent different databases using the same storage from accidentally overwriting each other's files.

## ASM Library Driver (ASMLib)

- ASMLib simplifies the management of ASM disks.
- ASMLib has three components:
  - oracleasm-support: Provides user space shell scripts, and is included with the Oracle Linux distribution
  - oracleasmlib: Provides the user space library, and is installed from Unbreakable Linux Network (ULN)
  - oracleasm: Is the kernel driver included in kernel-uek
- To configure ASMLib:
  - # oracleasm configure -i
- To mark disks as ASM disks:
  - # oracleasm createdisk *ASM\_DISK\_NAME* *candidate\_disk*



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

If you intend to use ASM for database storage for Linux, Oracle recommends that you install the ASMLib RPMs to simplify storage administration.

ASMLib is free, optional software for the ASM feature of Oracle Database. ASMLib simplifies the management and discovery of ASM disks and makes I/O processing and kernel resource usage with ASM storage more efficient. It provides persistent paths and permissions for storage devices used with ASM, eliminating the need for updating `udev` or `devlabel` files with storage device paths and permissions.

ASMLib also contains Linux data integrity features. To enable Oracle application-to-disk data integrity checking, ASMLib must be used. The ASMLib kernel driver is what connects the data integrity dots between Oracle database and ASM. See the following for more information:  
<http://oss.oracle.com/~mfp/docs/data-integrity-webcast.pdf>.

ASMLib updates are delivered via Unbreakable Linux Network (ULN) for both Oracle Linux or Red Hat Enterprise Linux installations. For Oracle Linux 6, to use ASMLib, you must replace any Red Hat kernel with Unbreakable Enterprise Kernel. For Oracle Linux 5, Oracle does provide ASMLib for the Red Hat compatible kernel. Refer to the following for more information:

<http://ovmjira.us.oracle.com/confluence/display/OLPM/Oracle+Linux+FAQ#OracleLinuxFAQ-ASMLib>.

ASMLib has three components:

- **oracleasm-support**: This package provides user space shell scripts.
- **oracleasmlib**: This package provides the user space library and is closed source.
- **oracleasm**: This is the kernel driver and is included in kernel-uek.

The `oracleasm-support` package is included with the Oracle Linux distribution. Install the `oracleasmlib` package from ULN. The `oracleasm` kernel driver is included in the UEK. You do not need to install any driver package when using this kernel.

The following web page describes getting ASMLib from ULN.

```
http://www.oracle.com/technetwork/server-storage/linux/uln-095759.html
```

The full installation guide is part of the *Oracle Database Documentation*.

## Configuring ASMLib

Configure ASMLib by logging in as `root` and entering the following command:

```
# oracleasm configure -i
```

You are prompted to provide the following information:

- The default user to own the driver interface
- The default group to own the driver interface
- Whether to start Oracle ASM library driver on boot
- Whether to fix permissions of Oracle ASM disks on boot

The user who owns the driver interface should be the same user who owns the software installation, typically `oracle`. The group to own the driver interface should be the group used for DBAs, typically `dba`. You want ASMLib to start when the system boots, and to fix permissions of ASM disks. If you enter the command `oracleasm` configured without the `-i` flag, then you are shown the current configuration. After it is configured, to load and initialize the ASMLib driver, run the `oracleasm` utility with the `init` option as shown:

```
# oracleasm init
```

## Marking Disks as ASM Disks

A disk that is configured for use with ASM is known as a candidate disk. For OUI to recognize partitions as Oracle ASM disk candidates, you must log in as `root` and mark the disk partitions that Oracle ASM can use. Disks are marked by using the `createdisk` option. Use the following syntax, where `ASM_DISK_NAME` is the name of the Oracle ASM disk group, and `candidate_disk` is the name of the disk device that you want to assign to that disk group:

```
# oracleasm createdisk ASM_DISK_NAME candidate_disk
```

Meaningful names can be assigned for each disk. You can create multiple disk groups. By providing descriptive names to each disk, you have an easier time assigning disks to disk groups when creating the ASM instance. When choosing names for drives, consider using the physical location of the drive in the name. Example:

```
# oracleasm createdisk VOL1 /dev/sda1
# oracleasm createdisk VOL2 /dev/sdb1
# oracleasm createdisk VOL3 /dev/sde1
```

To make the disks available, use the `scandisks` option as shown:

```
# /etc/init.d/oracleasm scandisks
```

## Using ASMLib Commands

Available options for the `oracleasm` script:

- `configure`: Configure the ASM library driver.
- `init/exit`: Change the behavior of the ASMLib when the system starts.
- `createdisk`: Mark a disk device for use with ASM.
- `deletedisk`: Unmark a named disk device.
- `querydisk`: Determine whether a disk device or disk name is being used by the ASMLib.
- `listdisks`: List the disk names of marked disks.
- `scandisks`: Enable cluster nodes to identify which shared disks have been marked as ASMLib disks on another node.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

To administer the Automatic Storage Management library driver and disks, use the `oracleasm` initialization script with different options. The following summarizes the available options for the `oracleasm` script:

- `configure`: Use this to configure the Automatic Storage Management library driver.
- `init/exit`: Use this to change the behavior of the ASMLib when the system starts. The `init` option causes the ASMLib driver to load when the system starts.
- `createdisk`: Use this to mark a disk device for use with the ASMLib and give it a name.
- `deletedisk`: Use this to unmark a named disk device. Do not use this command to unmark disks that are being used by an ASM disk group. You must drop the disk from the disk group, before you unmark it. The syntax is as follows:  

```
# oracleasm deletedisk DISKNAME
```
- `querydisk`: Use this to determine whether a disk device or disk name is being used by the ASMLib. The syntax is as follows:  

```
# oracleasm querydisk {DISKNAME|devicename}
```

- **listdisks:** Use this to list the disk names of marked ASMLib disks.
- **scandisks:** Use this to enable cluster nodes to identify which shared disks have been marked as ASMLib disks on another node.

## ASM Rebalance Operations

ASM attempts to use the same amount of space on all the disks of a disk group. The data is striped and mirrored across all the disks of a disk group at the file level. Even though the disk group has a default for mirroring and striping, each file can have its own stripe and mirror properties.

There are two modes of striping:

- 1 MB allocation units
- 128 KB units

The redundancy can be set to one of the following:

- **Normal:** Normal redundancy is two-way mirroring.
- **High:** High redundancy is three-way mirroring.
- **External:** External redundancy does no mirroring. It assumes that the disk volumes are mirrored by some external means, such as RAID 1 arrays.

When a disk is added to a disk group, a rebalance operation is started. ASM moves a set of data blocks (allocation units) from the existing disks to the new disk. The number of allocation units moved is proportional to the size of the new disk compared to the total size of the disk group. If a disk is dropped from the disk group, or fails, then the data is redistributed across the remaining disks to re-establish the redundancy requirements.

The rebalance operation is controlled through an ASM instance parameter or by a parameter associated with the operation. This parameter is named `ASM_POWER_LIMIT` and can be set from 0-11. A setting of 0 stops the rebalance, and 11 takes all the resources that can be effectively used to minimize the time to complete the operation. A setting of 1 is the default to prevent rebalance operations from interfering with normal database operations.

Whenever a disk group is altered by adding or dropping disks, a rebalance operation is triggered. If there is insufficient remaining disk space for a drop operation, the `alter` command fails. The `alter disk group` command does not complete until the rebalance operation is finished.

## Quiz

Which of the following statements is true with regard to ASM?

- a. ASMLib is required to use ASM.
- b. The `oracleasm` kernel driver is included in the Red Hat Compatible Kernel (RHCK).
- c. For Oracle Universal Installer (OUI) to recognize partitions as Oracle ASM disk candidates, you must mark the disk partitions that Oracle ASM can use.
- d. A RAID array cannot be included in an ASM disk group.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Summary

In this lesson, you should have learned how to:

- Prepare your Oracle Linux server for Oracle Database installation
- Create Oracle software user and group accounts
- Set kernel parameters for Oracle Database
- Set Oracle database shell limits
- Configure HugePages
- Configure Oracle Database Smart Flash Cache (DBSFC)
- Use Oracle pre-install RPM
- Install and use ASMLib



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 18: Overview

The practices for this lesson cover the following:

- Using `sftp` to upload `oracle-rdbms-server` package
- Installing and running Oracle RDBMS Pre-install
- Preparing disks for ASM use
- Installing and configuring ASMLib

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# 19

## System Monitoring

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Use the `sosreport` utility
- Use the `iostat`, `mpstat`, `vmstat`, `sar`, `top`, `iotop`, and `strace` utilities
- Use the `netstat` and `tcpdump` utilities
- Use the Wireshark network analyzer GUI
- Use the OSWatcher Black Box (OSWbb) tool
- Use OSWatcher Black Box Analyzer (OSWbba)
- Describe Enterprise Manager Ops Center
- Describe Linux Patch and Provisioning using Enterprise Manager Ops Center
- Describe Spacewalk



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## sosreport Utility

- The **sosreport** utility:
  - Collects debugging information about a system
  - Stores the information in a single compressed file in /tmp
- Run the tool as follows:

```
# sosreport
...
Press ENTER to continue, or CTRL-C to quit.
Please enter your first initial and last name:
Please enter the case number...:
```
- **sosreport** uses plug-ins. Options exist to manage plug-ins:
  - -l: List the status of all available plug-ins.
  - -n *PLUGNAME*: Do not load specified plug-in(s).
  - -e *PLUGNAME*: Enable the specified plug-in(s).

**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The **sosreport** tool collects information about a system, such as hardware configuration, installed software packages, configuration, and operational state. You can also enable diagnostics and analytical functions. This information is stored in a single compressed file in the /tmp directory, and the file can be sent to a support representative to assist in troubleshooting a problem. The **sosreport** tool replaces an earlier version of the tool called **sysreport**.

To run the tool, first install the **sos** package:

```
# yum install sos
```

Run the report as the root user. The version of the tool is displayed along with a short description of the tool and the output it produces. You are prompted to press Enter to continue or Ctrl + C to quit.

```
# sosreport
...
Press ENTER to continue, or CTRL-C to quit.
```

Press Enter to start. You are prompted as follows:

```
Please enter your first initial and last name [host03]:
Please enter the case number you are generating this report for:
```

The name and case number that you provide becomes part of the file name created by the tool. After the tool completes, you can uncompress the file and view the contents, by doing the following:

```
# cd /tmp
# xz -d <sosfile>.xz
# tar xvf <sosfile>.tar
```

Extracting the file creates a directory, which includes the output of several system status commands as well as the contents of some configuration directories on your system. The following is a sample list of the output collected on a system named host03:

```
# ls /tmp/host03*
boot/          free           lib/          netstat        sar20
uname          chkconfig      hostname      lsb-release   /proc
sbin/          uptime         date         ifconfig      lsmod
...
...
```

The sosreport uses plug-ins, which can be turned on and off. Use the following command to list the plug-ins, which are enabled and disabled, and plug-in options:

```
# sosreport -l
The following plugins are currently enabled:
acpid          acpid related information
anaconda       Anaconda / Installation information
...
The following plugins are currently disabled:
amd            Amd automounter information
cloudforms     Cloudforms related information...
...
The following plugin options are available:
apache.log      off gathers all apache logs
auditd.syslogsize 15 max size (MiB) to collect per syslog file
...
...
```

Additional options exist to control the plug-ins and the tool. The following is a partial list:

- **-n PLUGNAME:** Do not load specified plug-in(s).
- **-e PLUGNAME:** Enable the specified plug-in(s).
- **-o PLUGNAME:** Enable only the specified plug-in(s), disable all others.
- **-k PLUGNAME.PLUGOPT=[VALUE]:** Specify options for plug-ins.
- **--upload FTP\_SERVER:** Upload the report to Oracle.
- **-a:** Enable all (Boolean) options for all loaded plug-ins.
- **--diagnose:** Turn on diagnostic functions.

# iostat Utility

- The iostat utility:
  - Reports CPU and I/O statistics
  - Is used during performance analysis to balance I/O load
- The iostat utility report has the following sections:
  - CPU utilization
  - Device utilization
  - NFS statistics
- Include the `-x` option for extended statistics:
  - `# iostat -x`
- Execute iostat continuously at a specific *interval*, up to *count* times:
  - `# iostat interval count`
- For example, to run iostat every 10 seconds for 5 times:
  - `# iostat 10 5`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The iostat command is used for monitoring system input/output device loading by observing the time that the physical disks are active in relation to their average transfer rates. This information can be used to change system configuration to better balance the input/output load between physical disks and adapters.

```
# iostat
Linux 3.8.13-16.2.1.el6uek.x86_64 (host03.example.com)
12/20/2013 _x86_64_ (1 CPU)

avg-cpu: %user   %nice  %system %iowait  %steal    %idle
          25.99     0.78     7.43    12.77     0.00    53.03

Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
xvda   27.40      797.19      201.27    800902    202208
xvdb   0.03       1.24        0.00      1248         0
```

The first line displays the Linux kernel version, host name, current date, architecture, and number of CPUs on your system.

## CPU Utilization Report

The next two lines display CPU statistics. For multiprocessor systems, the CPU values are global averages among all processors. The columns are defined as follows:

- **%user**: The percentage of CPU used while executing applications at the user level
- **%nice**: The percentage of CPU used while executing at the user level with nice priority
- **%system**: The percentage of CPU used while executing at the system (kernel) level
- **%iowait**: The percentage of time the CPU(s) were idle while the system had an outstanding disk I/O request
- **%steal**: The percentage of time spent in involuntary wait by the virtual CPU or CPUs while the hypervisor was servicing another virtual processor
- **%idle**: The percentage of time that the CPU was (or the COUs were) idle and the system did not have an outstanding disk I/O request

## Device Utilization Report

The remaining lines in the example display statistics on a per-physical device or per-partition basis. You can include block devices and partitions as arguments to the `iostat` command. If no arguments are included, the report displays all devices that the kernel has statistics for. The columns are defined as follows:

- **Device**: Device or partition name as listed in the `/dev` directory
- **tps**: Number of transfers (I/O request) per second issued to the device
- **Blk\_read/s**: Amount of data read from the device expressed in number of blocks per second. Block size with kernels 2.4 and later are 512 bytes.
- **Blk\_wrtn/s**: Amount of data written to the device expressed in number of blocks per second
- **Blk\_read**: Total number of blocks read
- **Blk\_wrtn**: Total number of blocks written

## Network File system Report

The example does not include NFS statistics, but `iostat` also provides statistics for each mounted network file system. This section displays the host name of the NFS server followed by the directory name where the file system is mounted. Read and write information is displayed for each file system, depending on the command-line options provided.

More detailed statistics can be included by providing different options to the `iostat` command. Some of the command-line options are listed:

- **-c**: Display the CPU utilization report.
- **-d**: Display the device utilization report.
- **-n**: Display the network file system (NFS) report.
- **-k**: Display statistics in kilobytes per second instead of blocks per second.
- **-x**: Display extended statistics.

Multiple reports can be run at different intervals by using *interval* and *count* arguments. The following example displays 6 reports at 2-second intervals for all devices:

```
# iostat -d 2 6
```

## mpstat Utility

- The mpstat utility:
  - Collects and displays performance statistics for all CPUs
  - Is used during performance analysis to determine CPU utilization
- Use the -P ALL option to include average usage of all CPUs:
  - # mpstat -P ALL
- Execute continuously at a specific *interval*, up to *count* times:
  - # mpstat *interval count*
- For example, to run mpstat every 2 seconds for 5 times:
  - # mpstat 2 5

**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The mpstat command collects and displays performance statistics for all logical CPUs in the system. When a CPU is occupied by a process, it is unavailable for processing other requests. These other processes must wait until the CPU is free. The mpstat command provides CPU usage to help you identify CPU-related performance problems.

```
# mpstat
Linux 3.8.13-16.2.1.el6uek.x86_64 (host03.example.com)
12/20/2013 _x86_64_ (2 CPUs)

07:44:18      CPU    %usr    %nice    %sys %iowait    %irq    %soft
%steal    %guest    %idle
07:44:18      all     3.01   57.31     0.36     0.13     0.01     0.00
0.00        0.00     0.00
07:44:18        0     5.87   69.47     0.44     0.05     0.01     0.01
0.00        0.00     0.00
07:44:18        1     1.79   48.59     0.36     0.23     0.00     0.00
0.00        0.00     0.00
```

The first line displays the Linux kernel version, host name, current date, architecture, and number of CPUs on your system.

The first column is a time stamp. The remaining columns are defined as follows:

- **CPU**: Processor number starting at 0. The keyword `all` indicates that statistics are calculated as averages among all processors.
- **%usr**: Percentage of CPU used while executing at the user level
- **%nice**: Percentage of CPU used while executing at the user level with nice priority
- **%sys**: Percentage of CPU used while executing at the system (kernel) level. This does not include time spent servicing hardware and software interrupts.
- **%iowait**: Percentage of time the CPU was (or the COUs were) idle during which the system had an outstanding disk I/O request
- **%irq**: Percentage of time spent by the CPU(s) to service hardware interrupts
- **%soft**: Percentage of time spent by the CPU(s) to service software interrupts
- **%steal**: Percentage of time spent in involuntary wait by the virtual CPU or CPUs while the hypervisor was servicing another virtual processor
- **%guest**: Percentage of time spent by the CPU or CPUs to run a virtual processor
- **%idle**: Percentage of time that the CPU was (or the COUs were) idle and the system did not have an outstanding disk I/O request

Similar to the `iostat` utility, `mpstat` allows multiple reports to run at different intervals. Use the following arguments:

```
# mpstat interval count
```

If you omit the `count` argument, the report runs at `interval` continuously. Press Ctrl + C to stop the report. The following example displays a report every 3 seconds, until terminated by pressing Ctrl + C:

```
# mpstat 3
```

The `-P` option followed by the keyword `ALL` displays statistics for processors. To report on a specific CPU, include the processor number as an argument to `-P`. The following displays 5 separate reports at 2-second intervals of all processors, and includes an average line:

```
# mpstat -P ALL 2 5
```

## vmstat Utility

- The vmstat utility:
  - Monitors system memory usage
  - Is useful for detecting shortages of physical memory
- The vmstat report has six sections:
  - Processes: Numbers of processes in wait or sleep states
  - Memory: Amounts of memory free, and amounts used for virtual memory, buffers, and cache
  - Swap: Number of page-ins and page-outs
  - IO: Number of blocks received and sent
  - System: Number of interrupts and context switches
  - CPU time: Percentages for user, kernel, idle, iowait, and stolen
- **Recommended:** Run the utility with a delay interval:
  - # vmstat 5
- Additional options are available.

**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The vmstat command allows you to monitor your system's memory usage. It shows how much virtual memory there is, and how much is free and paging activity. You can observe page-ins and page-outs as they happen. This is extremely useful for detecting shortages of physical memory, which can adversely affect system performance.

The vmstat output contains more than just memory statistics. Output is broken up into six sections: procs, memory, swap, io, system, and cpu. To prevent the sample output from wrapping, the output is shown in two parts. As with iostat and mpstat, vmstat accepts *interval* and *count* arguments. The following example runs 3 reports 5 seconds apart:

```
# vmstat 5 3
procs -----memory----- ---swap--
r b swpd free buff cache si so
1 0 13344 1444 1308 19692 0 168
1 0 13856 1640 1308 18524 64 516
3 0 13856 1084 1308 18316 56 64
```

This portion of the sample output shows only the first three sections. These three sections are described before the remaining three sections are shown.

The first two columns give information on processes:

- **r**: Number of processes that are in a wait state. These processes are not doing anything but waiting to run.
- **b**: Number of processes that were in sleep mode and were interrupted since the last update

The next four columns give information on memory:

- **swpd**: Amount of virtual memory used
- **free**: Amount of idle memory
- **buff**: Amount of memory used as buffers
- **cache**: Amount of memory used as cache

The next two columns give information on swap:

- **si**: Amount of memory swapped in from disk (per second)
- **so**: Amount of memory swapped out to disk (per second)

Nonzero **si** and **so** numbers indicate that there is not enough physical memory and that the kernel is swapping memory to disk.

The remaining three sections of the `vmstat` report:

```
# vmstat 5 3
-----io---- --system-- -----cpu-----
      bi      bo      in      cs      us      sy      id      wa      st
      129      42    1505     713     20     11     69     0     0
      379     129    4341     646     24     34     42     0     0
      14       0    320    1022     84      9      7     0     0
```

The first two columns give information on I/O (input-output):

- **bi**: Number of blocks per second received from a block device
- **bo**: Number of blocks per second sent to a block device

The next two columns give the following system information:

- **in**: Number of interrupts per second, including the clock
- **cs**: Number of context switches per second

The last five columns give the percentages of total CPU time:

- **us**: Percentage of CPU cycles spent on user processes
- **sy**: Percentage of CPU cycles spent on system (kernel) processes
- **id**: Percentage of CPU cycles spent idle
- **wa**: Percentage of CPU cycles spent waiting for IO
- **st**: Percentage of CPU cycles stolen from a virtual machine

Additional information can be included by providing different options to the `vmstat` command.

Some of the command-line options are listed:

- **-a**: Display active and inactive memory.
- **-f**: Display the number of forks since boot.
- **-t**: Add a time stamp to the output.
- **-d**: Report the disk statistics.

## sar Utility

- Provided by the sysstat package:
  - sar: Collects and displays ALL system activities statistics
  - sadc: The sar back-end tool that does the data collection.
  - sa1: A script that runs sadc and stores system activities in a binary data file. sa1 runs from cron.
  - sa2: Creates daily summary of the collected statistics. sa2 runs from cron.
  - pidstat: Reports statistics based on the process ID (PID)
  - cifsiostat: Generates CIFS statistics
- Many options exist for sar:
  - -A, -r, -b, -B, -d, -S, and more
- You can specify *interval* and *count* parameters.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The iostat and mpstat commands are provided by the sysstat package. Additional resource monitoring tools, including sar and sadc (system activity data collector), are also provided by this package. See <http://sebastien.godard.pagesperso-orange.fr/> for more information on sysstat. The following is a partial list of the files provided by the package:

```
# rpm -ql sysstat
/etc/cron.d/sysstat                                /usr/bin/sadf
/etc/rc.d/init.d/sysstat                            /usr/bin/sar
/etc/sysconfig/sysstat                             /usr/lib64/sa
/etc/sysconfig/sysstat.ioconf                      /usr/lib64/sa/sa1
/usr/bin/cifsiostat                               /usr/lib64/sa/sa2
/usr/bin/iostat                                    /usr/lib64/sa/sadc
/usr/bin/mpstat                                    /var/log/sa
/usr/bin/pidstat
```

The sadc command collects system resource utilization data and writes it to a file. The sadc command is normally run by the sa1 script, which is invoked by cron via the /etc/cron.d/sysstat file. By default, cron runs the sa1 script every 10 minutes.

The `sar` command produces system utilization reports based on the data collected by `sadc`. The `sar` command is normally run by the `sa2` script, which is also invoked by `cron` via the `/etc/cron.d/sysstat` file. By default, `cron` runs the `sa2` script once a day at 23:53, allowing it to produce a report for the entire day's data. Example:

```
# cat /etc/cron.d/sysstat
*/10 * * * * root /usr/lib64/sa/sa1 ...
53 23 * * * root /usr/lib64/sa/sa2 ...
```

The `sa1` script logs output into `sysstat` binary log file format, and the `sa2` script reports it back in human-readable format. By default, the data is written to files in the `/var/log/sa` directory. The files are named `sa<dd>`, where `<dd>` is the current day's two-digit date. Running the `sar` command without any options uses the current daily data file as the data source. Use the `-f` filename option to specify a different data source. Sample output from `sar` is shown here:

```
# sar
Linux 3.8.13-16.2.1.el6uek.x86_64 (host03.example.com) ... (1 CPU)
11:00:01 AM CPU %user %nice %system %iowait %steal %idle
11:10:01 AM all 0.01 0.00 0.01 0.09 0.00 99.89
11:20:01 AM all 0.02 0.00 0.09 0.11 0.00 99.79
11:30:01 AM all 0.08 0.00 0.11 0.32 0.00 99.49
11:40:01 AM all 0.01 3.69 7.41 6.68 0.00 82.21
Average: all ...
```

Many options exist for the `sar` command including the following:

- `-A`: Display all the statistics saved in the current daily data file.
- `-r`: Display memory utilization statistics.
- `-b`: Report I/O and transfer rate statistics.
- `-B`: Report paging statistics.
- `-d`: Report activity for each block device.
- `-s`: Report swap statistics.

The `sar` command also accepts `interval` and `count` parameters. If the `interval` parameter is set to zero, `sar` displays the average statistics for the time since the system was started. Reports are generated continuously if the `interval` parameter is specified without the `count` parameter.

## top Utility

- The `top` utility monitors system processes in real time.
- The upper section of the `top` output displays load averages, number of running and sleeping tasks, and overall CPU and memory usage.
- The lower section has a sorted list of processes, owner, running time, and CPU and memory usage.
- `top` sorts the list by most CPU-intensive tasks, and refreshes the list every three seconds by default.
- `top` provides an interactive interface for manipulating processes:
  - `h` or `?`: Display the help screen.
  - `f`: Select the columns to display.
  - `F` or `O`: Select the sort field.
  - `q`: Quit.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `top` command provides an ongoing look at processor activity in real time. It displays a list of the most CPU-intensive processes or tasks on the system and provides a limited interactive interface for manipulating processes. The following is a partial example of the `top` output:

```
# top
top - 03:55:32 up 21 days, 21:11 3 users, load average: ...
Tasks: 151 total, 1 running, 149 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.3%us, 0.0%ni, 99.7%id, 0.0%wa, 0.0%hi, 0.0%si...
Mem: 2056632k total, 1339688k used, 716944k free, 204764k buf...
Swap: 4286460k total, 0k used, 4286460k free, 759712k ca...
PID  USER  PR NI  VIRT  RES  SHR S %CPU %MEM   TIME+  COMMAND
1744  root  20  0 125m  23m 7872 S  0.3    0.2   1:57:94  Xorg
      1  root  20  0 19448 1556 1244 S  0.0    0.1   0:03:54  init
16437 oracle 20  0 282m 13m 9408 S  0.3    0.1   0:24:55  gnom...
```

This sample listing is a point-in-time view of the output that `top` produces. The output is dynamic and refreshes every 3 seconds by default.

The output is divided into two main sections. The upper section displays general information such as the load averages during the last 1, 5, and 15 minutes (same output as the `uptime` command), number of running and sleeping tasks, and overall CPU and memory usage. The following keys change the output displayed in the upper section:

- **l**: Toggles load average and uptime on and off
- **m**: Toggles memory and swap usage on and off
- **t**: Toggles tasks and CPU states on and off

The lower section displays a sorted list of processes (usually by CPU usage) and their PIDs (process ID number), the user who owns the process, running time, and CPU and memory that the processes uses. The following describes the columns in the lower section:

- **PID**: Task's unique process ID
- **USER**: Effective username of the task's owner
- **PR**: Priority of the task
- **NI**: Nice value of the task. A negative value means higher priority, a positive value means lower priority. Zero in this field means priority is not adjusted in determining a task's dispatchability.
- **VIRT**: Total amount of virtual memory used by the task. It includes all code, data, and shared libraries, plus pages that have been swapped out.
- **RES**: Non-swapped physical memory (resident size) a task has used
- **SHR**: Amount of shared memory used by a task. This memory could potentially be shared with other processes.
- **S**: Status of the task, which can be one of: D (uninterruptible sleep), R (running), S (sleeping), T (traced or stopped), or Z (zombie)
- **%CPU**: Task's share of the elapsed CPU time (CPU usage) since the last screen update, expressed as a percentage of total CPU time
- **%MEM**: Task's currently used share of available physical memory (memory usage)
- **TIME+**: Total CPU time that the task has used since it started
- **COMMAND**: Command-line or program name used to start a task

There are several keystroke commands that can be used while `top` is running. The following is a partial list:

- **h or ?**: Displays a list of available commands (help screen)
- **f**: Allows you to select different columns to display
- **o**: Allows you to change the order of the columns
- **F or o**: Allows you to select the sort field. CPU usage is the default.
- **d or s**: Allows you to change the refresh interval
- **c**: Toggles the display of command-line and program name
- **i**: Toggles the display of all tasks or just active tasks
- **s**: Toggles the cumulative time on and off. When on, each process is listed with the CPU time that it and its dead children have used. When off, programs that fork into many separate tasks appear less demanding.
- **u**: Allows you to display only those tasks owned by a specific user
- **k**: Allows you to kill a process
- **q**: Allows you to exit or quit the `top` utility

# iotop Utility

The screenshot shows a terminal window titled "root@host03:~" displaying the output of the iotop command. The output lists 22 processes, each with its TID, PRIO, USER, DISK READ, DISK WRITE, SWAPIN, IO, and COMMAND. Most processes show 0.00 B/s for both disk read and write, indicating low disk activity.

Total DISK READ: 0.00 B/s   Total DISK WRITE: 0.00 B/s							
TID	PRIO	USER	DISK READ	DISK WRITE	SWAPIN	IO>	COMMAND
1024	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ext4-dio-unwrit]
1	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	init
2	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kthreadd]
3	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ksoftirqd/0]
5	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kworker/0:0H]
15685	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	packagekitd
7	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kworker/u:0H]
8	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/0]
9	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_bh]
10	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_sched]
11	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[watchdog/0]
12	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[cpuset]
13	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[khelper]
14	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kdevtmpfs]
15	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[netns]
16	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[xenwatch]
17	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[xenbus]
18	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[bdi-default]
19	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kintegrityd]
20	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kblockd]
21	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[xenbus_frontend]
22	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ata_sff]

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `iotop` command is a Python program. `iotop` has a user interface similar to `top`, but it is used for monitoring swap and disk I/O on a per-process basis. If you are getting more disk activity on your system than you would like, `iotop` can help identify which process or processes are responsible for the excessive I/O.

The `iotop` command requires a kernel version 2.6.20 or higher and Python 2.5 or higher. Run `uname -r` to obtain your kernel version and `python -v` to get the Python version.

The top of the output displays the sum of the DISK READ and DISK WRITE bandwidth in B/s (bytes per second). After this is a list of all processes running on the system. Each process has a column labeled DISK READ and DISK WRITE, as well as SWAPIN and IO. The COMMAND column displays the name of the process.

By default `iotop` monitors all users on the system and all processes. Several options are available. The following is a partial list of options to `iotop`:

- `-h`: Display help and a list of options.
- `-o`: Show only processes and threads actually doing I/O.
- `-u USER`: Show specific *USER* processes.
- `-a`: Show accumulated I/O instead of bandwidth.

Press the letter `q` to exit.

## strace Utility

- The strace utility is a debugging tool.
- It prints the system calls made by another program or process.
- Each line contains the system call name, followed by its arguments in parentheses and its return value.
- Errors typically return a value of -1 and have the errno symbol and error string appended.
- Signals are printed as a signal symbol and a signal string.
- Output is printed on standard error or to the file specified with the -o option.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The strace command is a debugging tool, which prints a list of all the system calls made by another program or process. It displays the system calls that are called by a process and the signals that are received by a process. It is particularly useful in determining why a program continually crashes or does not behave as expected.

Each line in the trace contains the system call name, followed by its arguments in parentheses and its return value. Following is a partial output from stracing the ls command:

```
# strace ls
execve("/bin/ls", ["ls"], /* 29 vars */) = 0
brk(0)...
mmap(NULL, 4096 ...
access("/etc/ld.so.preload", R_OK)...
open("/etc/ld.so.cache", O_RDONLY)...
...
...
```

Errors typically return a value of -1 and have the errno symbol and error string appended. Signals are printed as a signal symbol and a signal string. Output is printed on standard error or to the file specified with the -o option.

## netstat Utility

- The `netstat` utility displays various network-related information.
- The `netstat` command without options displays a list of open sockets for each address family (AF).
- Several options exist:
  - `-A`: Specify the address family.
  - `-r`: Display the route table.
  - `-i`: Display network interface information.
  - `-s`: Display summary statistics for each protocol.
  - `-g`: Display multicast group membership information.
  - `-n`: Display IP addresses instead of the resolved names.
  - `-c`: Print information every second continuously.
  - `-e`: Display additional information.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `netstat` command displays current TCP/IP network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. The `ss` command provides dump socket statistics but also shows information similar to `netstat`.

A number of command-line options and arguments exist, but `netstat` by itself displays a list of open sockets. Sockets are the interface between the user process and the network protocol stacks in the kernel. The protocol modules are grouped into protocol families such as `AF_INET`, `AF_IPX`, and `AF_PACKET`, and socket types such as `SOCK_STREAM` or `SOCK_DGRAM`. If you do not specify any address families, the active sockets of all configured address families are printed.

To specify the address families (low-level protocols) for which connections are to be shown, use the `-A` option followed by a comma-separated list of address family keywords. Possible address family keywords are `inet`, `unix`, `ipx`, `ax25`, `netrom`, and `ddp`. Example:

```
# netstat -A unix
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags Type      State      I-Node Path
unix    2          [ ]     DGRAM           7137     @/org/kernel/udev/udevd
...
...
```

Some of the other options for `netstat` are listed:

- **`-r` or `--route`:** Display the kernel routing table:

```
# netstat -r
Destination Gateway   Genmask   Flags MSS Window irtt Iface
default      192.0.2.1 0.0.0.0  UG        0      0      0 eth0
...
• -i or -I=iface: Display a table of all network interfaces or the specified iface:
```

```
# netstat -I=eth0
Iface  MTU Met     RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-DRP TX-OV...
eth0   1500 0 1131204      0      16      0 174989      0 0...
```

- **`-s` or `--statistics`:** Display summary statistics for each protocol:

```
# netstat -s
Ip:
    106564 total packets received
    0 forwarded
    0 incoming packets discarded
    10427 incoming packets delivered
    106069 requests sent out
```

Icmp:

...

- **`-l` or `--listening`:** Display all ports that have a process currently listening for input.

```
# netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address      Foreign Address      State
tcp        0      0 *:pop3s            *:*                  LISTEN
...

```

- **`-g` or `--groups`:** Display multicast group membership information for IPv4 and IPv6:

```
# netstat -g
Interface RefCnt Group
-----
lo          1      224.0.0.1
eth0        1      224.0.0.251
...

```

- **`-n` or `--numeric`:** Display IP addresses instead of the resolved names.
- **`-c` or `--continuous`:** Print information every second continuously.
- **`-e` or `--extend`:** Display additional information. Use this option twice for maximum detail.
- **`-p` or `--program`:** Show the PID and name of the program to which each socket belongs.

Any invalid option or argument displays a help screen listing usage and a brief description of available options.

## tcpdump Utility

- The `tcpdump` utility is a packet-capture utility for network troubleshooting.
- Traffic is captured based on a specified filter.
- A variety of options exist, including:
  - `-D`: Print a list of network interfaces.
  - `-i`: Specify an interface on which to capture.
  - `-c`: Specify the number of packets to receive.
  - `-v, -vv, -vvv`: Increase the level of detail (verbosity).
  - `-w`: Write captured data to a file.
  - `-r`: Read captured data from a file.
- You can also specify host, source, or destination of traffic, and a specific protocol to capture.
- Boolean operators (AND, OR, NOT) allow complex filters.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The `tcpdump` utility allows you to capture packets that flow within your network to assist in network troubleshooting. The following are several examples of using `tcpdump` with different options.

To print a list of network interfaces available on which `tcpdump` can capture packets:

```
# tcpdump -D
1.eth0
2.eth1
3.any (Pseudo-device that captures on all interfaces)
4.lo
```

For each network interface, a number and an interface name is printed. The interface name or the number can be supplied to the `-i` flag to specify an interface on which to capture.

```
# tcpdump -i 1
listening on eth0, link-type EN10MB (Ethernet), capture size...
03:57:25.845920 ARP, Request who-has host02.example.com tell...
03:57:25.846093 ARP, Reply host02.example.com is-at 00:16:3e...
```

In this example, output is continuous until terminated by pressing Ctrl + C.

To exit `tcpdump` after receiving a specific number of packets, use the `-c` (count) option followed by the number of packets to receive. The following example captures two packets:

```
# tcpdump -i 1 -c2
...
2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

As shown in this example, when `tcpdump` finishes capturing packets, it reports the following:

- **packets captured:** This is the number of packets that `tcpdump` has received and processed.
- **packets received by filter:** A filter can be specified on the command line and only those packets that match the defined filter are processed by `tcpdump` and counted.
- **packets dropped by kernel:** This is the number of packets that were dropped due to a lack of buffer space. Use the `-B` option to set the buffer size.

To increase the detail (verbosity) of the output, use the `-v` option, or `-vv` for even more verbose output, or `-vvv` for the most verbose level of output:

```
# tcpdump -i 1 -v
# tcpdump -i 1 -vv
# tcpdump -i 1 -vvv
```

Using the `tcpdump` utility with the `-w` option allows you to write captured data to a file. This allows the captured data to be read by other network analysis tools, such as Wireshark. The following example captures data to a file named `capture_file`:

```
# tcpdump -i 1 -v -c2 -w capture_file
```

You can also read captured data from a file by using the `-r` option:

```
# tcpdump -r capture_file
```

Many other options and arguments can be used with `tcpdump`. The following are some specific examples of the power of the `tcpdump` utility.

To display all traffic between two hosts (represented by variables `host1` and `host2`):

```
# tcpdump host host1 and host2
```

To display traffic from only a source (`src`) or destination (`dst`) host:

```
# tcpdump src host
# tcpdump dst host
```

Provide the protocol as an argument to display only traffic for a specific protocol, for example `tcp`, `udp`, `icmp`, `arp`:

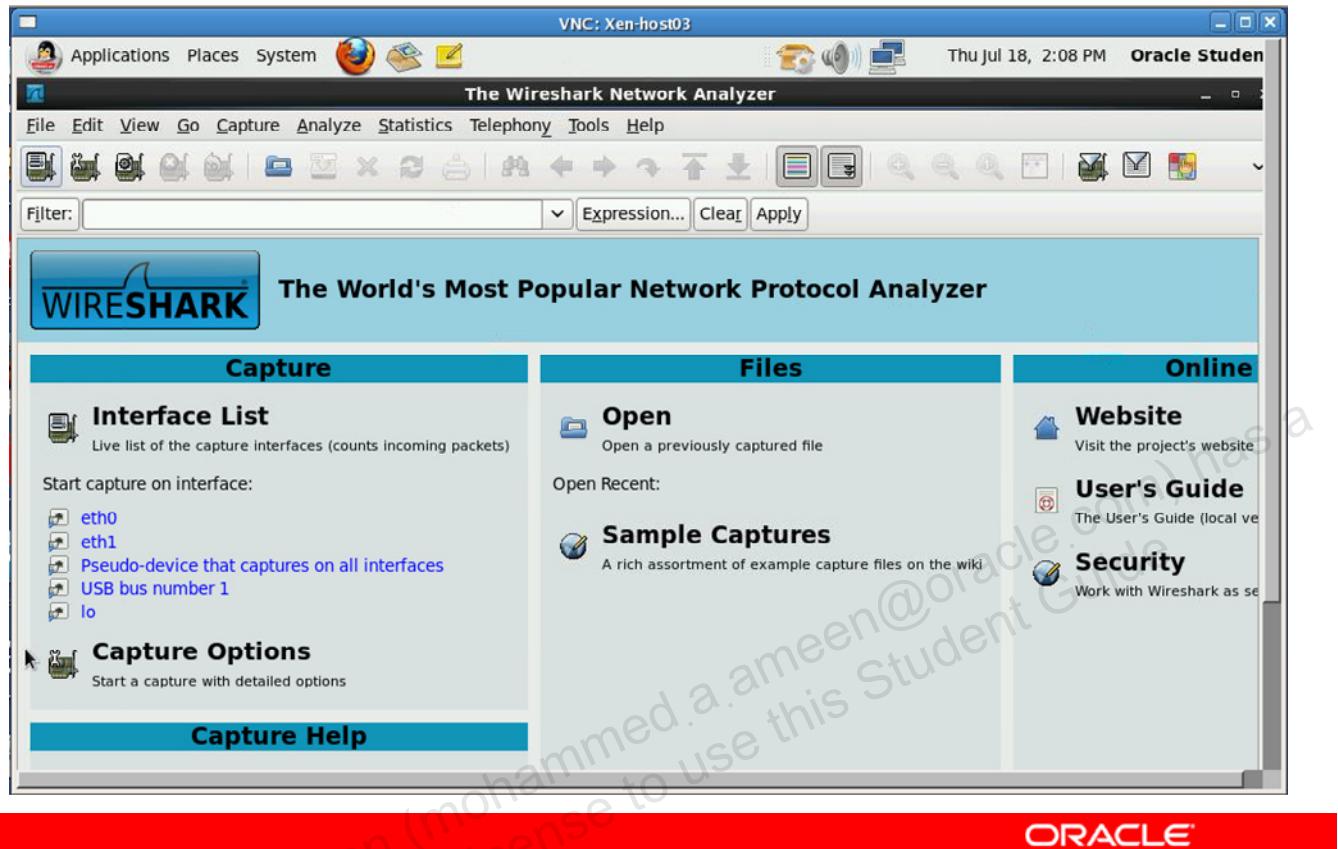
```
# tcpdump protocol
```

To filter based on a source or destination port:

```
# tcpdump src port ftp
# tcpdump dst port http
```

The `tcpdump` utility also accepts Boolean operators (AND, NOT, OR) and grouping of operators, allowing you to create complex filters for capturing network data.

# Wireshark



The slide shows the Wireshark GUI. Wireshark is a network protocol analyzer that allows you to interactively browse packet data from a live network or from a previously saved capture file. The GUI is provided by the `wireshark-gnome` RPM but you also need to install the same version of the `wireshark` RPM. Documentation for Wireshark is installed in the `/use/share/wireshark` directory.

As indicated on the GUI, you can start a capture from any available network interface. Each live capture can be saved to a file for future analysis. You also open a previously captured file for analysis. Various capture options can be selected such as the following:

- Capture packets in promiscuous mode
- Stop the capture after a specified number of packets, bytes, or time period
- Enable MAC name resolution
- Enable network name resolution

You can also filter a capture based on MAC address, IP address, protocol, or create your own filter expression. Wireshark provides packet search capabilities as well as packet coloring rules.

Also included with the Wireshark package is `tshark`, a text-based network protocol analyzer. `tshark` also allows you to capture packet data from a live network, or read packets from a previously saved capture file.

## OSWatcher Black Box (OSWbb)

- OSWbb collects and archives operating system and network metrics to aid in diagnosing performance issues.
- OSWbb includes a built-in analyzer called OSWbba.
- Download the OSWbb TAR file from My Oracle Support (MOS).
- To install OSWbb, use the `tar` command:
  - `# tar xvf oswbb703.tar`
- To start OSWbb, use the following command:
  - `# ./startOSWbb.sh`
- To stop OSWbb, use the following command:
  - `# ./stopOSWbb.sh`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Oracle OSWatcher Black Box (OSWbb) product is a collection of shell scripts intended to collect and archive operating system and network metrics to aid in diagnosing performance issues. OSWbb operates as a set of background processes on the server and gathers data on a regular basis, invoking such UNIX utilities as `vmstat`, `netstat`, `iostat`, `top`, and others.

Beginning with release 4.0.0, OSWbb includes a built-in analyzer called OSWbba, which analyzes the data that OSWbb collects. It provides information on system slowdowns, hangs and other performance problems. It also provides the ability to graph `vmstat` and `iostat` data.

OSWbb is particularly useful for Oracle Real Application Clusters (RAC) and Oracle Grid Infrastructure configurations. OSWbb is included in the RAC-DDT (Diagnostic Data Tool) script file, but is not installed by RAC-DDT.

### Installing OSWbb

You must install OSWbb on each node where data is to be collected. For RAC or shared disk systems, each node requires an OSWbb installation into a unique directory (for example, `/oswbb_node1` and `/oswbb_node2`). OSWbb is available through MOS and can be downloaded as a TAR file named `oswbb703.tar`. After downloading the TAR file, copy the file to the directory where OSWbb is to be installed and run the following command:

```
# tar xvf oswbb703.tar
```

Extracting the TAR file creates a directory named `oswbb`, which contains all the files associated with OSWbb.

```
# ls oswbb
analysis/      htop.sh        OSWatcherFM.sh      oswlnxio.sh
profile/       stopOSWbb.sh   vmsub.sh          docs/
iosub.sh       OSWatcher.sh   oswnet.sh          pssub.sh
tarupfiles.sh  xtop.sh        Exampleprivate.net locks/
oswbba.jar     oswrds.sh      src/               tmp/
gif/           mpsub.sh      oswib.sh          oswsub.sh
StartOSWbb.sh  topaix.sh
```

## Starting OSWbb

To start the OSWbb utility, execute the `startOSWbb.sh` shell script. The `startOSWbb.sh` script accepts two optional arguments that control the frequency (in seconds) that data is collected and the number of hours worth of data to archive. If you do not enter any arguments, the script runs with default values of 30 and 48, meaning collect data every 30 seconds and store the last 48 hours of data in archive files.

The following example starts the tool and collects data at 60-second intervals and logs the last 10 hours of data to archive files. Some of the output produced when starting the tool is shown:

```
# ./startOSWbb.sh 60 10
Testing for discovery of OS Utilities...
VMSTAT found on your system.
IOSTAT found on your system.
MPSTAT found on your system.
NETSTAT found on your system.
TOP found on your system.
Testing for discovery of OS CPU COUNT
...
Starting Data Collection...
oswbb heartbeat: date/time
oswbb heartbeat: date/time (60 seconds later)
...
```

## Stopping OSWbb

To stop OSWbb, execute the `stopOSWbb.sh` shell script. This terminates all processes associated with OSWbb and is the normal, graceful mechanism for stopping the tool.

## OSWbb Diagnostic Data Output

- OSWatcher.sh is the main controlling script that spawns other scripts to collect diagnostic data.
- The data is stored in hourly archive files:
  - <node\_name>\_<OS\_utility>\_YY.MM.DD.HH24.dat
- Subdirectories are created in the archive directory.
- oswiostat: Contains the output from the iostat utility
- oswmeminfo: The contents of the /proc/meminfo file
- oswmpstat: Contains the output from the mpstat utility
- oswnetstat: Contains the output from the netstat utility
- oswprvtnet: Contains the status of RAC private networks
  - Requires you to manually create an executable file named private.net, which runs traceroute commands



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The OSWatcher.sh shell script is the main controlling script that spawns individual shell processes to collect specific kinds of data by using UNIX operating system diagnostic utilities. Control is passed to individually spawned operating system data collector processes, which in turn collect specific data, time-stamp the data output, and append the data to files.

Data collectors exist for the top, vmstat, iostat, mpstat, netstat, and ps utilities, and for /proc/meminfo and /proc/slabinf0. There is also an optional collector for tracing private networks. The collected data files are stored in the archive subdirectory, which is created when OSWbb is started for the first time. The archive directory contains nine subdirectories, one for each data collector.

```
# ls archive
oswiostat/  oswmeminfo/    oswmpstat/   oswnetstat/   oswprvtnet/
oswps/       oswslabinfo/   oswtop/      oswvmstat/
```

The data is stored in hourly archive files during the time that OSWbb is running. Files are named using the following format:

<node\_name>\_<OS\_utility>\_YY.MM.DD.HH24.dat

Each entry in the file contains a time stamp prefixed by \*\*\* characters. The contents of each of the nine archive directories are described here:

**oswiostat**

OSWbb runs the `iostat` utility at the specified interval and stores the data in this directory. The default options and arguments for `iostat` are set in the `oswlnxio.sh` script. By default, `iostat` produces extended output (-x option). Look for average service times, `svctm`, greater than 20 msec for long durations and high average wait times, `await`, as indicators of performance problems.

**oswmeminfo**

OSWbb reads the `/proc/meminfo` file at the specified interval and stores the data in this directory. Information about available memory, `MemTotal`, and swap, `SwapTotal`, is included in this file.

**oswmpstat**

OSWbb runs the `mpstat` utility at the specified interval and stores the data in this directory. Be aware of involuntary context switches and the number of times a CPU failed to obtain a mutex. Values consistently greater than 200 per CPU cause system time to increase.

**oswnetstat**

OSWbb runs the `netstat` utility at the specified interval and stores the data in this directory. Each protocol type has a specific set of measures associated with it. Network analysis requires evaluation of these measurements on an individual level and all together to examine the overall health of the network communications.

The information in the upper section of the report helps diagnose network problems when there is connectivity but response is slow. The lower section of the report contains protocol statistics. The TCP protocol is used more often than UDP in Oracle database and applications. Many performance problems associated with the network involve the retransmission of the TCP packets. Some implementations for RAC use UDP for the interconnect protocol, instead of TCP. The statistics cannot be divided up on a per-interface basis, so these should be compared to the interface statistics in the upper portion of the report.

**oswprvtnet**

Information about the status of RAC private networks is collected and stored in this directory only if you have configured private network tracing. This requires you to manually add entries for these private networks into an executable file named `private.net` located in the `oswbb` directory.

An example of what this file should look like is named `Exampleprivate.net` with samples for each operating system: `solaris`, `linux`, `aix`, `hp`, and so on, in the `oswbb` directory. This file can be edited and renamed `private.net` or a new file named `private.net` can be created. This file contains entries for running the `traceroute` command to verify RAC private networks. The following is an example of a `private.net` entry on Linux:

```
traceroute -r -F node1  
traceroute -r -F node2
```

In this example, `node1` and `node2` are two nodes in addition to the `hostnode` of a three-node RAC cluster. If the `private.net` file does not exist or is not executable, then no data is collected and stored under the `oswprvtnet` directory. Review the collected data to ensure that the network interface is up and responding and that the network is reachable. If `traceroute` indicates that the target interface is not on a directly connected network, validate that the address is correct or the switch it is plugged in to is on the same VLAN.

## OSWbb Diagnostic Data Output

- **oswps:** Contains the output from the `ps` utility
- **oswslabinfo:** Contents of the `/proc/slabinfo` file
  - Contains statistics on the kernel slab cache
- **oswttop:** Contains the output from the `top` utility
- **oswvmstat:** Contains the output from the `vmstat` utility

**ORACLE**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The remaining data collection directories are described.

### **oswps**

OSWbb runs the `ps` command at the specified interval and stores the data in this directory. The `ps` command lists all the processes currently running on the system and provides information about CPU consumption, process state, priority of the process, and other information. OSWbb runs the command with the `-elf` option.

The information in the `ps` command is helpful supporting information for RAC diagnostics. For example, the status of a process before a system crash may be important for root cause analysis. To discover the amount of memory that a process consumes is another example of how this data can be used.

### **oswslabinfo**

OSWbb reads the `/proc/slabinfo` file at the specified interval and stores the data in this directory. Frequently used objects in the Linux kernel have their own cache. This file gives statistics on the kernel slab cache.

For each slab cache entry, the file includes the cache name, the number of currently active objects (memory blocks), the total number of available objects, the size of each object in bytes, the number of pages with at least one active object, the total number of allocated pages, and the number of pages per slab.

### **oswtop**

OSWbb runs the `top` utility at the specified interval and stores the data in this directory.

The load average line displays the load averages over the last 1, 5, and 15 minutes. Load average is defined as the average number of processes in the run queue. A runnable UNIX process is one that is available right now to consume CPU resources and is not blocked on I/O or on a system call. The higher the load average, the more work your machine is doing.

The three numbers are the average of the depth of the run queue over the last 1, 5, and 15 minutes. It is important to determine what the average load of the system is through benchmarking and then look for deviations. A dramatic rise in the load average can indicate a serious performance problem.

The tasks line displays the total number of processes running at the time of the last update. It also indicates how many processes exist, how many are sleeping (blocked on I/O or a system call), how many are stopped (someone in a shell has suspended it), and how many are actually assigned to a CPU. Like load average, the total number of processes on a healthy machine usually varies just a small amount over time. Suddenly having a significantly larger or smaller number of processes could be a warning sign.

The memory line reflects how much real and swap memory your system has, and how much is free. Real memory is the amount of RAM installed in the system, or the physical memory. Swap is virtual memory stored on the machine's disk. Performance deteriorates when a computer runs out of physical memory and starts using swap space.

Look for a large run queue. A large number of processes waiting in the run queue may be an indication that your system does not have sufficient CPU capacity. Also look for processes that are consuming lots of CPU, these processes can possibly be tuned.

### **osvvmstat**

OSWbb runs the `vmstat` utility at the specified interval and stores the data in this directory.

Again, when trying to determine the cause of performance problems, a large run queue can indicate CPU saturation. Also look at CPU usage to determine whether more CPUs are required. Memory bottlenecks are determined by the scan rate. If this rate is continuously high, then there is a memory shortage. Disk problems may exist if the number of processes blocked exceeds the number of processes on the run queue.

## OSWatcher Black Box Analyzer (OSWbba)

- OSWbba is a graphing and analysis utility that is included with OSWbb v4.0.0 and higher.
- OSWbba graphically displays data collected, and generates reports.
- OSWbba includes a built-in analyzer to provide details on performance problems.
  - The ability to create a graph and analyze this information relieves you of manually inspecting all the files.
- To start OSWbba, use the following command:

```
# java -jar oswbba.jar -i ~/oswbb/archive
```
- The OSWbba menu provides options to graph and analyze the collected data.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

OSWatcher Black Box Analyzer (OSWbba) is a graphing and analysis utility that comes bundled with OSWbb v4.0.0 and higher. OSWbba allows you to graphically display the data that is collected, to generate reports containing these graphs, and provides a built-in analyzer to analyze the data and provide details on any performance problems that it detects. The ability to create a graph and analyze this information relieves you of manually inspecting all the files.

OSWbba replaces the OSWg utility. This was done to eliminate the confusion caused by having multiple tools in support named OSWatcher. OSWbba is supported only for data collected by OSWbb and no other tool.

OSWbba is written in Java and requires a minimum of Java Version 1.4.2 or higher. OSWbba can run on any UNIX X Windows or PC Windows platform. OSWbba uses Oracle Chartbuilder, which requires an X Windows environment.

OSWbba parses all the OSWbb `vmstat`, `iostat`, and `top` utility log files contained in the `oswbb/archive` directory. When the data is parsed, you are presented with a command-line menu that has options for displaying graphs, creating binary GIF files of these graphs, and generating an HTML report containing all the graphs with a narrative on what to look for, and the ability to self-analyze the files that OSWbb creates.

OSWbba requires no installation. It comes shipped as a stand-alone Java JAR file with OSWbb v4.0.0 and higher.

## Starting OSWbba

Before starting the OSWbba utility, run the following command to ensure that you have Java Version 1.4.2 or higher installed on your system. In this example, the version is 1.7.0\_45:

```
# java -version  
java version "1.7.0_45"  
...
```

OSWbba requires an input directory to run. This input directory is the fully qualified path name of the archive directory containing the OSWbb logs. The archive directory must have the same directory structure as the archive directory for OSWbb. It must contain the subdirectories; oswvmstat, oswiostat, oswps, oswtop, and oswnetstat. Use the -i <archive\_directory> option to specify the input directory:

```
# java -jar oswbba.jar -i ~/oswbb/archive  
Starting OSW Black Box Analyzer V7.0.3  
...  
Parsing Data. Please Wait...  
Parsing Completed.
```

After the parsing completes, the following menu is displayed, providing options to create a graph and analyze the collected data:

```
Enter 1 to Display CPU Process Queue Graphs  
Enter 2 to Display CPU Utilization Graphs  
Enter 3 to Display CPU Other Graphs  
Enter 4 to Display Memory Graphs  
Enter 5 to Display Disk IO Graphs  
Enter 6 to Generate All CPU Gif Files  
Enter 7 to Generate All Memory Gif Files  
Enter 8 to Generate All Disk Gif Files  
Enter L to Specify Alternate Location of Gif Directory  
Enter T to Specify Different Time Scale  
Enter D to Return to Default Time scale  
Enter R to Remove Currently Displayed Graphs  
Enter A to Analyze Data  
Enter S to Analyze Subset of Data (Changes analysis dataset ...)  
Enter P to Generate A Profile  
Enter X to Export Parsed Data to File  
Enter Q to Quit Program
```

Please Select an Option:

The first three options display graphs of specific CPU components of `vmstat`. All options are described as follows:

- Option 1 – Displays the process run, wait, and block queues
- Option 2 – Displays CPU utilization graphs for system, user, and idle
- Option 3 – Displays graphs for context switches and interrupts
- Option 4 – Displays memory graphs for free memory and available swap
- Option 5 – Uses the extended disk statistics option of `iostat` to display a list of all devices. The device name along with the average service time of each device is listed. You can then select one of the devices from the list. Graphs are available for reads/second, writes/second, service time, and percent busy. Example:

The Following Devices and Average Service Times Are Ready to Display:

Device Name	Average Service Times in Milliseconds
xvda	0.03258620689655172
scd0	
xvdb	
xvdd	

Specify A Case Sensitive Device name to View (Q to exit) :

- Options 6, 7, 8 – Generate images of the graph for the specific category (CPU, memory, disk) to a file. The file is created in the OSWbba directory by default.
- Option L – Allows you to specify an alternative location for the image files that you create using options 6, 7, and 8
- Option T – Allows you to specify a different subset of time to graph. The default time span is based on the entire time span of the logs. For example, if OSWbb keeps the last 48 hours of logs in the archive, the default graph contains all 48 hours of data. You can specify to graph a two-hour period, for example, out of the entire 48-hour collection.
- Option D – Resets the graphing time scale back to the time encompassing the entire log collection
- Option R – Removes all previously displayed graphs from the screen
- Option A – Analyzes the files in the archive and produces a report
- Option S – Analyzes a subset of data
- Option P – Generates an HTML profile
- Option X – Exports parsed data to a file
- Option Q – Exits the program

## Analyze OSWbb Archive Files

- Start the analyzer from the OSWBBA directory.
- Select Option A from the OSWbba menu.
- You can also run the analyzer from the command line:

```
# java -jar oswbba.jar -i ~/oswbb/archive -A
```

- The analyzer output is divided into eight sections:
  - Section 1: System Status
  - Section 2: System Slowdown
  - Section 3: System General Findings
  - Section 4: CPU Detailed Findings
  - Section 5: Memory Detailed Findings
  - Section 6: Disk Detailed Findings
  - Section 7: Network Detailed Findings
  - Section 8: Process Detailed Findings



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Select Option A from the OSWbba menu to analyze the files in the archive directory and produce a report. You need to be in the directory where OSWbba is installed to run the analyzer. You can also run the analyzer directory from the command line by including the -A option:

```
# java -jar oswbba.jar -i ~/oswbb/archive -A
A new analysis file analysis/host...txt has been created.
```

The following is a sample analysis file name:

```
# ls ~/oswbb/analysis
host...txt
```

The analyzer output is divided into sections for easy readability.

- |                                       |                                      |
|---------------------------------------|--------------------------------------|
| • Section 1: System Status            | Section 7: Network Detailed Findings |
| • Section 2: System Slowdown          | Section 8: Process Detailed Findings |
| • Section 3: System General Findings  |                                      |
| • Section 4: CPU Detailed Findings    |                                      |
| • Section 5: Memory Detailed Findings |                                      |
| • Section 6: Disk Detailed Findings   |                                      |

Section 1 provides a quick status of each major subsystem. Example:

#### Section 1: System Status

...

Subsystem	Status
CPU	OK
MEMORY	OK
I/O	OK
NET	OK

Other possible status values are Warning, Critical, and Unknown.

Section 2 provides a system slowdown summary ordered by impact. This section lists:

- Slowdown time and duration
- Most likely causes of slowdown
- Offending process
- Advice on what to do

Section 3 provides system general findings such as the following:

- CPU run queue observed very high spikes.
- Severe memory swapping was observed.

Section 4 provides a summary of CPU metrics collected in the archive. The following metrics are reported:

- Number of snapshots in the archive
- Number of snapshots with a high CPU run queue
- Times when the run queue was reported high
- root processes with high CPU consumption
- oracle processes with high CPU consumption

Section 5 provides a summary of memory metrics collected in the archive. The following metrics are reported:

- Process swap queue
- Scan rate
- Snapshot times when scan rate was high

Section 6 provides detailed disk findings. Only devices that are busy more than 50% are included in the report. The following metrics are reported:

- Device percent busy for devices with percent busy > 50%
- Device service time for devices with service time > 10 msec
- Device throughput for devices with percent busy > 50%

Section 7 provides detailed network findings including data link findings, IP findings, UDP findings, and TCP findings.

Section 8 provides detailed process findings ordered by time as well as top processes increasing memory.

# Enterprise Manager Ops Center

## Enterprise Manager Ops Center:

- Provides management services for operating systems, virtual machine, servers, storage, and networks
- Enables you to provision, update (patch), monitor, and manage assets in one or more data centers from a single console
- Includes built-in integration with My Oracle Support, with automatic service request generation
- Has the following architecture components:
  - Enterprise Controller
  - Proxy Controller
  - Agent Controller
  - User Interface
  - Knowledge Base



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle Enterprise Manager Ops Center provides management services for operating systems (including Oracle Linux), virtual machines, servers, storage, and networks. You can provision, update (patch), monitor, and manage the physical and virtual managed assets in one or more of your data centers from a single console, regardless of where the asset is located. It includes built-in integration with My Oracle Support, with automatic service request generation.

### Key Features

The software provides features tailored for administrating the data center infrastructure, including the following:

- **Dashboards:** View assets including a graphical representation of the status and membership.
- **Incident Management:** Monitor assets according to rules and thresholds that you set.
- **Integration with Oracle Enterprise Manager Cloud Control:** View configuration, health and performance information, and incidents of managed assets using either product.
- **Profiles For Assets:** Create software profiles and operational profiles that contain your custom executable scripts.
- **Operational Plans:** Deploy a single script as an operational profile. You can use the scripts to perform specific tasks in your environment, such as configuration options, or to assist in incident management.

- **Deployment Plans:** Combine one or more profiles and scripts to create a multi-task plan that provisions operating systems or firmware efficiently and consistently.
- **Plan Management:** Use the provided default templates, profiles, and plans to create and deploy plans.
- **Hardware Management:** Update system component firmware and track hardware configuration changes over time.
- **Virtualization Management:** Manage virtual assets such as Oracle Solaris Zones, Oracle VM Servers for SPARC, Oracle VM Servers for x86, and their guests.
- **Reports:** Create reports for assets and activities and export the reports as files.

## Architecture

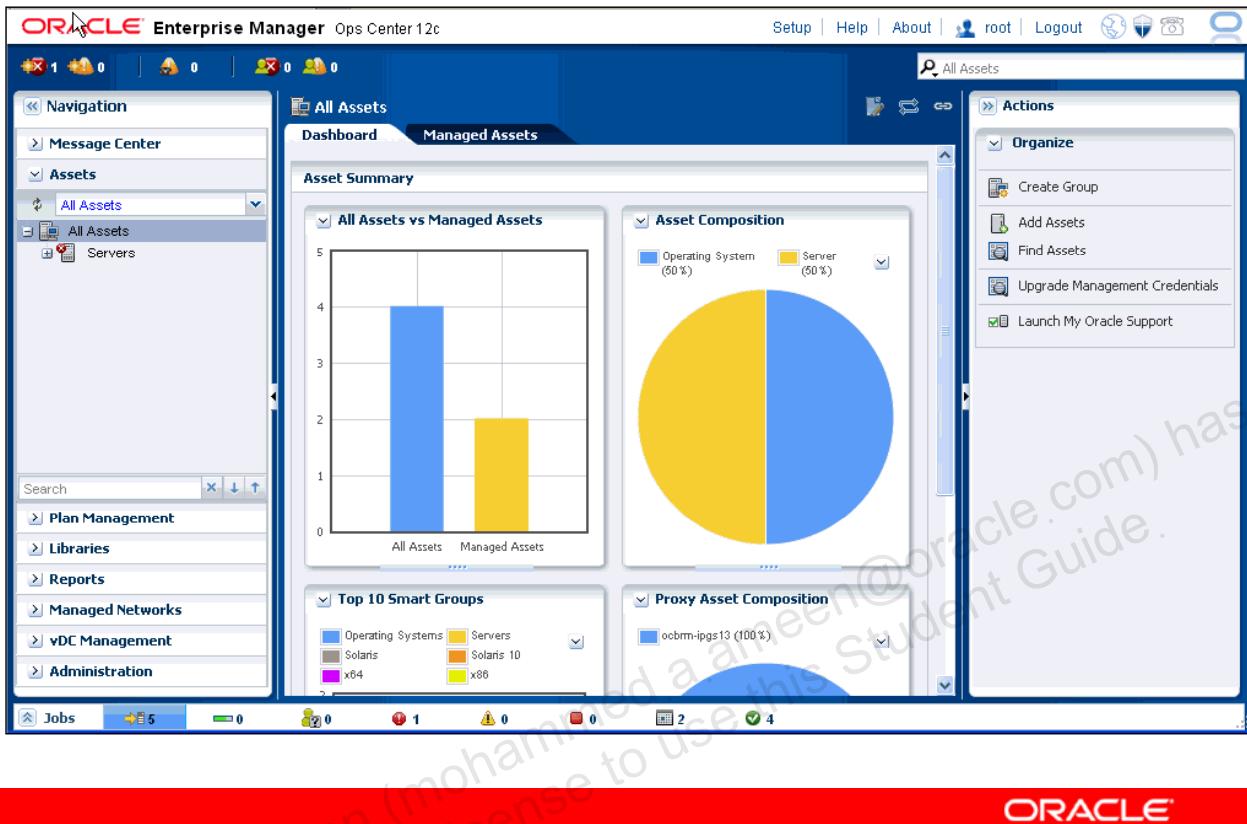
The Enterprise Controller, Proxy Controller, Agent Controller, and user interface are the major architectural components, along with Knowledge Base, which is hosted by Oracle Corporation and accessed through the Internet.

- **Enterprise Controller:** This is the central server for Oracle Enterprise Manager Ops Center. All operations, or jobs, are initiated from the Enterprise Controller. It manages firmware and operating system images, plans, profiles, and policies. It connects to the Internet to get access to contract information, to create service requests, and to download updates. You can also operate the software in Disconnected mode if your site policy does not allow an Internet connection.
- **Proxy Controller:** This distributes the operation load and provides for fan-out capabilities to minimize network load. It links the managed assets to the Enterprise Controller and performs operations that must be located close to the managed assets, such as operating-system provisioning. You can install the Proxy Controller and Enterprise Controller software on the same system, but to enhance performance and scalability, the preferred method is to install the Proxy Controller on a separate machine.
- **Agent Controllers:** The Agent Controller is lightweight Java software that identifies the asset and responds to a Proxy Controller. When an operating system is agent-managed, the agent receives the command from its Proxy Controller, performs the required action, and notifies the Proxy Controller of the results. When an operating system is agentlessly managed, the Proxy Controller uses SSH to perform tasks and to monitor the operating system. You can use many of the monitoring and management features without installing an Agent Controller on the operating system. Hardware management does not require the Agent Controller. Instead, a Proxy Controller runs commands on the hardware system and reports the results to the Enterprise Controller.
- **Knowledge Base and Package Repository:** This stores metadata about Oracle Solaris and Linux operating system components. The metadata includes patch dependencies, standard patch compatibilities, withdrawn patches, and rules for download and deployment. Knowledge Base keeps track of the URLs for the operating systems and retrieves the components from the appropriate vendor download site.

The entire Ops Center product is included as a default part of all Systems support agreements. This means that every customer of Oracle's servers, storage, network equipment, operating systems, and virtualization technology can add Ops Center to their data center management suite.

To see a demonstration of the product, visit <http://www.youtube.com/watch?v=tRWTWDBUIQU>.

# Enterprise Manager Ops Center GUI



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

ORACLE

This slide displays the Oracle Enterprise Manager Ops Center browser interface. The format of the information is in text, tables, graphs, and charts, and the information is organized into hierarchies and tabs. You can view the information and perform actions according to the role that you have been assigned.

The interface consists of five panes:

- **Masthead:** The top pane displays the global functions and information about the Oracle Enterprise Manager Ops Center instance.
- **Navigation Pane:** The left pane consists of several drawers that display assets and objects that are managed by the Oracle Enterprise Manager Ops Center instance.
- **Actions Pane:** The right pane displays the actions that operate on the object currently selected in the Navigation pane. The actions of the Actions pane are enabled or disabled based on the state of the object or your role.
- **Jobs Pane:** The bottom pane displays the number of jobs in Oracle Enterprise Manager Ops Center, categorized by the status of respective jobs.
- **Center Pane:** This pane displays detailed information of the object that is currently selected in the Navigation pane.

To learn more about an incident, place your cursor over the incident icons in the left-side corner of the user interface.

# Enterprise Manager Ops Center Provisioning

- Provisioning Firmware
  - Updates firmware from the Enterprise Controller library to managed servers, chassis, or storage devices
  - Action is controlled by a Firmware Profile
- Provisioning Operating Systems
  - Enables you to install supported operating systems from the software library on the Enterprise Controller onto managed assets
  - Action is controlled by an OS Provisioning Profile
- Applying Deployment Plans
  - Apply profiles in sequence to combine OS provisioning, updates, software installation, script execution and monitoring



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Enterprise Manager Ops Center facilitates automated firmware provisioning and OS provisioning by using a combination of image libraries, profiles, and deployment plans. This allows you to perform consistent installation of an asset by using a deployment plan outlining a combination of an OS Provisioning, OS Update, Software Installation and Update, and post-install scripting, and assigning a monitoring profile.

The images, profiles, and deployment plans are stored on the Enterprise Controller.

Provisioning can be performed on a single asset, or a group of assets.

## Enterprise Manager Ops Center Patching

- Installs, updates, and removes software and patches
- Reduces the complexity of updating a large number of systems
- Automates patching without user interaction
- Automatically manages the patch and software dependencies
- Provides version control and rollback capability
- Supports an update simulation capability
- Action controlled by an OS Update Profile



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Enterprise Manager Ops Center is designed to reduce the complexity of updating a large number of systems, standardize the patch installation process, minimize downtime, track changes, and automate patching without user interaction.

You control the update process, the level of automation, the scheduling, and the number of concurrent updates. You can apply customized controls for one system or a group of systems and schedule the updates to deploy during periods of low usage.

# Enterprise Manager Ops Center Monitoring

- Enterprise Manager Ops Center provides monitoring capabilities for:
  - Hardware
  - Operating Systems
  - Storage Devices
  - Switches
- You can configure thresholds on system-defined parameters to trigger alerts.
  - OS performance statistics
  - Hardware status (temperature, fan speed, voltage, and so on)
  - Power consumption
- Monitoring profiles can be defined and assigned to assets



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The software is designed to make it easy to monitor and manage a large numbers of assets from a single console. It provides end-to-end server awareness and robust monitoring capabilities for the hardware, storage devices, and operating systems in your data center.

You can track system-defined parameters for hardware power consumption, hardware status (temperature, fan speed, and voltage), and key OS statistics (load, CPU, memory.)

For more robust monitoring, the software uses editable rules and event thresholds to monitor your systems. A rule defines a specific monitored resource and the rule parameter defines when an alert is triggered.

# Spacewalk

- Spacewalk is available to manage Oracle Linux systems while transitioning to Oracle Enterprise Manager.
  - The community project can be found at <https://fedorahosted.org/spacewalk/>.
- Documentation is available at <http://linux.oracle.com/documentation/spacewalk/>.
- The RPMs are available at PublicYum.
  - Spacewalk Server for OL6-x86\_64
  - Spacewalk Client for OL6
  - Spacewalk Client for OL5
- Spacewalk Server will only install on OL6-x86\_64.
- Spacewalk Client is available for OL5 and OL6 both i386 and x86\_64 architectures.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Customers migrating from Red Hat Enterprise Linux to Oracle Linux have asked for options to provide a transitional solution for their existing system management tools (such as Red Hat Satellite Server) while evaluating and planning migrations to Oracle Enterprise Manager. Based on this request, Oracle is now offering support for the open-source community project, Spacewalk. Spacewalk is the basis for both Red Hat Satellite Server and SUSE Manager. The community project can be found at <https://fedorahosted.org/spacewalk/>. Spacewalk is similar to Oracle Enterprise Manager in that it allows you to install and update software on your systems, provision systems, and monitor and manage your Oracle Linux systems.

Spacewalk is available to customers with Oracle Linux Basic and Premier Support subscriptions. Oracle Enterprise Manager is still the recommended enterprise solution for managing Oracle Linux systems, but Spacewalk can now be used by customers while they plan and implement their migration to Oracle Enterprise Manager.

Oracle has made a few changes to Spacewalk to ensure easy and complete support for Oracle Linux. RPMs for Spacewalk Server for Oracle Linux 6 x86\_64 architecture and Spacewalk Client for Oracle Linux 5 and Oracle Linux 6 are available at:

- <http://public-yum.oracle.com/repo/OracleLinux/OL6/spacewalk20/server/>
- <http://public-yum.oracle.com/repo/OracleLinux/OL6/spacewalk20/client/>
- <http://public-yum.oracle.com/repo/OracleLinux/OL5/spacewalk20/client/>

Spacewalk documentation is available at <http://linux.oracle.com/documentation/spacewalk>. Refer to the release notes (<http://linux.oracle.com/documentation/spacewalk/relnotes.html>) for Oracle Linux and Oracle Database requirements. Requirements include the following:

- Spacewalk Server provided by Oracle is supported only on Oracle Linux 6 (x86\_64).
- You must remove the `jta` package before installing Spacewalk. Otherwise the Spacewalk services fail to start.
- A database is required to store the Spacewalk data. Oracle supports only Oracle Database for use with Spacewalk. Oracle Database XE and PostgreSQL are not supported.
- The Oracle database must have a user named `spacewalk`.
- The `spacewalk` user must have the `CONNECT` and `RESOURCE` roles.
- The `spacewalk` user must have the `ALTER SESSION`, `CREATE SYNONYM`, `CREATE TABLE`, `CREATE TRIGGER`, `CREATE VIEW`, and `UNLIMITED TABLESPACE` system privileges.
- To connect to an Oracle database, Oracle Instant Client 11.2.0.3 packages must be installed on the Spacewalk Server. The Spacewalk Server configuration fails if these packages are missing.
- When you have installed the Oracle Instant Client, you must add the library path to `ldconfig` as follows:

```
# echo /usr/lib/oracle/11.2/client64/lib >
/etc/ld.so.conf.d/oracle-instantclient11.2.conf
# ldconfig
```
- Use the following command to install Spacewalk:

```
# yum install spacewalk-oracle
```
- When the installation completes, run the Spacewalk configuration tool. Make sure to run the command with the following two arguments:

```
# spacewalk-setup --disconnected --external-db
```
- At the end of the setup script, your Spacewalk Server is fully configured and you can log in to the web portal. Use your favorite browser to connect to the website:  
`http://[spacewalkserverhostname]`

The release notes also include storage requirements, network requirements, and additional setup requirements.

## Quiz

Which of the following utilities allows you to collect system information for sending to Oracle support?

- a. sosreport
- b. sar
- c. OSWbb
- d. strace



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Summary

In this lesson, you should have learned how to:

- Use the `sosreport` utility
- Use the `iostat`, `mpstat`, `vmstat`, `sar`, `top`, `iotop`, and `strace` utilities
- Use the `netstat` and `tcpdump` utilities
- Use the Wireshark network analyzer GUI
- Use the OSWatcher Black Box (OSWbb) tool
- Use OSWatcher Black Box Analyzer (OSWbba)
- Describe Enterprise Manager Ops Center
- Describe Linux Patch and Provisioning using Enterprise Manager Ops Center
- Describe Spacewalk



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 19: Overview

The practices for this lesson cover the following topics:

- Using `sosreport` to collect system information
- Using standard Linux performance monitoring tools
- Installing and using OSWatcher
- Using OSWatcher Black Box Analyzer



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# 20

## System Logging

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe the contents of the `rsyslog` configuration file
- Describe facility/priority-based filters
- Describe `rsyslog` actions
- Describe `rsyslog` templates
- Configure log rotation
- Describe `logwatch`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

# System Log File Configuration

- Logs store system, kernel, service, and application messages.
- `/etc/rsyslog.conf` is the main configuration file.
  - It contains global directives, modules, and rules.
- Global directives specify configuration options for the `rsyslogd` daemon.
- Modules:
  - Provide configuration directives
  - Must be loaded
- Rules define filters and actions:
  - Filter: Selects a subset of `rsyslog` messages
  - Action: Specifies what to do with the selected messages
- Templates are used to modify and format messages.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Log files contain messages about the system, the kernel, services, and applications. Some log files are controlled by the `rsyslogd` daemon. The main configuration file for system logging is `/etc/rsyslog.conf`, which contains global directives, modules, and rules.

## Global Directives

Global directives specify configuration options that apply to the `rsyslogd` daemon. All configuration directives must start with a dollar sign (\$) with only one directive specified per line. The following is an example of a global directive to include all configuration files found in the `/etc/rsyslog.d` directory:

```
$IncludeConfig /etc/rsyslog.d/*.conf
```

A list of all available configuration directives and their descriptions can be found in:

```
/usr/share/doc/rsyslog-<version-number>/rsyslog_conf_global.html
```

## Modules

`rsyslog` has a modular design. This enables functionality to be dynamically loaded from modules. Each module provides configuration directives. Modules must be loaded for their configuration directives and functionality to be available. The syntax to load a module is:

```
$ModLoad <MODULE>
```

Modules are split into the following main categories:

- **Input modules:** Gather messages from various sources. Input module names always start with the `im` prefix (examples: `imfile`, `imrelp`).
- **Output modules:** Provide a facility to store messages into various targets such as sending them across a network, storing them in a database, or encrypting them. Output module names always start with the `om` prefix (examples: `omsnmp`, `omrelp`).
- **Filter modules:** Provide the ability to filter messages according to specified rules. The name of a filter module always starts with the `fm` prefix.
- **Parser modules:** Use the message parsers to parse the message content of any received messages. The name of a parser module always starts with the `pm` prefix.
- **Message modification modules:** Change the content of an `rsyslog` message.
- **String generator modules:** Generate strings based on the message content and cooperate with the template feature provided by `rsyslog`. The name of a string generator module always starts with the `sm` prefix.
- **Library modules:** Library modules provide functionality for other loadable modules. These modules are loaded automatically by `rsyslog` when needed and cannot be configured by the user.

Messages are received by input modules and then passed to one or many parser modules, which generate the in-memory representation of the message and may also modify the message itself. The internal representation is passed to output modules, which may output a message and may also modify message object content.

A list of available modules and detailed descriptions can be found at:

[http://www.rsyslog.com/doc/rsyslog\\_conf\\_modules.html](http://www.rsyslog.com/doc/rsyslog_conf_modules.html)

## Rules

A rule is specified by a *filter* part, which selects a subset of `rsyslog` messages, and an *action* part, which specifies what to do with the selected messages. To define a rule in the `/etc/rsyslog.conf` configuration file, define both a *filter* and an *action* on one line and separate them with one or more spaces or tabs.

## Filters

`rsyslog` offers various ways to filter `rsyslog` messages according to various properties. A defined filter is called a selector.

## Actions

Actions specify what is to be done with the filtered messages.

## Templates

Any output that is generated by `rsyslog` can be modified and formatted by using templates.

## Facility/Priority-Based Filters

Messages are filtered based on two conditions: Facility and priority.

- Syntax to create a filter (or selector):
  - *Facility.Priority*
- Select all kernel rsyslog messages with any priority:
  - kern.\*
- Select all mail rsyslog messages with priority crit and higher:
  - mail.crit
- Select all cron rsyslog messages except those with info or debug priority:
  - cron.!info, !debug



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

rsyslog offers various ways to filter rsyslog messages according to various properties. The following lists the most commonly used filter conditions:

- **Facility/priority-based filters:** Filter rsyslog messages based on two conditions: facility and priority.
- **Property-based filters:** Filter rsyslog messages by any property, such as *timegenerated* or *syslogtag*.
- **Expression-based filters:** Select rsyslog messages according to defined arithmetic, Boolean, or string operations. Expression-based filters use rsyslog scripting language.

### Facility/Priority-Based Filters

Facility/priority-based filters filter rsyslog messages based on two conditions: *facility* and *priority*. To create a selector, use the syntax:

*Facility.Priority*

## Facility

Facility specifies the subsystem that produces a specific `rsyslog` message and can be represented by one of the following keywords:

- `auth/authpriv`: security/authorization messages
- `cron`: crond messages
- `daemon`: other system daemons
- `kern`: kernel messages
- `lpr`: line printer subsystem
- `mail`: mail system
- `news`: network news subsystem
- `syslog`: messages generated internally by `rsyslogd`
- `user`: user-level messages
- `UUCP`: UUCP subsystem
- `local0 through local7`: local use

## Priority

Priority can be represented by one of these keywords (listed in an ascending order):

- `debug`: debug-level messages
- `info`: informational messages
- `notice`: normal bug significant condition
- `warning`: warning conditions
- `err`: error conditions
- `crit`: critical conditions
- `alert`: action must be taken immediately
- `emerg`: system is unstable

All messages of the specified priority and higher are logged according to the given action. Use an asterisk (\*) to define all facilities or priorities. To define multiple facilities and priorities, separate them with a comma ( , ). To define multiple filters on one line, separate them with a semicolon ( ; ). Following are examples of facility/priority-based filters:

To select all kernel messages with any priority:

```
kern.*
```

To select all mail messages with priority `crit` and higher:

```
mail.crit
```

Preceding a priority with an exclamation mark (!) selects all `rsyslog` messages except those with the defined priority. The following example selects all `cron` messages, except those with the `info` or `debug` priority:

```
cron.!info,!debug
```

## rsyslog Actions

- Actions specify what to do with the filtered messages.
- Options include:
  - Save rsyslog messages to log files
  - Send rsyslog messages over the network
  - Send rsyslog messages to specific users
  - Execute a program
  - Input rsyslog messages to a database
  - Discard rsyslog messages
- To save cron messages to /var/log/cron.log:
  - cron.\* /var/log/cron.log
- To send rsyslog messages over the network:
  - \*.\* @example.com:18



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Actions specify what is to be done with the messages filtered out by an already-defined selector. The following are some of the actions that you can define in a rule:

### Saving rsyslog Messages to Log Files

The majority of actions specify to save an rsyslog message to a log file. This is done by specifying a file path after your already-defined selector. The following is a rule comprised of a selector that selects all cron messages and an action that saves them into the /var/log/cron.log log file:

```
cron.* /var/log/cron.log
```

If the file you specified is an existing tty or /dev/console device, rsyslog messages are sent to standard output.

### Sending rsyslog Messages over the Network

To forward rsyslog messages to a remote machine, use the following syntax:

```
@ [ (<OPTION>) ] <HOST>: [<PORT>]
```

The at sign (@) indicates that the rsyslog messages are forwarded to a host using the UDP protocol. To use the TCP protocol, use two at signs with no space between them (@@).

The <HOST> attribute specifies the host that receives the selected rsyslog messages.

The <PORT> attribute specifies the host machine's port.

To forward messages to 192.168.0.1 via the UDP protocol:

```
*.* @192.168.0.1
```

To forward messages to "example.com" using port 18 and the TCP protocol:

```
*.* @@example.com:18
```

## Sending rsyslog Messages to Specific Users

Specify the username to send rsyslog messages to. To specify more than one user, separate each username with a comma ( , ). To send messages to every user that is currently logged on, use an asterisk (\*).

## Executing a Program

You can execute a program for selected rsyslog messages. To specify a program to be executed, prefix it with a caret character (^). Specify a template that formats the received message and passes it to the specified executable as a one-line parameter.

## Inputting rsyslog Messages in a Database

Selected rsyslog messages can be directly written into a database table by using the database writer action. The database writer uses the following syntax:

```
:<PLUGIN>:<DB_HOST>,<DB_NAME>,<DB_USER>,<DB_PASSWORD>;[<TEMPLATE>]
```

## Discarding rsyslog Messages

To discard selected messages, use the tilde character (~). The following rule discards any cron messages:

```
cron.* ~
```

For each selector, you are allowed to specify multiple actions. To specify multiple actions for one selector, write each action on a separate line and precede it with an ampersand character (&). Only the first action is allowed to have a selector specified on its line. The following is an example of a rule with multiple actions:

```
kern.=crit joe  
& ^test-program;temp  
& @192.168.0.1
```

In the preceding example, all kernel messages with the critical priority (crit) are:

- Sent to user joe
- Processed by the template temp and passed on to the test-program executable
- Forwarded to 192.168.0.1 via the UDP protocol

## rsyslog Templates

Templates modify and format output generated by rsyslog.

- Syntax to create a template:
  - \$template <TEMPLATE\_NAME>, "text %<PROPERTY>% more text", [<OPTION>]
- Templates can be used to generate dynamic file names:
  - \$template DynamicFile, "/var/log/test\_logs/%timegenerated%-test.log"
- Template example:
  - \$template verbose, "%syslogseverity%,%syslogfacility%,%timegenerated%,%hostname%,%syslogtag%,%msg%\n"
  - \*.\* /var/log/logfile; verbose



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Any output that is generated by rsyslog can be modified and formatted by using templates.

To create a template, use the following syntax:

```
$template <TEMPLATE_NAME>, "text %<PROPERTY>% more text", [<OPTION>]
```

- \$template: The template directive that defines a template
- <TEMPLATE\_NAME>: The name of the template
- "...": The template text is between the quotation marks
- %<PROPERTY>%: The property is specified within percent signs
- <OPTION>: Specifies options that modify the template functionality

Templates can be used to generate dynamic file names. Specify a property as a part of the file path to create a new file for each unique property. For example, use the timegenerated property to generate a unique file name for each rsyslog message:

```
$template DynamicFile, "/var/log/test_logs/%timegenerated% -test.log"
```

The \$template directive only specifies the template. You must use it inside a rule for it to take effect. Dynamic files are represented by a template and a question mark (?) prefix.

Example:

```
*.* ?DynamicFile
```

## Properties

Properties defined inside a template allow you to access various contents of an `rsyslog` message by using a property replacer. Define a property inside a template using the following syntax:

- `%<PROPERTY_NAME> [ :<FROM_CHAR>:<TO_CHAR>:<OPTION>] %`
- `<PROPERTY_NAME>`: Specifies the name of a property
- `<FROM_CHAR>` and `<TO_CHAR>`: Denotes a range of characters that the specified property will act upon
- `<OPTION>`: Specifies any property options

See `/usr/share/doc/rsyslog-<version-number>/property_replacer.html` for a list of available properties and property options. Properties are also described at [http://www.rsyslog.com/doc/property\\_replacer.html](http://www.rsyslog.com/doc/property_replacer.html). The following are property examples:

The following property obtains the entire message text of an `rsyslog` message:

```
%msg%
```

The following property obtains the first two characters of an `rsyslog` message:

```
%msg:1:2%
```

The following property drops the last line-feed character of an `rsyslog` message:

```
%msg:::drop-last-lf%
```

The following property formats the first 10 characters of the time stamp according to the RFC 3999 date standard:

```
%timegenerated:1:10:date-rfc3339%
```

## Template Example

The following example defines a template named `verbose` which formats an `rsyslog` message to output the message's severity, facility, time stamp, host name, message tag, and message text, and end with a new line:

```
$template verbose, "%syslogseverity%, %syslogfacility%,
%timegenerated%, %hostname%, %syslogtag%, %msg%\n"
```

To use the template for `/var/log/logfile` messages, include the template name as follows:

```
*.*    /var/log/logfile; verbose
```

## Configuring Log Rotation (`logrotate`)

- `logrotate` is a utility to automatically manage log files.
- `/etc/logrotate.conf` is the global configuration file for all logs.
- The `/etc/logrotate.d` directory contains a separate configuration file for any specific log file.
- Configuration options include:
  - How often to rotate files
  - The number of rotated log files to keep
  - Scripts to run before or after rotating
  - Specify log files to be mailed
  - Enable compression of log files
- See `man logrotate` for a list of options.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Most log files are located in the `/var/log` directory. Some services such as `cups`, `httpd`, and `samba` have a directory within `/var/log` for their log files.

The `logrotate` utility helps manage log files automatically by rotating, compressing, mailing, and removing each as you specify. Rotating means saving a series of log files, renaming each file as a new one is saved. Log files are rotated so file sizes do not become too large. Rotating allows you to keep log information for future reference.

Some files in `/var/log` have numbers at the end of the file name. These numbers represent a rotated log with the time stamp added to the log file name.

Normally `logrotate` is run as a daily cron job (`/etc/cron.daily/logrotate`) which automatically rotates log files according to the `/etc/logrotate.conf` configuration file and the configuration files in the `/etc/logrotate.d` directory. You can configure how often to rotate files:

- Daily
- Weekly
- Monthly

You also can specify the number of rotated log files to keep. These parameters are configured in the `/etc/logrotate.conf` configuration file.

## /etc/logrotate.conf

The following is a sample /etc/logrotate.conf configuration file:

```
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# uncomment this if you want your log files compressed
compress
```

In the example, log files are rotated weekly, rotated log files are kept for four weeks, and all rotated log files are compressed by gzip into the .gz format.

## /etc/logrotate.d

You can create a separate configuration file for any specific log file in the /etc/logrotate.d directory and define any configuration options there. These options override the global options in /etc/logrotate.conf and also define additional options. Oracle Linux provides a few separate configuration files by default:

```
# ls /etc/logrotate.d
cups dracut httpd psacct sssd syslog up2date yum
```

The following is an example of the /etc/logrotate.d/httpd configuration file:

```
/var/log/httpd/* log {
    missingok
    notifempty
    sharedscripts
    delaycompress
    postrotate
        /sbin/service httpd reload > /dev/null 2>/dev/null || true
    endscript
}
```

The options listed indicate the following:

- **missingok:** If the log file is missing, do not issue an error message.
- **notifempty:** Do not rotate the log if it is empty.
- **postrotate/endscript:** The lines between these directives are executed after the log file is rotated.
- **sharedscripts:** The postrotate script will only be run once, not once for each log that is rotated.
- **delaycompress:** Postpone compression of the previous log file to the next rotation cycle.

For the full list of directives and configuration options, refer to the `logrotate` man page.

## logwatch

logwatch is a utility to perform basic log file monitoring and analysis.

- To run a daily cron job that monitors logs and sends daily email to administrator:
  - /etc/cron.daily/0logwatch
- The local configuration file is:
  - /etc/logwatch/conf/logwatch.conf
- The main configuration file is:
  - /usr/share/logwatch/default.conf/logwatch.conf
- logwatch can also be run from command line; see:
  - logwatch --help



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

logwatch is a customizable log monitoring system. It will go through system logs for a given time period and report on specific areas of interest.

It may be necessary to install the `logwatch` package. An RPM file is available. After it is installed, `logwatch` is configured by default to run each night from `/etc/cron.daily/0logwatch` and email a report to the root user.

Local configuration options can be set in `/etc/logwatch/conf/logwatch.conf` but the main configuration file is `/usr/share/logwatch/default.conf/logwatch.conf`. Within this file, you can specify various options including:

- Level of detail
- Logfile to report on
- Name of a service to report on
- Username to mail the report to
- File name to save the report to

You can also run `logwatch` from the command line with various options. Run the following command to get information on using `logwatch`:

```
# logwatch --help
```

# Quiz

Which of the following entries in /etc/rsyslog.conf cause warning, err, crit, alert, and emerg messages from the kernel to be logged?

- a. kern.\*
- b. kern.warning
- c. kern.err
- d. \*.kern

## Summary

In this lesson, you should have learned how to:

- Describe the contents of the `rsyslog` configuration file
- Describe facility/priority-based filters
- Describe `rsyslog` actions
- Describe `rsyslog` templates
- Configure log rotation
- Describe `logwatch`



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 20: Overview

The practices for this lesson cover the following:

- Viewing the system log file configuration
- Using `rsyslog` templates
- Using `logwatch`

# 21

## Troubleshooting

ORACLE®

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.

# Objectives

After completing this lesson, you should be able to:

- Describe the two-phased approach to troubleshooting
- Describe the type of information needed to troubleshoot a problem
- Describe the available operating system logs to assist in troubleshooting
- Use the dmesg utility
- Describe the available troubleshooting resources
- Describe causes of common problems
- Describe troubleshooting boot problems
- Describe typical causes of NFS problems



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Two-Phased Approach to Troubleshooting

- Fault Analysis Phase
  - State the problem.
  - Gather information.
  - Identify what is and what is not working.
- Fault Diagnosis Phase
  - Based on the fault analysis findings and past experiences, determine the most probable causes of the fault.
  - Test and verify the probable causes.
  - Take corrective action.
  - Ensure you do not introduce any new problems.
- Document the results of the Fault Analysis and the Fault Diagnosis phases.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Use a two-phased approach to troubleshooting. Begin with the fault analysis phase in which you state the problem and gather as much information as you can about the problem. Problem information can be gathered from error messages, log files, historical information such as previous problems and associated resolutions, and Oracle bug and support websites. Recent system changes are also an important source of information about a system fault.

In the second phase, you determine the most likely causes of the problem from the information you collected. When possible, take into consideration past experiences with diagnosing similar issues. You then test and verify your list of most likely causes. Through a process of elimination, you identify the actual cause of the fault while simultaneously verifying that you can correct the problem and not introduce any new problems.

Always document the steps you took to isolate and correct the problem for future reference.

## Gathering Information

- Get a complete description of the server.
- Describe exactly what the problem is.
  - Symptoms
  - Error messages
- Who is experiencing the problem?
  - One user or several users
- Can the problem be reproduced?
  - Steps to reproduce the problem
  - Is it an intermittent problem?
- Does the problem occur only at certain times of the day or certain days of the week?
- Have any changes been made to the server?



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

When a problem occurs, the first things to think about are how to locate the problem. The more information you have, the better. Ask probing questions to help clarify the problem. Some users may have difficulty in answering the questions but any extra information they can provide might help you find the problem.

Knowing when the problem occurs might help you determine the cause. The problem might occur only at a certain time of day. Or perhaps the problem occurred after a change was made to the server or peripherals, or by some new way in which clients are using the server.

Knowing who is experiencing the problem can help determine if the problem exists in a particular part of the network, or if the problem is application dependent. Determine if one person, one group of users, or a larger group is experiencing the problem.

If the problem can be reproduced, determine what steps are needed to reproduce the problem. Also determine if the problem can be reproduced on another system and by another user. A good procedure to remedy hard-to-reproduce problems is to perform general maintenance on the system such as bringing the system up to date on patches.

# Operating System Logs

- Files under /var/log:
  - boot.log – Messages from bootup
  - messages – Standard system error messages
  - anaconda – O/S install logs
  - dmesg – Log of boot messages showing hardware errors
- Other logs exist for mail, cron, security, and so on.
- Other directories in /var/log/ exist for cups, httpd, samba, and so on.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Linux maintains several system logs that help you administer your systems by informing you of important events. Checking system messages is a logical early step when trying to determine the probable cause of a system fault. At a glance, a system message can provide you with the following information:

- Process name/PID number
- Message ID number
- Facility that generated the message (for example, the kernel or a system daemon)
- Level of severity of the message (for example: emergency, error, warning, notice, or information)
- Message

Probably the most important log is the file /var/log/messages that records a variety of events, including system error messages, system startups, and system shutdowns. Like most other Linux files, the file contains ASCII text, so you can view it with a text editor or the text processing commands.

You can monitor a log file in real time by using the `tail -f <logfile>` command. This command keeps the file open and new messages are appended to the file. Use the `CTRL-C`, command to close the file.

## dmesg Utility

- dmesg: Print out a buffer showing latest hardware issues.
- The command prints only a memory structure (kernel ring buffer) in the memory.
- dmesg does not have times tamps.
- The buffer can truncate when it is full.
  - /var/log/boot\*
  - /var/log/dmesg\*

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The dmesg command is used to examine or control the kernel ring buffer. Messages related to the operation of the kernel are written to the ring buffer. A ring buffer is of a constant size and the oldest messages are removed when new messages are written.

Hardware-related information is available in dmesg output. This includes memory related issues, CPU information, and information about devices.

See `man dmesg` for more information.

## Troubleshooting Resources

- Man pages provide the usage of a command and the available options and configuration parameters.
- Many commands and services have a `-d/-D` option for debugging or a `-v/-V` option for verbose.
- The `/usr/share/doc/` directory contains information about packages installed on your system plus release notes and manuals.
- Oracle Linux administration guides:
  - [http://docs.oracle.com/cd/E37670\\_01/](http://docs.oracle.com/cd/E37670_01/)
- My Oracle Support website contains knowledge articles and other helpful information.
  - <https://support.oracle.com/>



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

When troubleshooting a problem, knowing what configuration files are used, the type of information stored in configuration files, and what services need to be running are the largest factors in troubleshooting.

Many resources are available to assist in troubleshooting. The Linux man pages provide configuration file parameters and the available options to the commands and services. Often there is a `-d` or `-D` option to a command, which allows you to turn on various levels of debugging to assist in troubleshooting a problem. These are normally turned off, or not specified as the default, because the amount of information displayed in the output is usually not desired when everything is running smoothly. However, when a problem occurs, the detailed information can be useful when troubleshooting where the problem may be.

The `/usr/share/doc` directory is another helpful resource. It is the central documentation directory and contains various documentation and release notes for your system.

The administration guides at the URL listed in the slide is another source of information.

The My Oracle Support (MOS) website also contains valuable information to assist in troubleshooting system problems.

Internet searches can also be helpful in troubleshooting. Entering the exact error message from a log file can often point you to a resolution to your problem.

# My Oracle Support

The screenshot shows a Mozilla Firefox browser window displaying the Oracle Support website. The search query 'linux boot' has been entered into the search bar. The results page lists 1518 articles, with the first few visible:

- 03-Apr-2013 Oracle Solaris 10 Booting Resource Center [Article ID 1397794.]
- 15-Aug-2013 How to Boot Oracle Linux into Rescue Mode [Article ID 1516777.]
- 01-Sep-2013 Oracle Linux 5.8 [Release OL5.8.4 (or OL5.8)] Linux x86-64 Linux x86 Gual 11 is possible to boot Linux in rescue [Article ID 1327564.]
- 10-Jul-2013 E43: Boot Application Server Domain or Process Scheduler on Linux Results in a Generic MessageBox [Article ID 1327564.]
- 01-Jun-2009 Pillar Axiom: Linux Servers Using Device-Mapper Fail to Mount Axiom SAN LUNs at Boot Time [Article ID 1391138.]
- 22-Jul-2012 Explanation of SCSI Error on Linux Boot: "sd0: I/O error: dev 0800, sector 0" [Article ID 556776.]
- 24-Aug-2013 Exalogic Compute Node Stuck At GRUB Linux Boot Sequence [Article ID 1370639.]
- 12-May-2013 Oracle Linux Fails to Boot from Multipathed LVM Storage [Article ID 1338089.]
- 12-Feb-2013 Oracle Linux Installation (to 5.6) (via update / yum) Installing Oracle Linux 5.6 from scratch the boot process may hang [Article ID 1040235.]
- 30-Jul-2012 Oracle 10g on Linux: automatic startup at boot [Article ID 1040235.]
- 20-Feb-2013 Sun Fire[TM] V20z v42z servers: Red Hat Linux Enterprise 3.0 AS: Boot problem after BIOS update/defa [Article ID 1017849.]

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The My Oracle Support website (<https://support.oracle.com/>) contains knowledge articles and other helpful information to assist in troubleshooting a problem.

The slide shows the results of a search for “linux boot.” A list of knowledge articles related to the query is displayed. Click each article to view the details.

My Oracle Support requires a user login and password.

## Causes of Common Problems

- Service(s) not running:
  - Use the `service` command to start a service or check the status of a service.
  - Use the `chkconfig` command to start a service at boot time.
- Configuration errors:
- Firewall (`iptables`) is prohibiting a connection.
  - Stop `iptables` and test to determine if a firewall is blocking.
- PAM is prohibiting authentication:
  - View `/var/log/secure` for authentication error messages.
- SELinux is denying a connection:
  - Set SELinux to permissive mode and test.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Use the `service` command to ensure a particular daemon (or service) is running. For example, to obtain the status of the `httpd` daemon:

```
# service httpd status  
httpd is stopped
```

Omit the third argument to get the usage. Example:

```
# service httpd  
Usage: httpd {start|stop|restart|condrestart|try-restart|force-  
reload|status|fullstatus|graceful|help|configtest}
```

Always start (or restart) a service whenever making a change to an associated configuration file. For example, after changing a network interface file in `/etc/sysconfig/network-scripts/`, restart the `network` service:

```
# service network restart
```

Use the `chkconfig` command to configure a service to start at boot time. For example, to configure the `httpd` daemon to start at run levels 3 and 5:

```
# chkconfig --level 35 httpd on
```

The `chkconfig` command does not actually start or stop a service.

Ensure configuration files contain valid information. Each service has at least one associated configuration file. Refer to the man pages or administration guides for configuration file parameters. Configuration files often contain comments that describe configuration file parameters.

Many of the system configuration files are located in the /etc/sysconfig/ directory. Refer to the /usr/share/doc/initscripts\*/sysconfig.txt file for information about these files.

Do not make kernel setting changes from the command line. To preserve custom settings for kernel features, add them to the /etc/sysctl.conf file. Changes made in the /etc/sysctl.conf file take effect immediately when issuing the following command:

```
# sysctl -p
```

The iptables service (firewall) is often the cause of a problem with a client-server process. Use the service iptables stop command to temporarily stop iptables and re-test to determine if the problem is resolved. If so, you can create a rule to open a specific port and restart the iptables service.

PAM modules might be causing authentication errors. Entries are usually written to the /var/log/secure log file when PAM is denying access.

SELinux stands for “Security-Enhanced Linux” and is covered in another course in the Oracle Linux curriculum map, but SELinux is often the cause of a problem. For purposes of this course, you can use the sestatus command to display information about SELinux.

```
# sestatus
SELinux status:      enabled
...
Current mode:        enforcing
```

From this output, you can see that SELinux is enabled and is in enforcing mode. You can temporarily change SELinux to “permissive” mode and re-test to see if the problem is fixed. Use the setenforce 0 command to temporarily change SELinux to “permissive” mode.

```
# setenforce 0
# sestatus
SELinux status:      enabled
...
Current mode:        permissive
```

Notice the “Current mode” is now set to “permissive.” To permanently change the mode, edit the /etc/selinux/config file and change the SELINUX directive to “permissive” or “disabled.”

## Troubleshooting Boot Problems

- Configuration errors in the following files can prevent your system from booting:
  - /boot/grub/grub.conf
  - /etc/inittab
  - /etc/fstab
- Boot into rescue mode to correct boot problems.
  - Rescue mode boots from installation media.
  - File systems are mounted under /mnt/sysimage.
  - Use chroot to change the root partition of the rescue mode environment.
  - Then use vi, fsck, rpm, and other utilities to fix the boot problem.
- Use the grub-install to re-install the boot loader.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Knowing the normal sequence of events that occurs in the boot process, and knowing at which point in the process a system had problems, are key to diagnosing and fixing boot-time problems. Configuration errors in important files, such as /boot/grub/grub.conf, /etc/inittab, and /etc/fstab can prevent your system from booting.

Rescue mode allows you to boot from the Oracle Linux installation media instead of booting from your system's hard drive. From rescue mode, you can access files on your hard drive and correct configuration errors, reinstall the boot loader, fix file system errors, or otherwise rescue your system. You might not be able to fix the boot problem but at least you can get copies of important data files.

Rescue mode attempts to mount your file systems under /mnt/sysimage. The /mnt/sysimage is a temporary root partition, not the root partition of the file system used during normal operations. You can use the chroot command to change the root partition of the rescue mode environment to the root partition of your file system. You can then correct any errors in configuration files, run fsck to check and repair a file system, use rpm to install or upgrade software packages, and other commands to rescue your environment.

You can re-install the GRUB boot loader in the event it has been corrupted or overwritten by another operating system. Use the grub-install command to re-install the boot loader.

## Typical Causes of NFS Problems

- The `rpcbind` or NFS daemons are not running:
  - NFS daemons are `nfs` and `nfslock`.
- Syntax errors:
  - On client `mount` command
  - In `/etc/exports` file on server
- Permission problems:
  - Check UIDs and GIDs.
- Firewall is blocking NFS packets:
  - Check `iptables` rules or stop `iptables` service.
- DNS host name resolution:
  - Ensure `/etc/resolv.conf` contains correct entries.



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Typical causes of NFS problems are given in the slide. Check and ensure that the `rpcbind`, `nfs`, and `nfslock` daemons are running on the server. Always start the `rpcbind` service first because the NFS services need `rpcbind` to be running.

Another common problem is syntax errors either in the `mount` command on the client or in the `/etc/exports` file on the server. The format for entries in the `/etc/exports` file is:

```
export-point client1(options) [client2(options) ... ]
```

A common error is inserting a space in the `client (options)` argument. A space after the client identifier and the bracket causes the options to be ignored.

If the NFS file system mounts but you cannot access it, check the permissions and the GIDs and UIDs. NFS requests contain numeric UIDs and GIDs. Just because the username is the same on both the client and the server, it does not mean the UIDs and GIDs are the same.

Firewalls can filter packets necessary for NFS. Check your `iptables` rules and service. The NFS service uses port 2049. The `rpcbind` service uses port 111.

Host name resolution provided by DNS must also be configured properly for NFS to work. Check the `/etc/resolv.conf` file and ensure you are querying the correct DNS server.

# Quiz

Which of the following commands is useful in determining if your system has hardware-related errors?

- a. service
- b. ps
- c. lsmod
- d. dmesg



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Summary

In this lesson, you should have learned:

- The two-phased approach to troubleshooting
- The type of information needed to troubleshoot a problem
- The available operating system logs to assist in troubleshooting
- Use of the `dmesg` utility
- The available troubleshooting resources
- Causes of common problems
- Troubleshooting boot problems
- Typical causes of NFS problems



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

## Practice 21: Overview

The practices for this lesson involve troubleshooting some common problems including:

- System boots into single-user mode
- Status commands fail
- A cron job fails to run
- User cannot log in
- File system troubleshooting
- Logical volume space is exhausted
- Network connectivity problem
- NFS permission problem
- Remote access problem
- Log file is not getting updated



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

In these practice, you configure a scenario and verify that everything works. You are directed to run a program that introduces an error. You are given some hints, or things to look at and check for, with regard to diagnosing the problem. Refer to previous lessons when necessary and attempt to diagnose and fix the problem.

Unauthorized reproduction or distribution prohibited. Copyright© 2017, Oracle and/or its affiliates.

Mohammed Ameen (mohammed.a.ameen@oracle.com) has a  
non-transferable license to use this Student Guide.